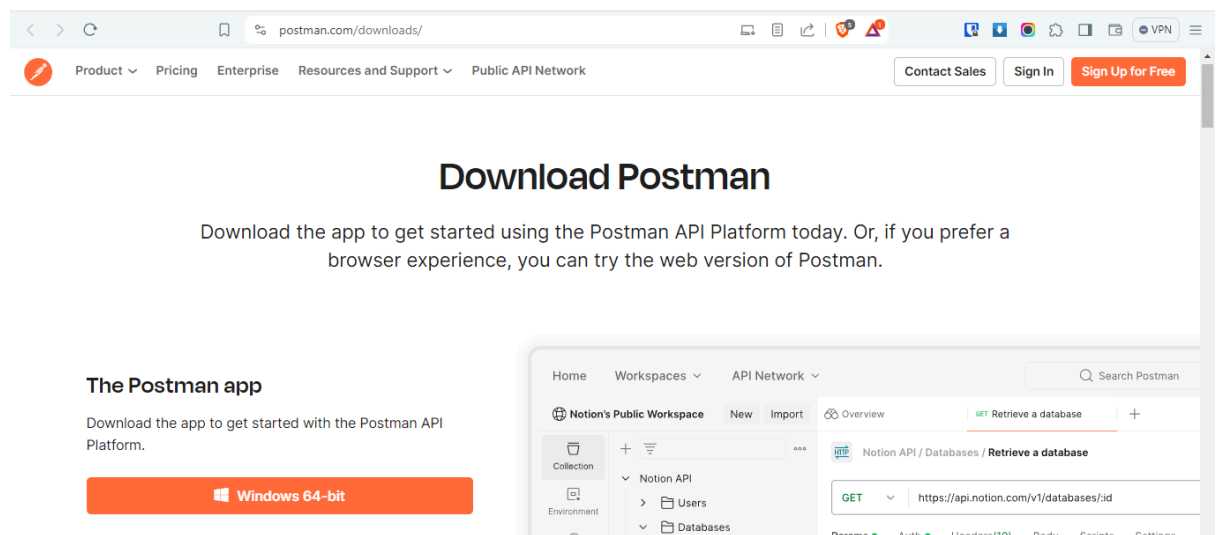


NIM : 2241720095
Nama : Ana Bellatus Mustaqfiro
Kelas : D4 TI - 2H
No. Urut : 04
Mata Kuliah : Pemrograman Web Lanjut Jobsheet 10 (REST API Bagian 1)

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.

```
PS C:\laragon\www\PWL_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 (2.2.0)
- Locking lcobucci/jwt (4.0.4)
- Locking stella-maris/clock (0.1.7)
- Locking tymon/jwt-auth (2.1.1)
```

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"

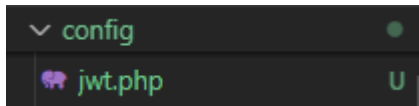
Which provider or tag's files would you like to publish?
All providers and tags ..... 0

> 9

[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 [oKBmUL63vDzUSRLQwvOo8j0Fn12nYyCgNW2tY5cMCTdWczpWrDAQXZyKYAVEWGHg] set successfully.

.env
61 JWT_SECRET=oKBmUL63vDzUSRLQwvOo8j0Fn12nYyCgNW2tY5cMCTdWczpWrDAQXZyKYAVEWGHg
```

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

```
config > auth.php

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

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

```
app > Models > UserModel.php > PHP Intelephense > UserModel
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\Relations\BelongsTo;
8 use Illuminate\Foundation\Auth\User as Authenticatable;
9 use Tymon\JWTAuth\Contracts\JWTSubject;
10
11 class UserModel extends Authenticatable implements JWTSubject
12 {
13     use HasFactory;
14     use Authenticatable;
15
16     public function getJWTIdentifier()
17     {
18         return $this->getKey();
19     }
20
21     public function getJWTCustomClaims()
22     {
23         return [];
24     }
25 }
```

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.

```
app > Http > Controllers > Api > RegisterController.php > ...
3 namespace App\Http\Controllers\Api;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\UserModel;
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         $validator = Validator::make($request->all(), [
15             'username' => 'required',
16             'nama' => 'required',
17             'password' => 'required|min:5|confirmed',
18             'level_id' => 'required'
19         ]);
20
21         if ($validator->fails()) {
22             return response()->json($validator->errors(), 422);
23         }
24     }
25 }
```

```

app > Http > Controllers > Api > RegisterController.php > ...
25     $user = UserModel::create([
26         'username' => $request->username,
27         'nama' => $request->nama,
28         'password' => bcrypt($request->password),
29         'level_id' => $request->level_id
30     ]);
31
32     if ($user) {
33         return response()->json([
34             'success' => true,
35             'user' => $user,
36         ], 201);
37     }
38
39     return response()->json([
40         'success' => false,
41     ], 409);
42 }
43 }

```

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

```

routes > api.php > ...
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.

POST localhost/PWL_POS/public/api/register

Send

Params Auth Headers (7) Body Scripts Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body 422 Unprocessable Content 1119 ms 569 B Save as example

Pretty Raw Preview Visualize JSON

