

Nama : Adinda Ivanka Maysanda Putri

Kelas : SIB 2B

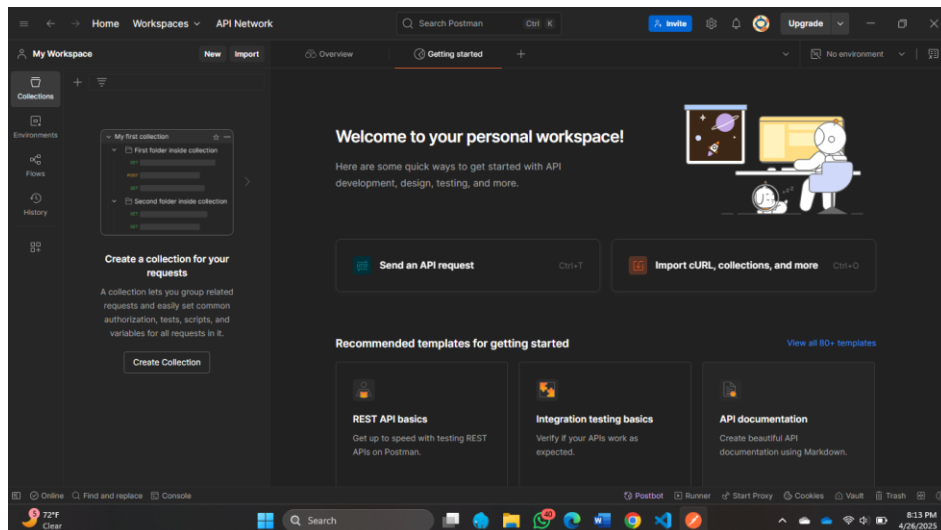
NIM : 2341760058

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.

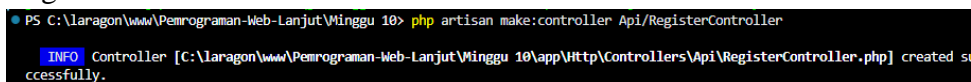


2. Lakukan instalasi JWT dengan mengetikkan perintah berikut:
`composer require tymon/jwt-auth:2.1.1`
Pastikan Anda terkoneksi dengan internet.



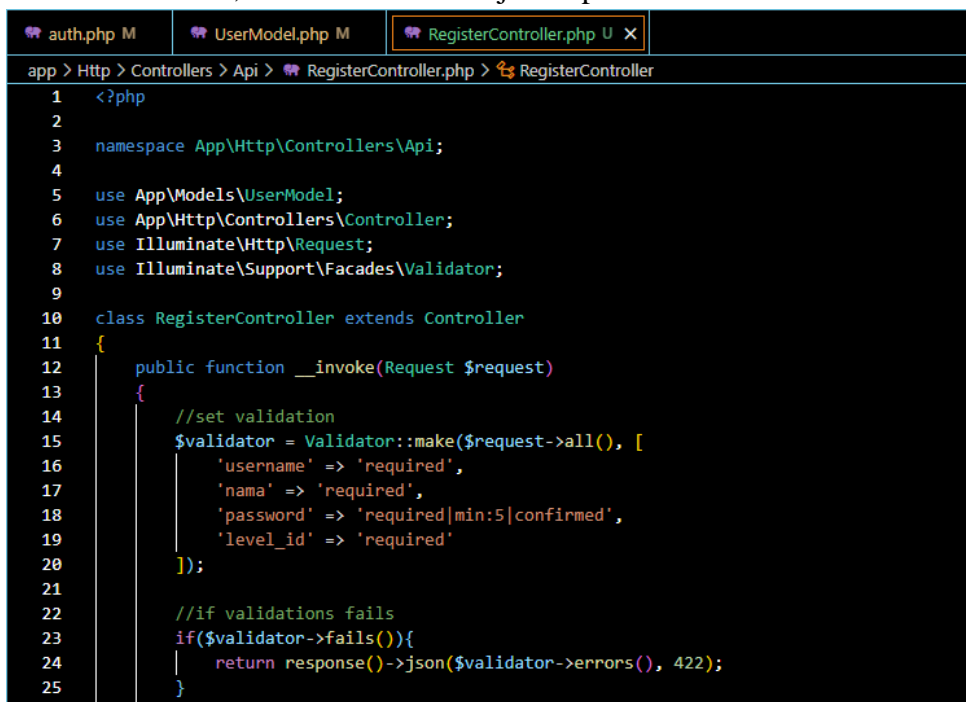
```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\Relations\BelongsTo;
8 use Tymon\JWTAuth\Contracts\JWTSubject;
9 use Illuminate\Foundation\Auth\User as Authenticatable; //implementasi class authenticatable
10
11
12 class UserModel extends Authenticatable implements JWTSubject
13 {
14     use HasFactory;
15     public function getJWTIdentifier(){
16         return $this->getKey();
17     }
18
19     public function getJWTCustomClaims(){
20         return [];
21     }
22 }
```

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\Pemrograman-Web-Lanjut\Minggu 10> php artisan make:controller Api/RegisterController
INFO Controller [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10\app\Http\Controllers\Api\RegisterController.php] created successfully.
```

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

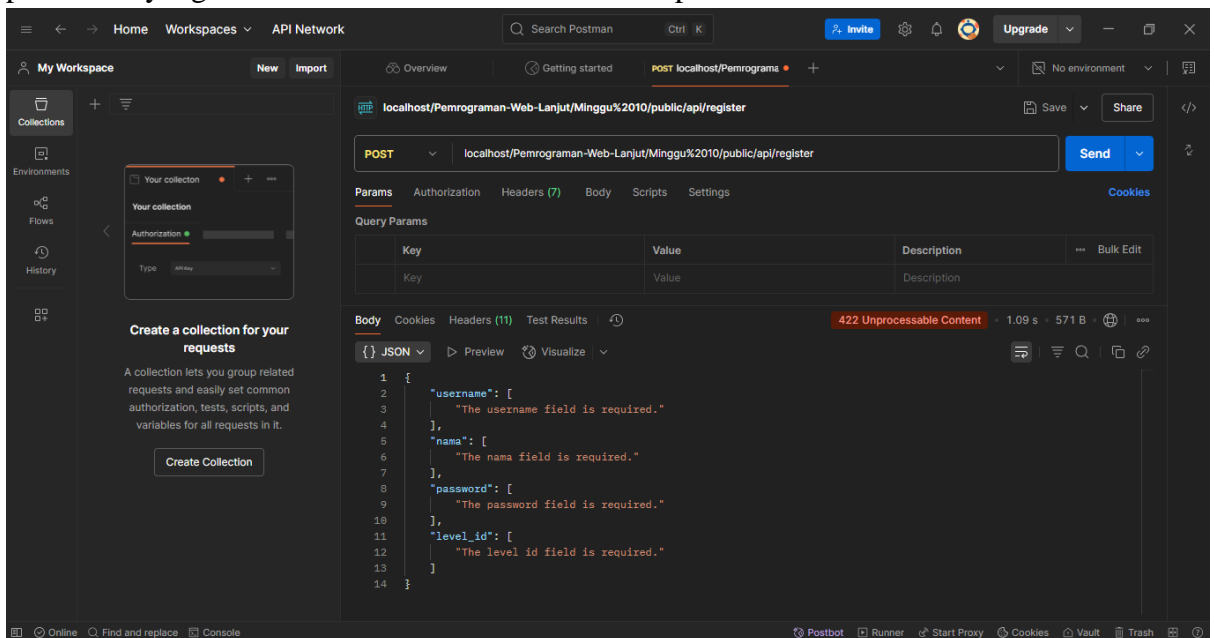