```

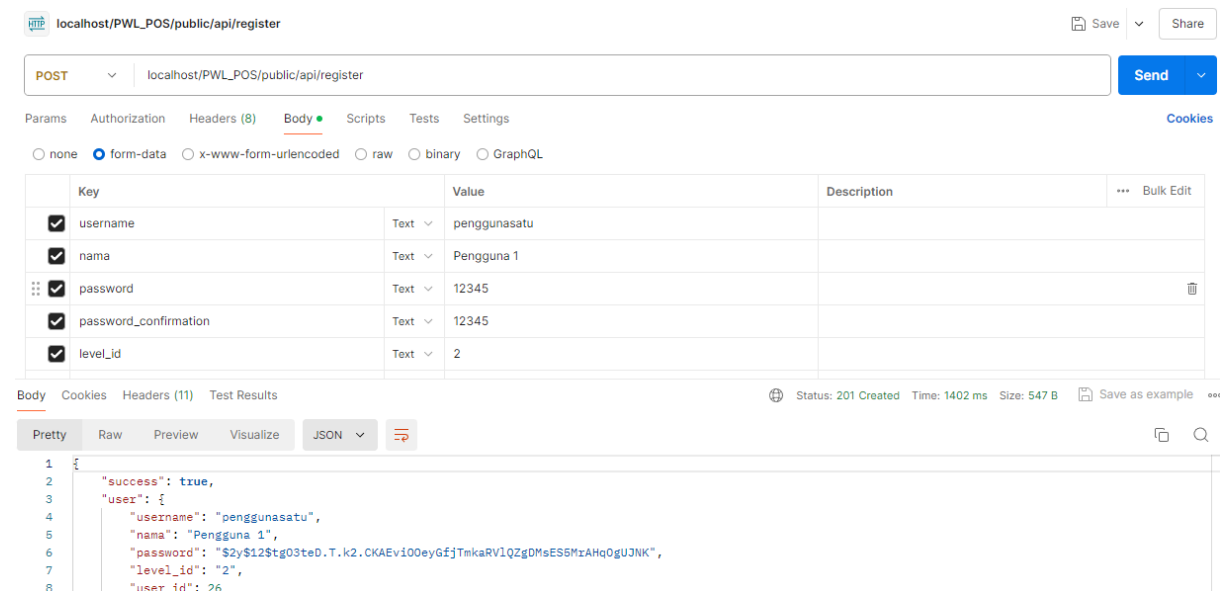
1 {
2   "username": [
3     "The username field is required."
4   ],
5   "nama": [
6     "The nama field is required."
7   ],
8   "password": [
9     "The password field is required."
10  ],
11  "level_id": [
12    "The level id field is required."
13  ]
14 }

```

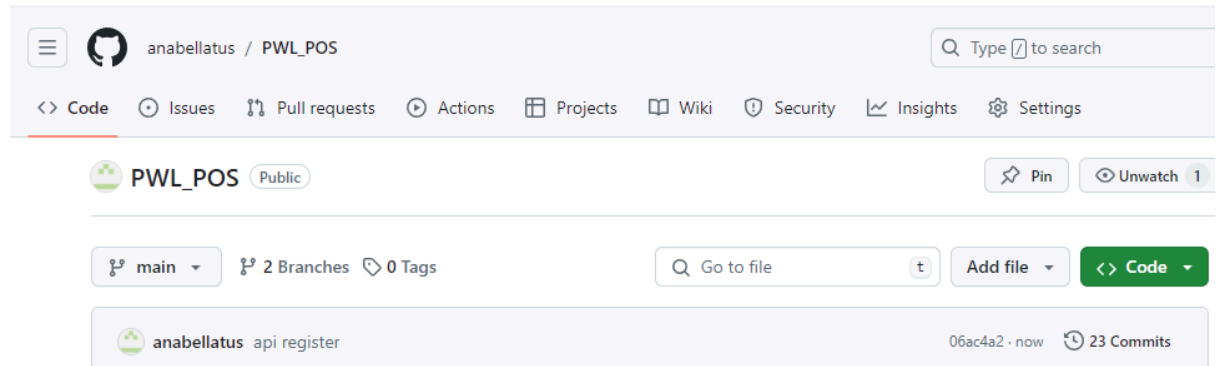
12. Sekarang kita coba masukkan data. Klik tab Body dan pilih form-data. Isikan key sesuai dengan kolom data, serta isikan data registrasi menggunakan nilai yang Anda inginkan

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

Lakukan percobaan yang sama dan berikan screenshot hasil percobaan Anda.



13. Lakukan commit perubahan file pada Github.



Praktikum 2 - Membuat RESTful API Login

1. Kita buat file controller dengan nama LoginController.

`php artisan make:controller Api/LoginController`

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

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

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

```
app > Http > Controllers > Api > LoginController.php > ...
5   use App\Http\Controllers\Controller;
6   use Illuminate\Http\Request;
7   use Illuminate\Support\Facades\Validator;
8
9   + class LoginController extends Controller
10  {
11      public function __invoke(Request $request)
12      {
13          $validator = Validator::make($request->all(), [
14              'username' => 'required',
15              'password' => 'required'
16          ]);
17
18          if ($validator->fails()) {
19              return response()->json($validator->errors(), 422);
20          }
21
22          $credentials = $request->only('username', 'password');
```