```
1 <?php
2
3 namespace App\Http\Controllers\Api;
4
5 use App\Models\UserModel;
6 use App\Http\Controllers\Controller;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Validator;
9
10 class RegisterController extends Controller
11 {
12     public function __invoke(Request $request)
13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

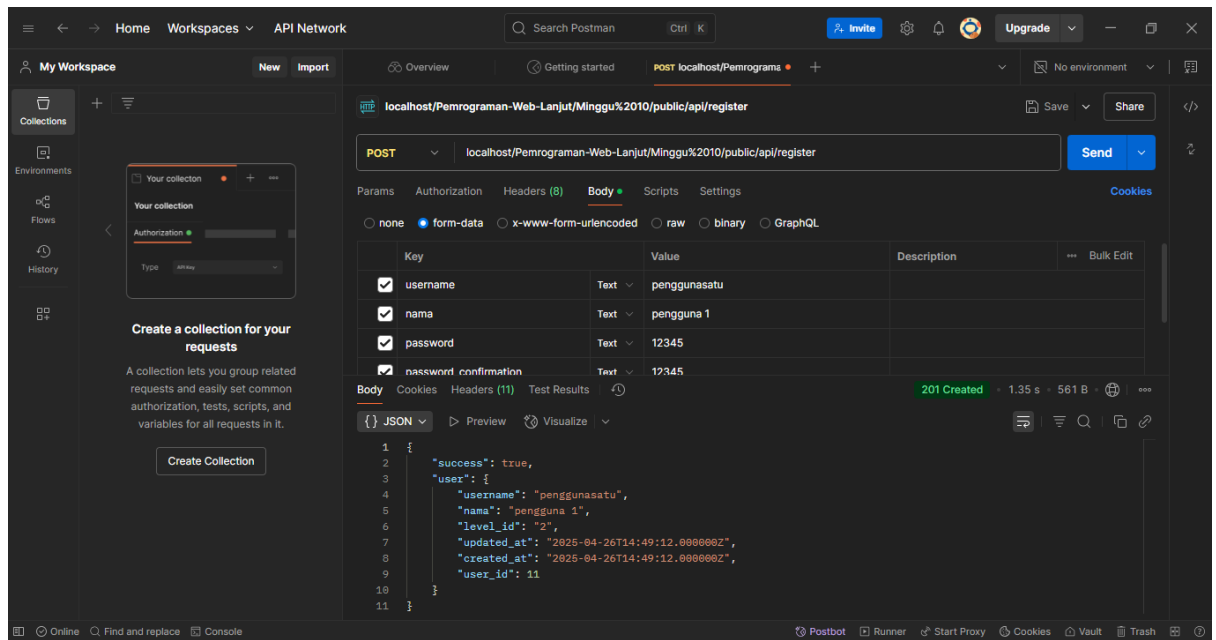
10. Selanjutnya buka routes/api.php, ubah semua kode menjadi seperti berikut.

```
auth.php M UserModel.php M RegisterController.php U api.php M X
routes > api.php > ...
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 /*
7 |-----
8 | API Routes
9 |-----
10 |
11 | Here is where you can register API routes for your application. These
12 | routes are loaded by the RouteServiceProvider and all of them will
13 | be assigned to the "api" middleware group. Make something great!
14 |
15 */
16
17 Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
```

11. Jika sudah, kita akan melakukan uji coba REST API melalui aplikasi Postman. Buka aplikasi Postman, isi URL localhost/PWL_POS/public/api/register serta method POST. Klik Send. Jika berhasil akan muncul error validasi seperti gambar di atas. Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.



12. Sekarang kita coba masukkan data. Klik tab Body and 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 screenshoot hasil percobaan Anda.



13. Lakukan commit perubahan file pada Github.

Praktikum 2 – Membuat RESTful API Login

- 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.

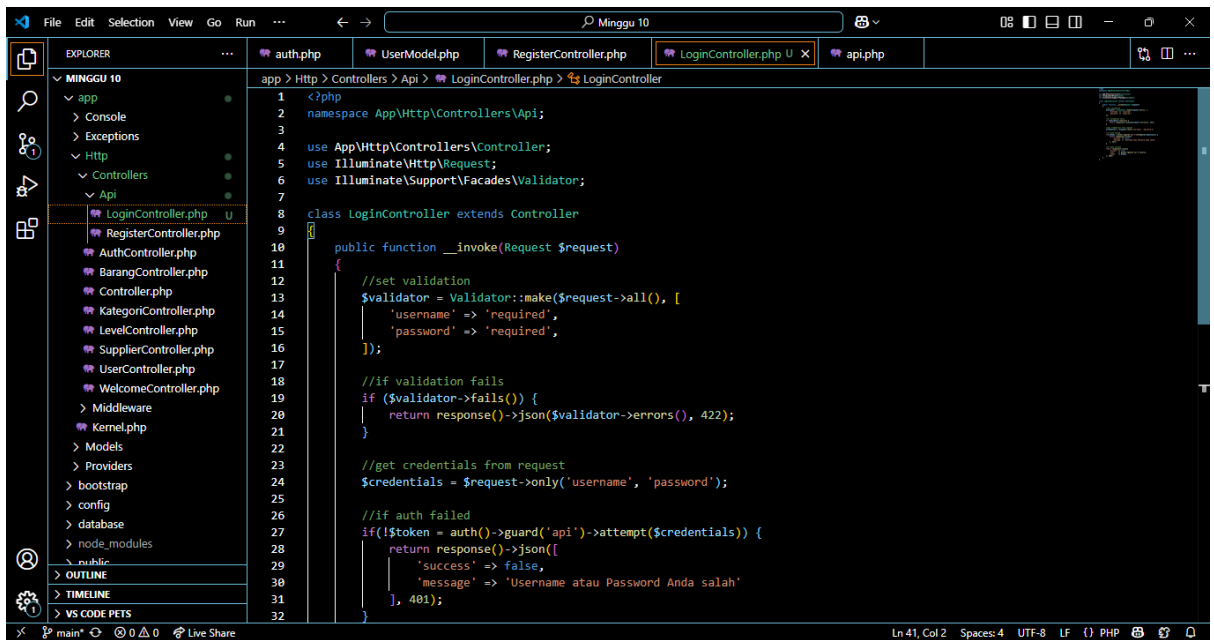
```

PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10> php artisan make:controller Api/LoginController

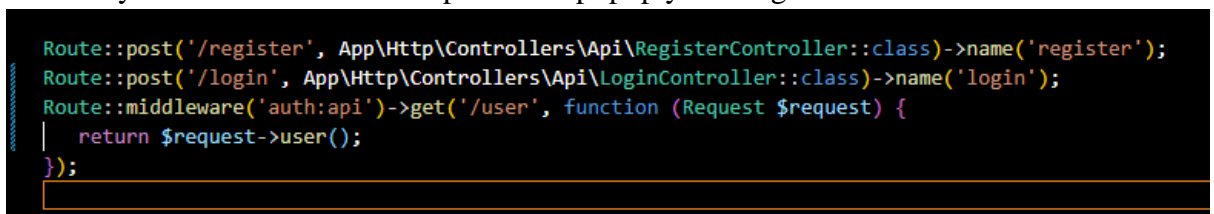
INFO Controller [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10\app\Http\Controllers\Api>LoginController.php] created successfully.

```

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

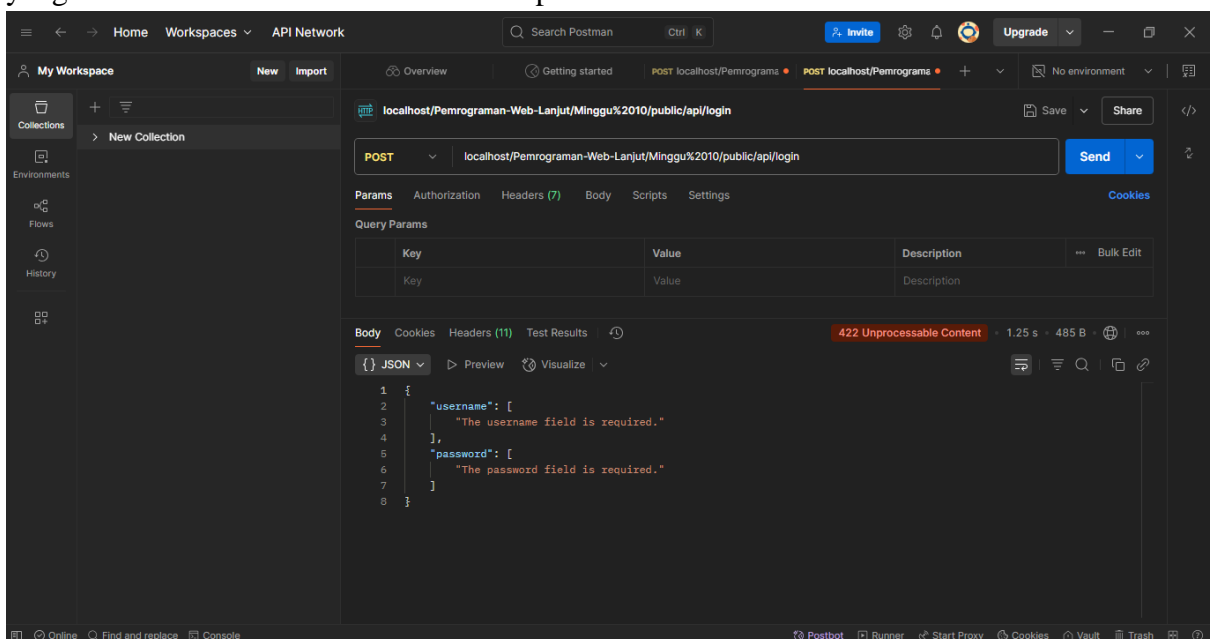


3. Berikutnya tambahkan route baru pada file api.php yaitu /login dan /user.

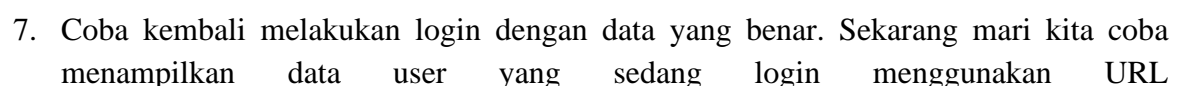
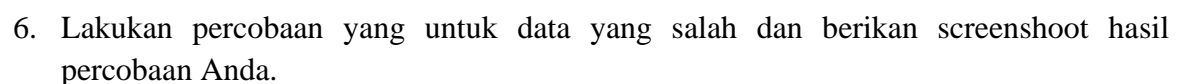


4. Jika sudah, kita akan melakukan uji coba REST API melalui aplikasi Postman. Buka aplikasi Postman, isi URL localhost/PWL_POS/public/api/login serta method POST. Klik Send