```
app > Http > Controllers > Api > LoginController.php > PHP Intelephense > LoginController > __invoke
24      if (!$token = auth()->guard('api')->attempt($credentials)) {
25          return response()->json([
26              'success' => false,
27              'message' => 'Username atau password salah'
28          ], 401);
29      }
30
31      return response()->json([
32          'success' => true,
33          'user' => auth()->guard('api')->user(),
34          'token' => $token,
35      ], 200);
36  }
37  }
```

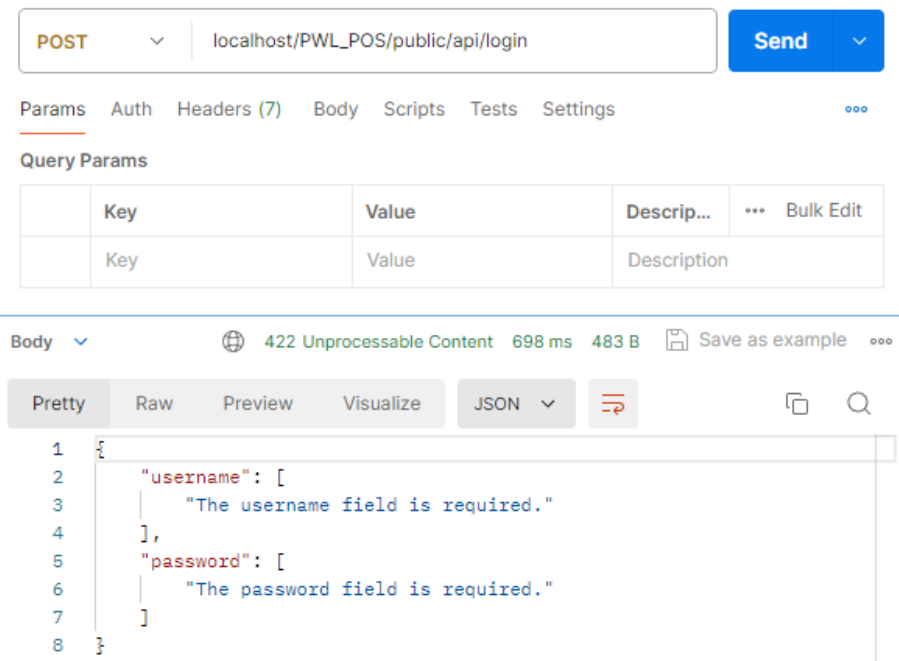
3. Berikutnya tambahkan route baru pada file api.php yaitu /login dan /user.