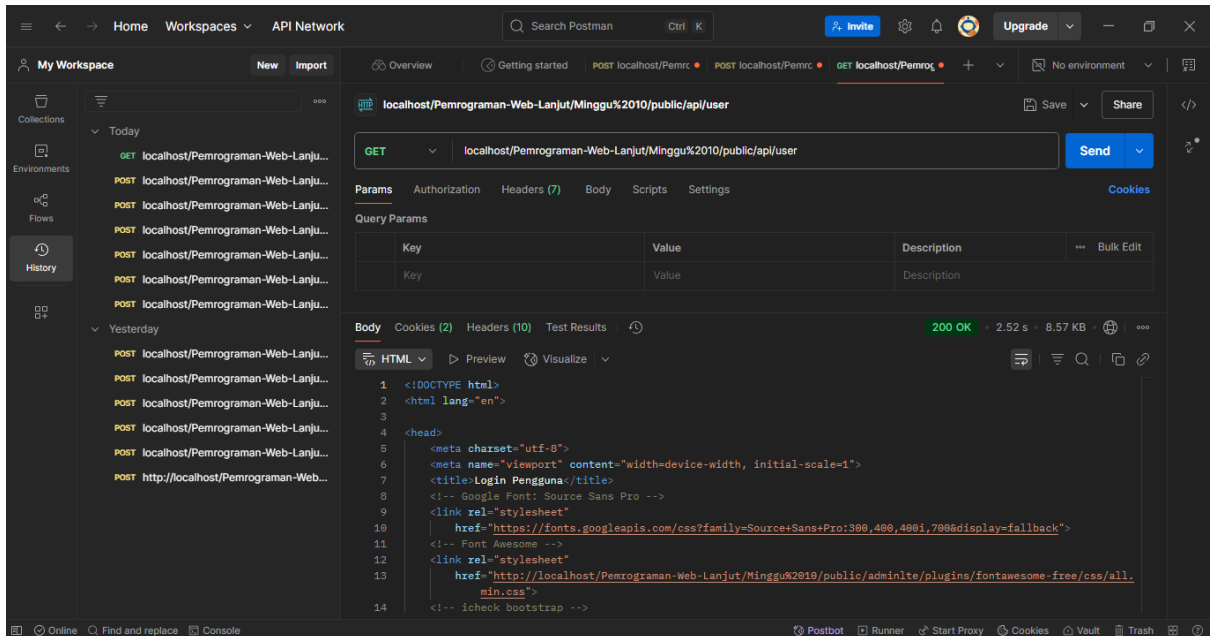
Jika berhasil akan muncul error validasi seperti gambar di atas. Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.



- Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.



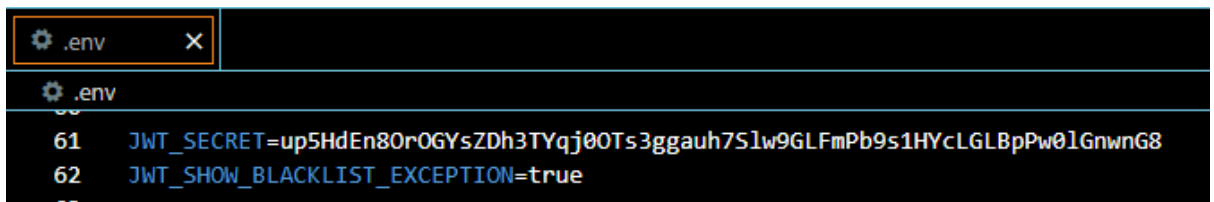
localhost/PWL_POS/public/api/user dan method GET. Jelaskan hasil dari percobaan tersebut.



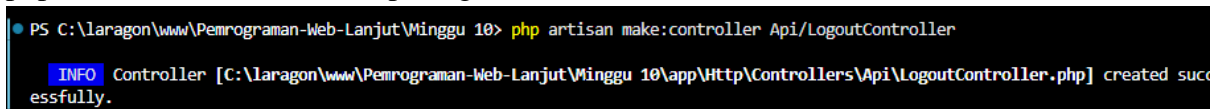
8. Lakukan commit perubahan file pada Github.

Praktikum 3 – Membuat RESTful API Logout

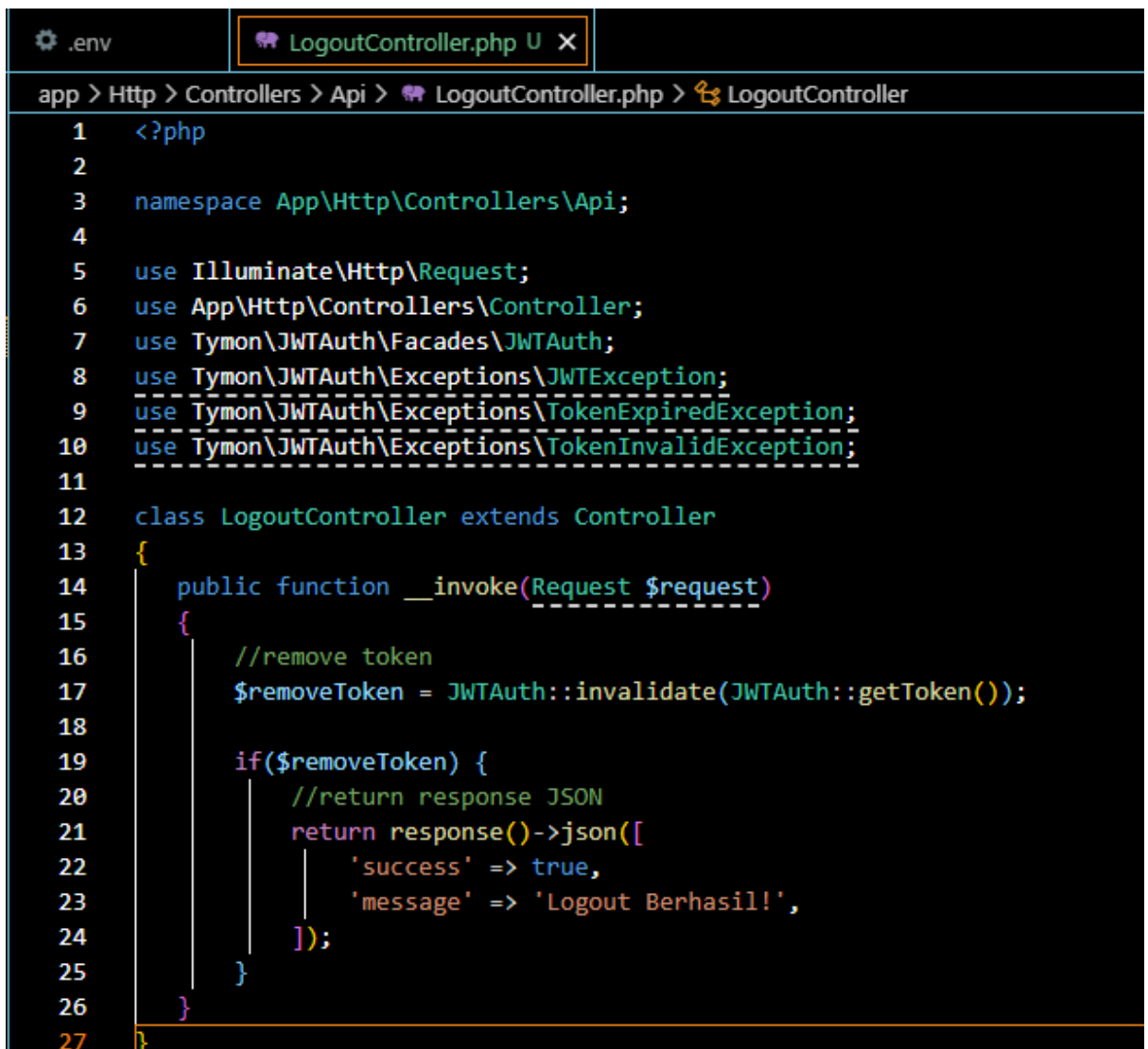
1. Tambahkan kode berikut pada file .env
JWT_SHOW_BLACKLIST_EXCEPTION=true