```
routes > api.php > ...
23  Route::post('/login', App\Http\Controllers\Api\LoginController::class)->name('login');
24  Route::middleware('auth:api')->get('/user', function (Request $request) {
25      return $request->user();
26  });
```

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.

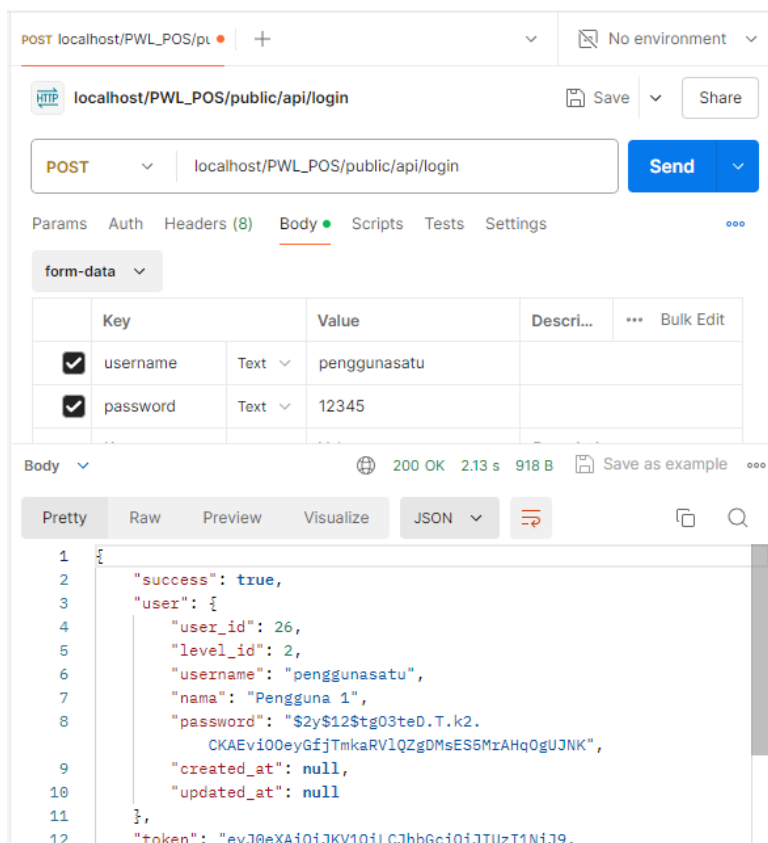
Jika berhasil akan muncul error validasi seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshot hasil percobaan Anda.

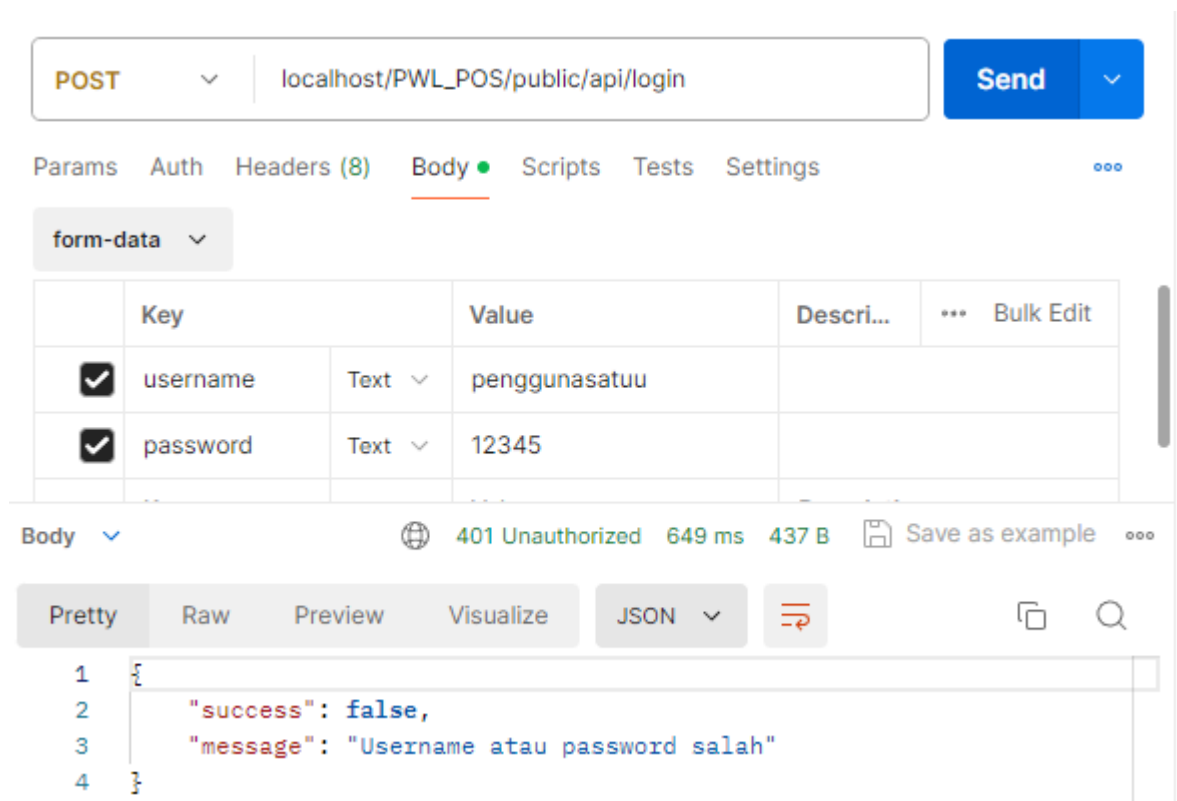


- Selanjutnya, isikan username dan password sesuai dengan data user yang ada pada database. Klik tab Body dan pilih form-data. Isikan key sesuai dengan kolom data, serta isikan data user. Klik tombol Send, jika berhasil maka akan keluar tampilan seperti berikut. Copy nilai token yang diperoleh pada saat login karena akan diperlukan pada saat logout.

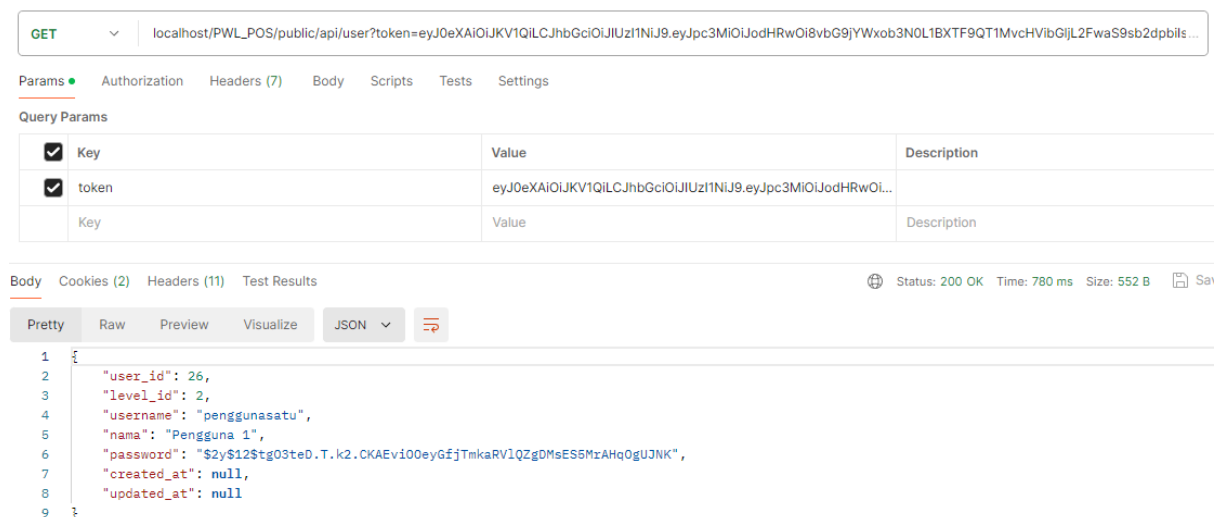
Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.



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

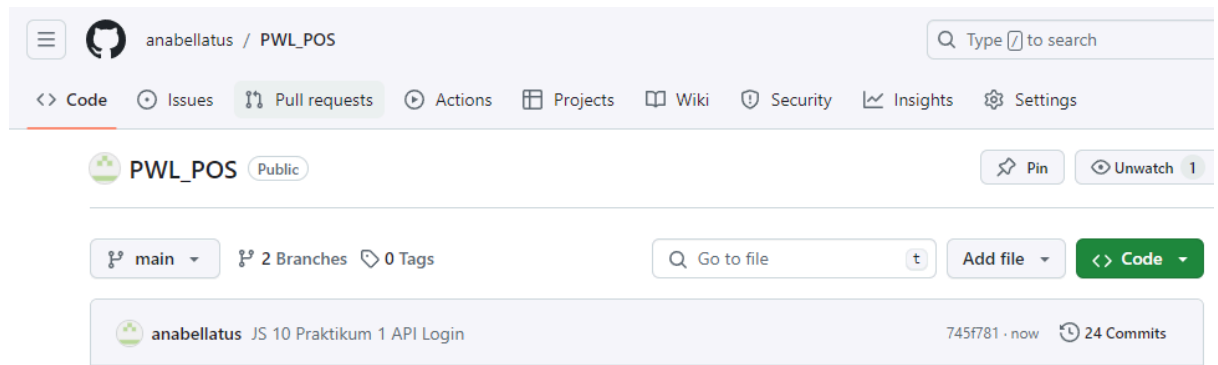


7. Coba kembali melakukan login dengan data yang benar. Sekarang mari kita coba menampilkan data user yang sedang login menggunakan URL `localhost/PWL_POS/public/api/user` dan method `GET`. Jelaskan hasil dari percobaan tersebut.