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

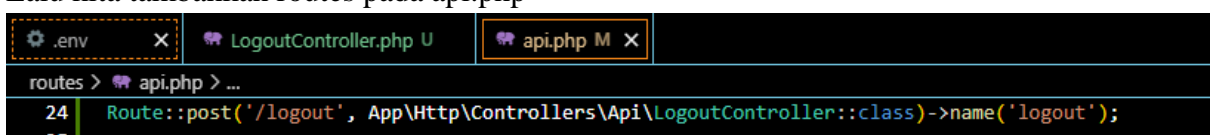


3. Buka file tersebut dan ubah kode menjadi seperti berikut.



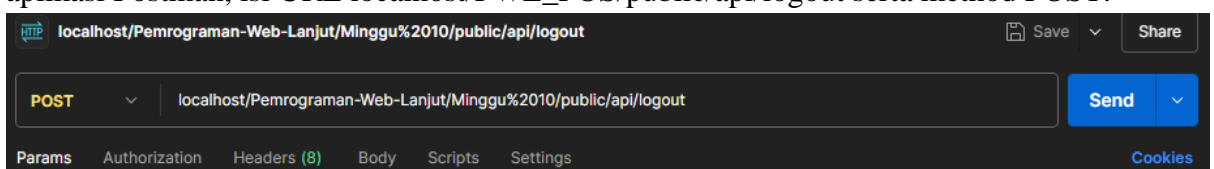
```
1 <?php
2
3 namespace App\Http\Controllers\Api;
4
5 use Illuminate\Http\Request;
6 use App\Http\Controllers\Controller;
7 use Tymon\JWTAuth\Facades\JWTAuth;
8 use Tymon\JWTAuth\Exceptions\JWTException;
9 use Tymon\JWTAuth\Exceptions\TokenExpiredException;
10 use Tymon\JWTAuth\Exceptions\TokenInvalidException;
11
12 class LogoutController extends Controller
13 {
14     public function __invoke(Request $request)
15     {
16         //remove token
17         $removeToken = JWTAuth::invalidate(JWTAuth::getToken());
18
19         if($removeToken) {
20             //return response JSON
21             return response()->json([
22                 'success' => true,
23                 'message' => 'Logout Berhasil!',
24             ]);
25         }
26     }
27 }
```

4. Lalu kita tambahkan routes pada api.php

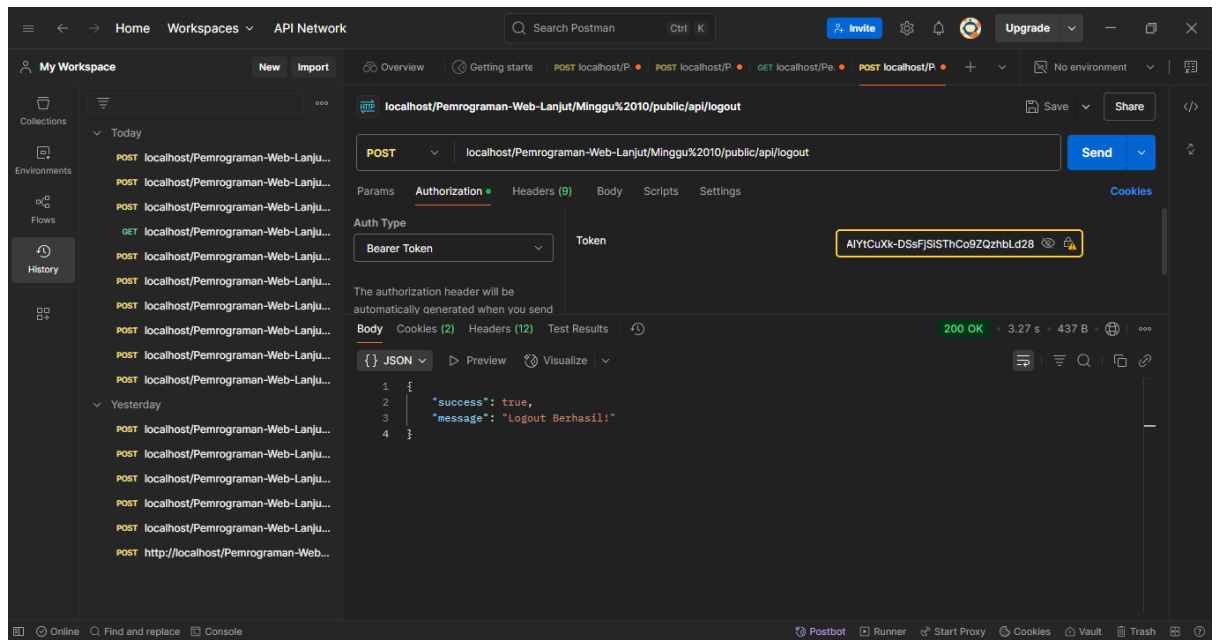


```
24 Route::post('/logout', App\Http\Controllers\Api\LogoutController::class)->name('logout');
25
```

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\Pemrograman-Web-Lanjut\Minggu 10> php artisan make:controller Api/LevelController