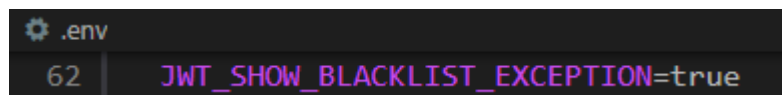
percobaan tersebut mengambil data user dengan method `GET` dan params berupa token yang didapat dari hasil login user sebelumnya, sehingga memberikan response data user yang telah login dengan format `JSON`

- Lakukan commit perubahan file pada Github.

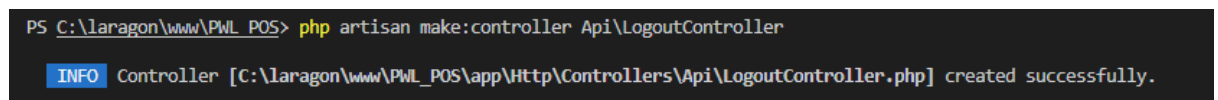


Praktikum 3 - Membuat RESTful API Logout

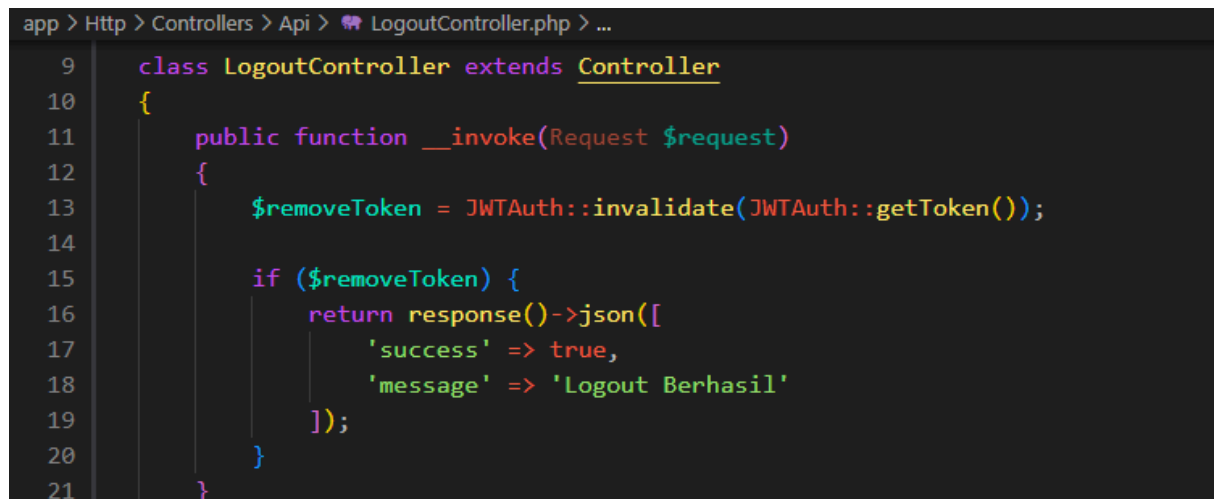
- Tambahkan kode berikut pada file .env
`JWT_SHOW_BLACKLIST_EXCEPTION=true`



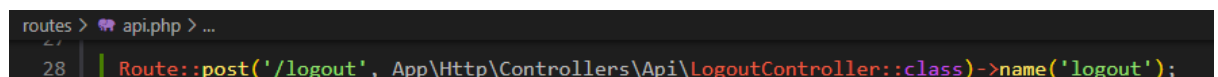
- Buat Controller baru dengan nama LogoutController.
`php artisan make:controller Api/LogoutController`



- Buka file tersebut dan ubah kode menjadi seperti berikut.

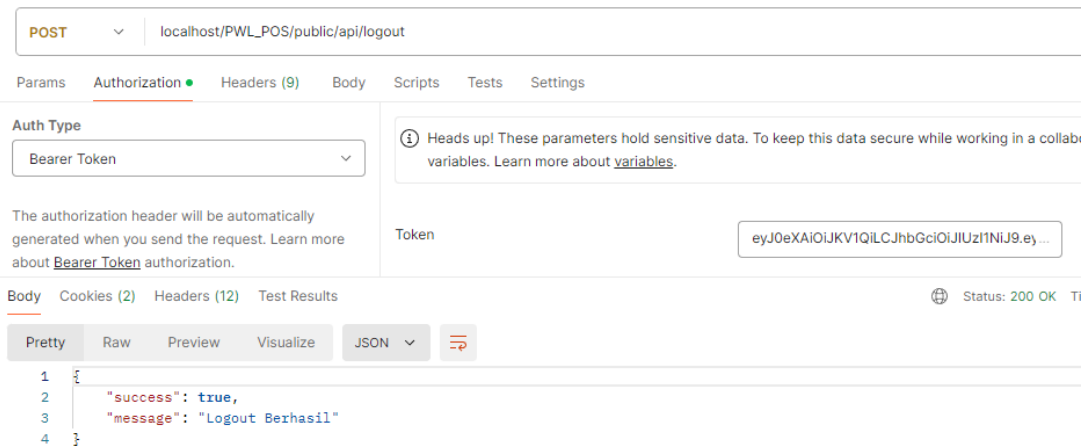


- Lalu kita tambahkan routes pada api.php

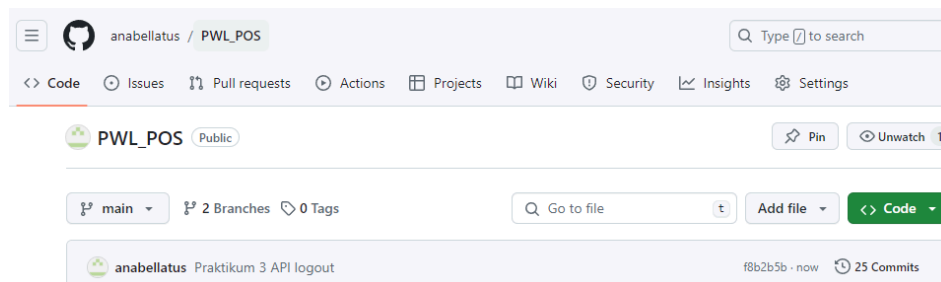


- Jika sudah, kita akan melakukan uji coba REST API melalui aplikasi Postman. Buka aplikasi Postman, isi URL localhost/PWL_POS/public/api/logout serta method POST.

- 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.



- Lakukan commit perubahan file pada Github.



Praktikum 4 - Implementasi CRUD dalam RESTful API

- 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.
```