INFO Controller [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10\app\Http\Controllers\Api\LevelController.php] created successfully.
```

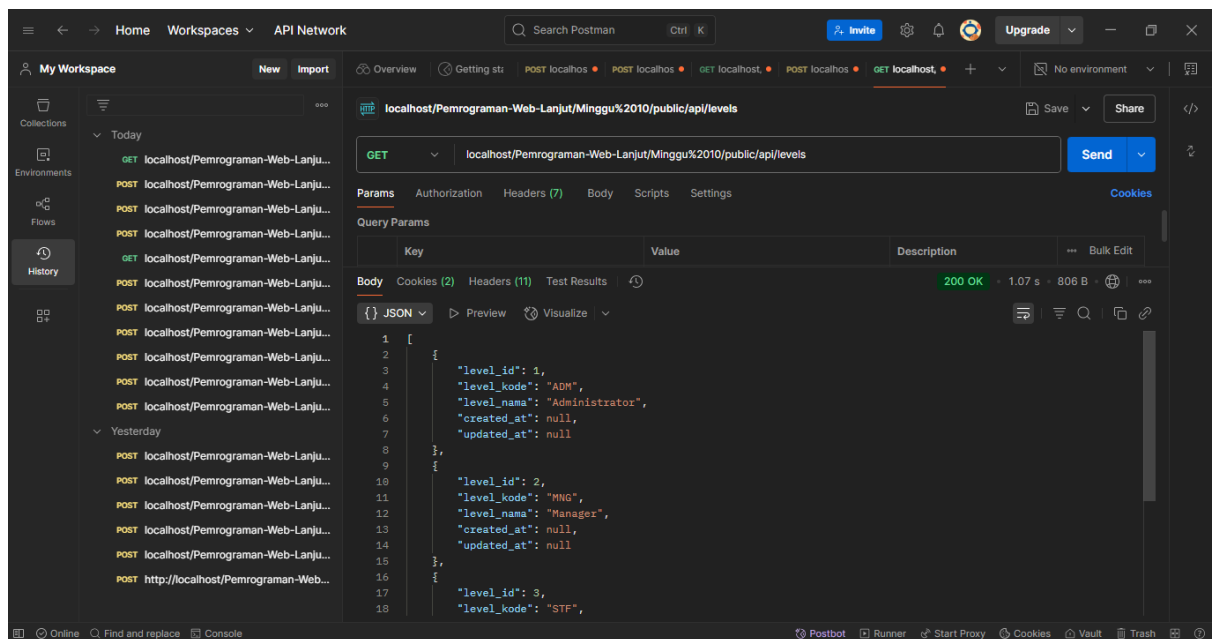
2. Setelah berhasil, buka file tersebut dan tuliskan kode seperti berikut yang berisi fungsi CRUDnya.

```
LevelController.php U X
app > Http > Controllers > Api > LevelController.php > LevelController
1  <?php
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  class LevelController extends Controller
10 {
11     public function index(){
12         return LevelModel::all();
13     }
14     public function store(Request $request)
15     {
16         $level = LevelModel::create($request->all());
17         return response()->json($level, 201);
18     }
19
20     public function show(LevelModel $level)
21     {
22         return LevelModel::find($level);
23     }
24
25     public function update(Request $request, LevelModel $level)
26     {
27         $level->update($request->all());
28         return LevelModel::find($level);
29     }
30
31     public function destroy(LevelModel $user)
32     {
```

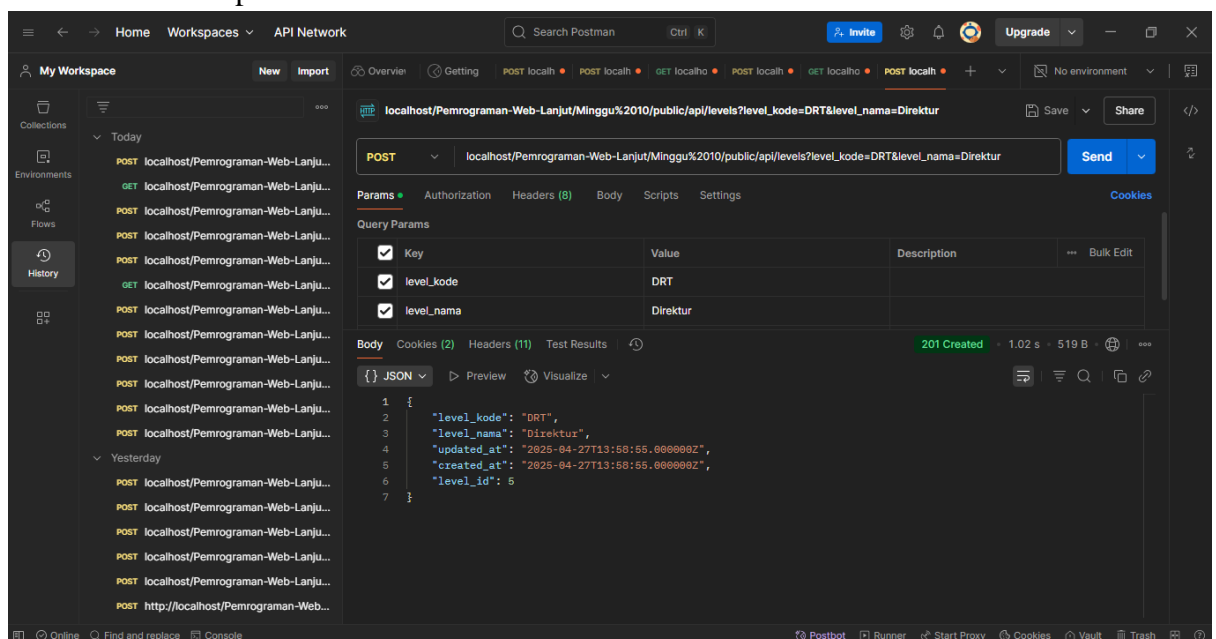
3. Kemudian kita lengkapi routes pada api.php.