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

```
app > Http > Controllers > Api > LevelController.php > PHP Intelephense > LevelController > destroy
9  class LevelController extends Controller
10 {
11     public function index()
12     {
13         return LevelModel::all();
14     }
15
16     public function store(Request $request)
17     {
18         $level = LevelModel::create($request->all());
19         return response()->json($level, 201);
20     }
21
22     public function show(LevelModel $level)
23     {
24         return LevelModel::find($level);
25     }
26 }
```

```

app > Http > Controllers > Api > LevelController.php > ...
27     public function update(Request $request, LevelModel $level)
28     {
29         $level->update($request->all());
30         return LevelModel::find($level);
31     }
32
33     public function destroy(LevelModel $level)
34     {
35         $level->delete();
36         return response()->json([
37             'success' => true,
38             'message' => 'Data terhapus',
39         ]);
40     }
41 }

```

3. Kemudian kita lengkapi routes pada api.php.

```

routes > api.php > ...
31 Route::get('levels', [LevelController::class, 'index']);
32 Route::post('levels', [LevelController::class, 'store']);
33 Route::get('levels/{level}', [LevelController::class, 'show']);
34 Route::put('levels/{level}', [LevelController::class, 'update']);
35 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 screenshoot hasil percobaan Anda.

GET localhost/PWL_POS-main/public/api/levels

Params Auth Headers (6) Body Scripts Tests Settings

Query Params

	Key	Value	Descri...
Body			

200 OK 1848 ms 780 B

Pretty Raw Preview Visualize JSON

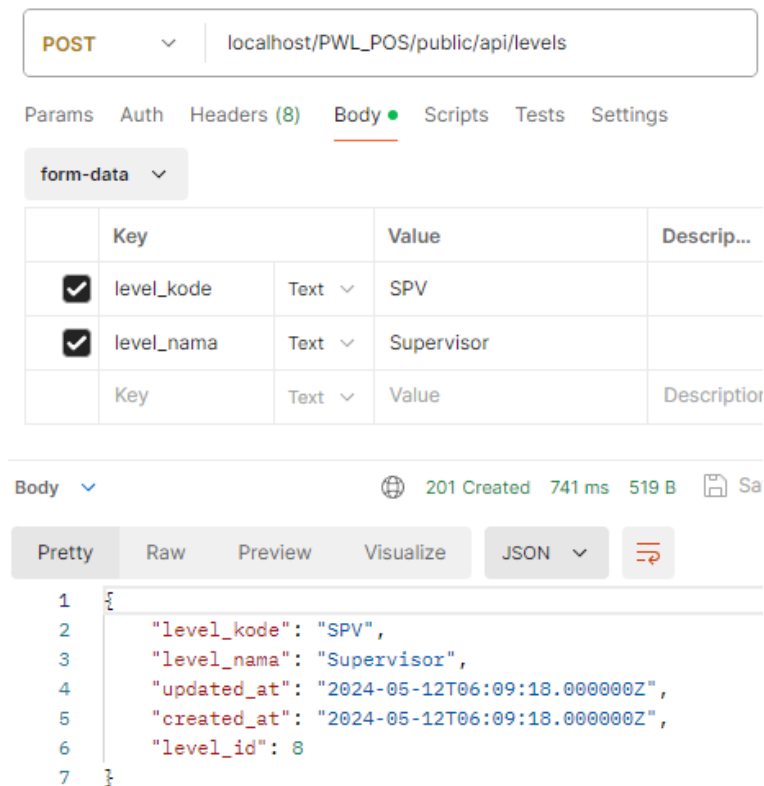
```

1 [
2   {
3     "level_id": 1,
4     "level_kode": "ADM",
5     "level_nama": "Administrator",
6     "created_at": null,
7     "updated_at": null
8   },
9   {
10    "level_id": 2,
11    "level_kode": "MNG",
12    "level_nama": "Manager",
13    "created_at": null,
14    "updated_at": null
15  }

```

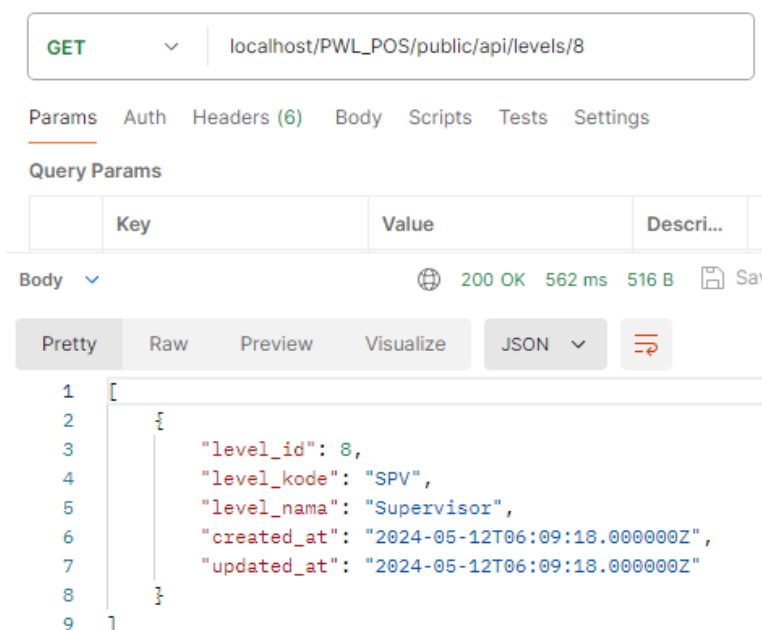
uji coba API untuk menampilkan data menggunakan method GET sehingga menghasilkan response semua data level berupa JSON

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.



penambahan data dilakukan dengan method POST dengan diberikan body berupa form-data yang berisi data yang diperlukan yaitu, level_kode dan level_nama, sehingga data tersimpan di database dan memberikan response berupa data level yang baru saja ditambahkan

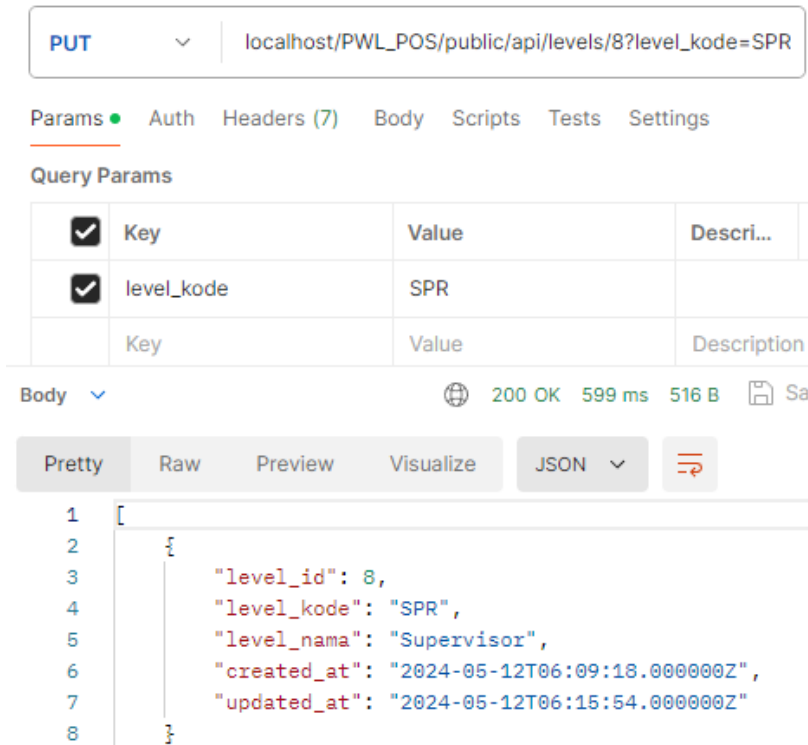
6. Berikutnya lakukan percobaan menampilkan detail data. Jelaskan dan berikan screenshot hasil percobaan Anda.



menampilkan data menggunakan method GET dengan dituliskan value id level yang ingin dilihat di akhir url, sehingga menampilkan detail data user dengan level_id 8

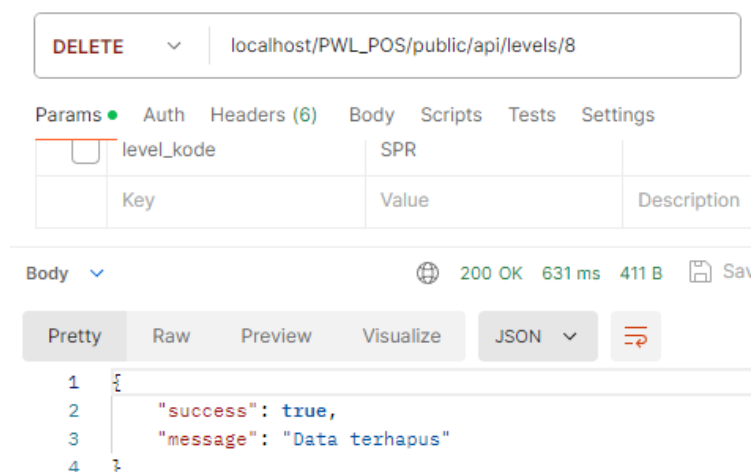
7. Jika sudah, kita coba untuk melakukan edit data menggunakan `localhost/PWL_POS/main/public/api/levels/{id}` dan method PUT. Isikan data yang ingin diubah pada tab Param.

Jelaskan dan berikan screenshot hasil percobaan Anda.



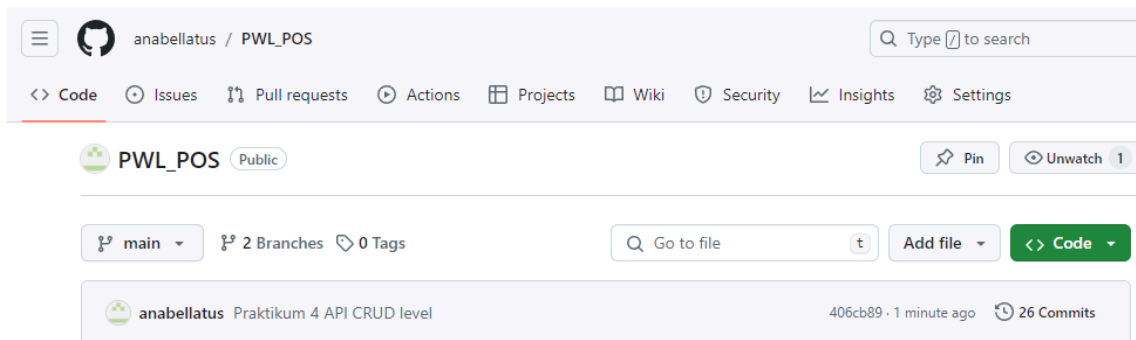
edit data dengan menggunakan method PUT dengan diberikan param berupa data yang ingin diubah yaitu level_kode menjadi SPR, sehingga data level pada database dengan id 8 level_kode nya berubah menjadi SPR, dan mengembalikan response berupa data level yang telah diperbarui

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



hapus data dengan method delete dengan dituliskan id level yang ingin dihapus pada akhir url, sehingga data level dengan id 8 akan terhapus di database dan mengembalikan response bahwa data telah terhapus

9. Lakukan commit perubahan file pada Github.



TUGAS

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

a. m_user

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

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