```
LevelController.php U  api.php M X
routes > api.php > Closure
27 //Route Levels
28 Route::get('levels', [LevelController::class, 'index']);
29 Route::post('levels', [LevelController::class, 'store']);
30 Route::get('levels/{level}', [LevelController::class, 'show']);
31 Route::put('levels/{level}', [LevelController::class, 'update']);
32 Route::delete('levels/{level}', [LevelController::class, 'destroy']);
```

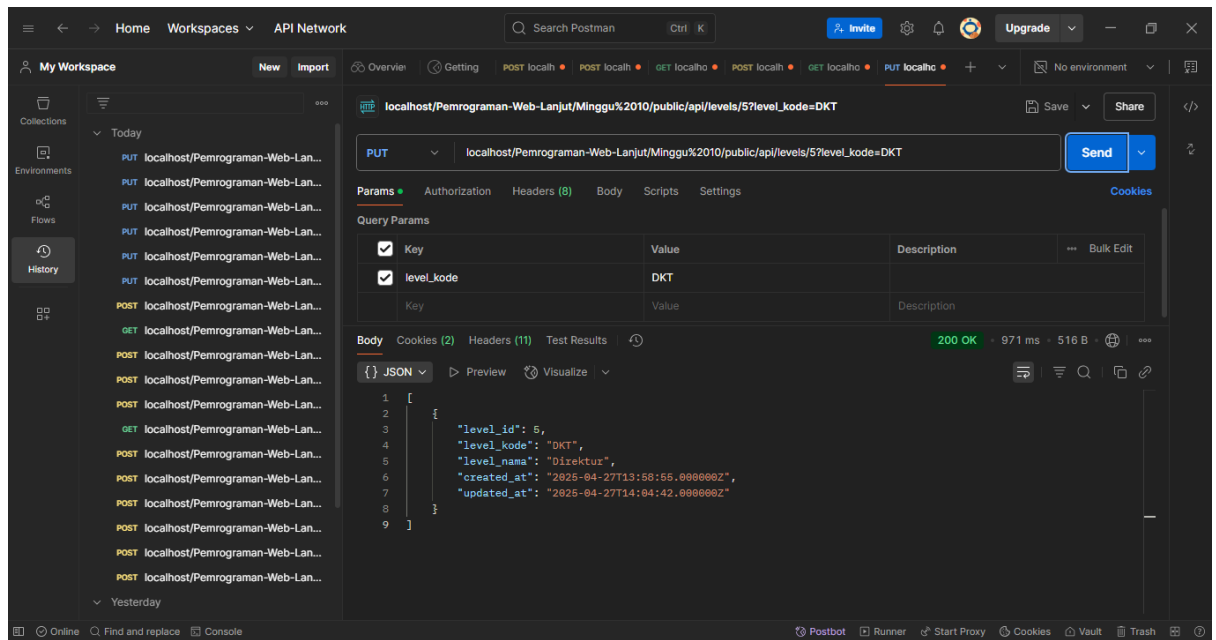
4. Jika sudah. Lakukan uji coba API mulai dari fungsi untuk menampilkan data. Gunakan URL: localhost/PWL_POS-main/public/api/levels dan method GET. Jelaskan dan berikan screenshot hasil percobaan Anda.



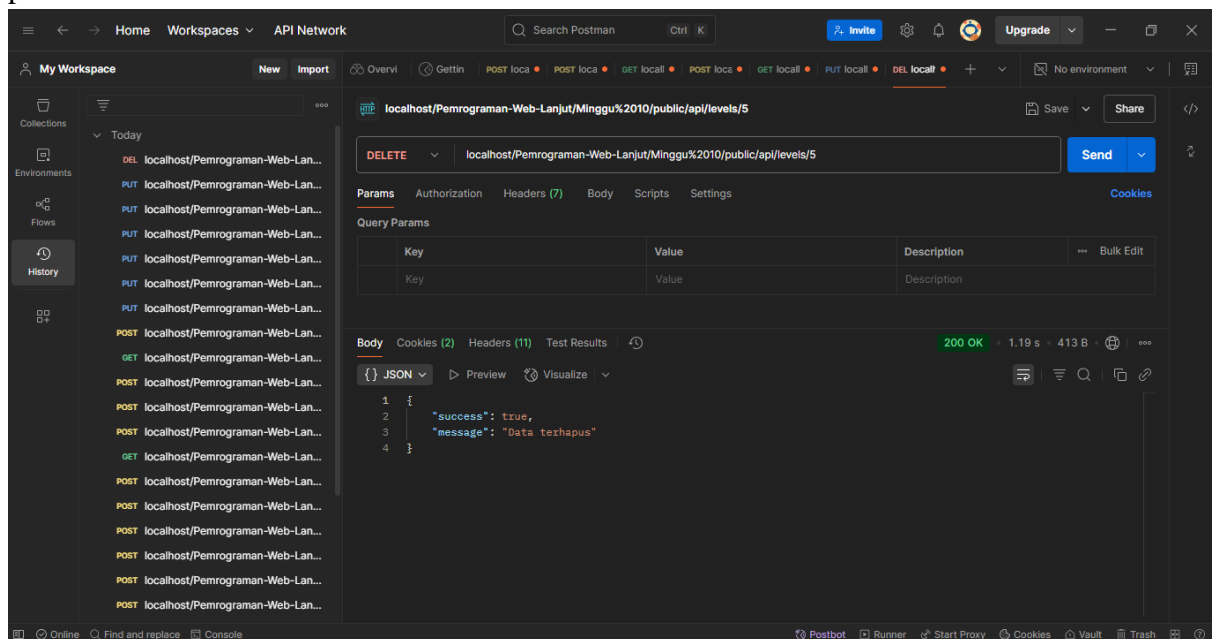
- Kemudian, lakukan percobaan penambahan data dengan URL : `localhost/PWL_POS main/public/api/levels` dan method POST seperti di bawah ini. Jelaskan dan berikan screenshoot hasil percobaan Anda.



- Berikutnya lakukan percobaan menampilkan detail data. Jelaskan dan berikan screenshoot hasil percobaan Anda.
- Jika sudah, kita coba untuk melakukan edit data menggunakan `localhost/PWL_POS main/public/api/levels/{id}` dan method PUT. Isikan data yang ingin diubah pada tab Param. Jelaskan dan berikan screenshoot hasil percobaan Anda.



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



- Lakukan commit perubahan file pada Github.

TUGAS

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

```

PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10> php artisan make:controller Api/UserController

INFO Controller [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10\app\Http\Controllers\Api\UserController.php] created successfully.

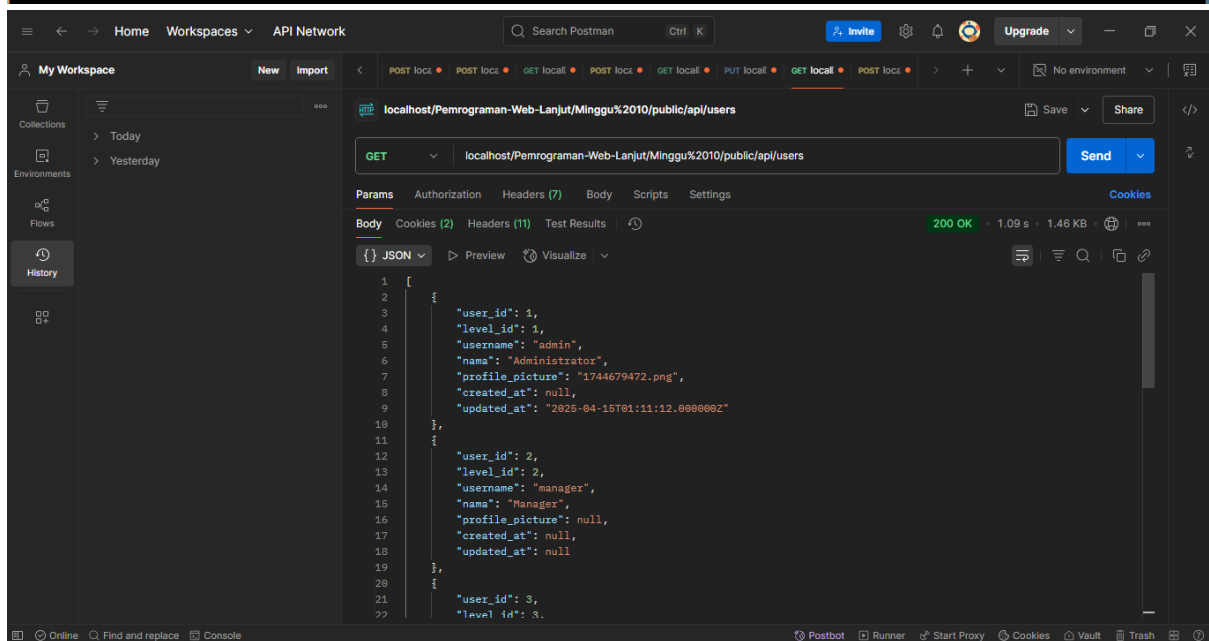
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10> php artisan make:controller Api/BarangController

INFO Controller [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10\app\Http\Controllers\Api\BarangController.php] created successfully.

PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10> php artisan make:controller Api/KategoriController

INFO Controller [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 10\app\Http\Controllers\Api\KategoriController.php] created successfully.

```

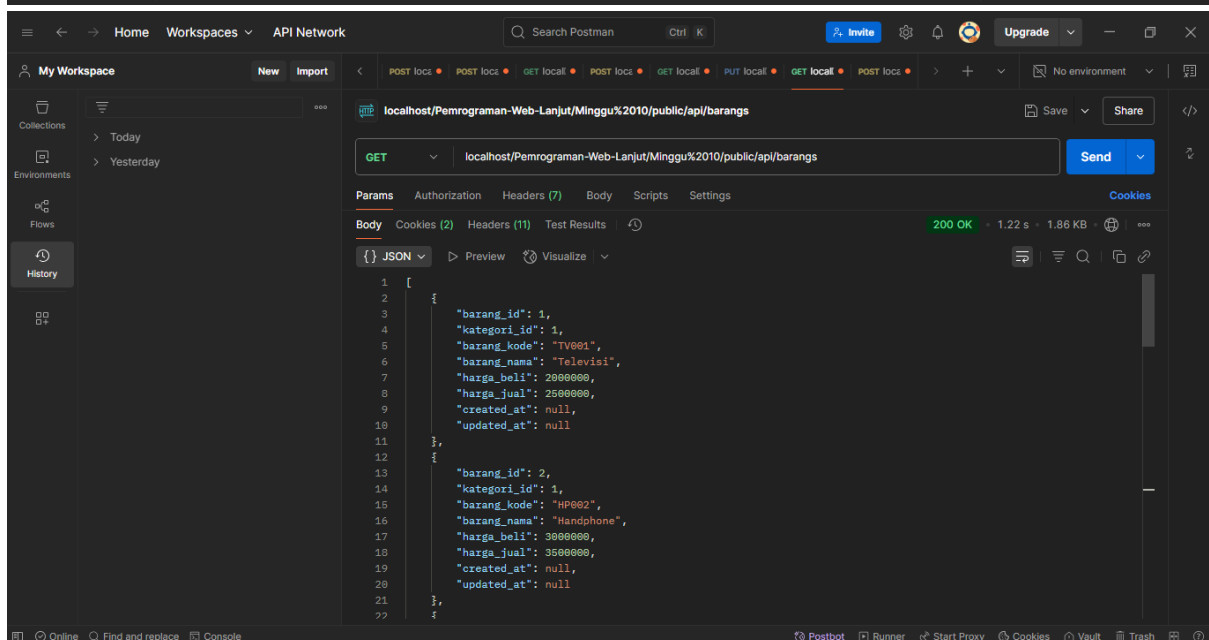


Postman interface showing a GET request to `localhost/Pemrograman-Web-Lanjut/Minggu%2010/public/api/users`. The response is a 200 OK status with a JSON array of three user objects.

```

[
  {
    "user_id": 1,
    "level_id": 1,
    "username": "admin",
    "name": "Administrator",
    "profile_picture": "1744679472.png",
    "created_at": null,
    "updated_at": "2025-04-15T01:11:12.000000Z"
  },
  {
    "user_id": 2,
    "level_id": 2,
    "username": "manager",
    "name": "Manager",
    "profile_picture": null,
    "created_at": null,
    "updated_at": null
  },
  {
    "user_id": 3,
    "level_id": 3
  }
]

```



Postman interface showing a GET request to `localhost/Pemrograman-Web-Lanjut/Minggu%2010/public/api/barangs`. The response is a 200 OK status with a JSON array of two barang objects.

```

[
  {
    "barang_id": 1,
    "kategori_id": 1,
    "barang_kode": "TV001",
    "barang_nama": "Televisi",
    "harga_beli": 2000000,
    "harga_jual": 2500000,
    "created_at": null,
    "updated_at": null
  },
  {
    "barang_id": 2,
    "kategori_id": 1,
    "barang_kode": "HP002",
    "barang_nama": "Handphone",
    "harga_beli": 3000000,
    "harga_jual": 3500000,
    "created_at": null,
    "updated_at": null
  }
]

```

HomeWorkspacesAPI Network

Search PostmanCtrl K

Invite

Upgrade

My Workspace

NewImport

Collections

Environments

Flows

History

localhost/Pemrograman-Web-Lanjut/Minggu%2010/public/api/kategoris

SaveShare

GETlocalhost/Pemrograman-Web-Lanjut/Minggu%2010/public/api/kategorisSend

ParamsAuthorizationHeaders (7)BodyScriptsSettingsCookies

BodyCookies (2)Headers (11)Test ResultsVisualize

```
1  [
2    {
3      "kategori_id": 1,
4      "kategori_kode": "EL",
5      "kategori_nama": "Elektronik",
6      "created_at": null,
7      "updated_at": null
8    },
9    {
10     "kategori_id": 2,
11     "kategori_kode": "PK",
12     "kategori_nama": "Pakaian",
13     "created_at": null,
14     "updated_at": null
15   },
16   {
17     "kategori_id": 3,
18     "kategori_kode": "MK",
19     "kategori_nama": "Makanan",
20     "created_at": null,
21     "updated_at": null
22   }
23 ]
```

OnlineFind and replaceConsole

PostbotRunnerStart ProxyCookiesVaultTrash