```
routes > api.php > ...
```

```
40 Route::get('users', [UserController::class, 'index']);
41 Route::post('users', [UserController::class, 'store']);
42 Route::get('users/{user}', [UserController::class, 'show']);
43 Route::put('users/{user}', [UserController::class, 'update']);
44 Route::delete('users/{user}', [UserController::class, 'destroy']);
```

```
app > Http > Controllers > Api > UserController.php > ...
```

```
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 class UserController extends Controller
10 {
11     public function index(){
12         return UserModel::all();
13     }
14
15     public function store(Request $request){
16         $user = UserModel::create([
17             'username' => $request->username,
18             'nama' => $request->nama,
19             'password' => bcrypt($request->password),
20             'level_id' => $request->level_id
21         ]);
22         return response()->json($user, 201);
23     }
24
25     public function show(UserModel $user){
26         return $user;
27     }
28 }
```

```
app > Http > Controllers > Api > UserController.php > ...
29     public function update(Request $request, UserModel $user){
30         $user->update($request->all());
31         return $user;
32     }
33
34     public function destroy(UserModel $user){
35         $user->delete();
36         return response()->json([
37             'success' => true,
38             'message' => 'Data terhapus',
39         ]);
40     }
41 }
```

GET localhost/PWL_POS/public/api/users

Params Authorization Headers (6) Body Scripts Tests Settings

Query Params

Key	Value	Description
-----	-------	-------------

Body Cookies Headers (11) Test Results Status: 200 OK Time: 705 ms Size: 2.67 KB

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "user_id": 1,
4     "level_id": 1,
5     "username": "admin",
6     "nama": "Administrator",
7     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
8     "created_at": null,
9     "updated_at": null
10  },
11  {
12     "user_id": 2,
13     "level_id": 2,
14     "username": "manager",
15     "nama": "Manager",
16     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
17     "created_at": null,
18     "updated_at": null
19  },
20  {
21     "user_id": 3,
22     "level_id": 3,
23     "username": "staff3",
24     "nama": "Staff 3",
25     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
26     "created_at": null,
27     "updated_at": null
28  },
29  {
30     "user_id": 4,
31     "level_id": 4,
32     "username": "staff4",
33     "nama": "Staff 4",
34     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
35     "created_at": null,
36     "updated_at": null
37  },
38  {
39     "user_id": 5,
40     "level_id": 5,
41     "username": "staff5",
42     "nama": "Staff 5",
43     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
44     "created_at": null,
45     "updated_at": null
46  },
47  {
48     "user_id": 6,
49     "level_id": 6,
50     "username": "staff6",
51     "nama": "Staff 6",
52     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
53     "created_at": null,
54     "updated_at": null
55  },
56  {
57     "user_id": 7,
58     "level_id": 7,
59     "username": "staff7",
60     "nama": "Staff 7",
61     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
62     "created_at": null,
63     "updated_at": null
64  },
65  {
66     "user_id": 8,
67     "level_id": 8,
68     "username": "staff8",
69     "nama": "Staff 8",
70     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
71     "created_at": null,
72     "updated_at": null
73  },
74  {
75     "user_id": 9,
76     "level_id": 9,
77     "username": "staff9",
78     "nama": "Staff 9",
79     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
80     "created_at": null,
81     "updated_at": null
82  },
83  {
84     "user_id": 10,
85     "level_id": 10,
86     "username": "staff10",
87     "nama": "Staff 10",
88     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
89     "created_at": null,
90     "updated_at": null
91  },
92  {
93     "user_id": 11,
94     "level_id": 11,
95     "username": "staff11",
96     "nama": "Staff 11",
97     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
98     "created_at": null,
99     "updated_at": null
100  },
101  {
102     "user_id": 12,
103     "level_id": 12,
104     "username": "staff12",
105     "nama": "Staff 12",
106     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
107     "created_at": null,
108     "updated_at": null
109  },
110  {
111     "user_id": 13,
112     "level_id": 13,
113     "username": "staff13",
114     "nama": "Staff 13",
115     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
116     "created_at": null,
117     "updated_at": null
118  },
119  {
120     "user_id": 14,
121     "level_id": 14,
122     "username": "staff14",
123     "nama": "Staff 14",
124     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
125     "created_at": null,
126     "updated_at": null
127  },
128  {
129     "user_id": 15,
130     "level_id": 15,
131     "username": "staff15",
132     "nama": "Staff 15",
133     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
134     "created_at": null,
135     "updated_at": null
136  },
137  {
138     "user_id": 16,
139     "level_id": 16,
140     "username": "staff16",
141     "nama": "Staff 16",
142     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
143     "created_at": null,
144     "updated_at": null
145  },
146  {
147     "user_id": 17,
148     "level_id": 17,
149     "username": "staff17",
150     "nama": "Staff 17",
151     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
152     "created_at": null,
153     "updated_at": null
154  },
155  {
156     "user_id": 18,
157     "level_id": 18,
158     "username": "staff18",
159     "nama": "Staff 18",
160     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
161     "created_at": null,
162     "updated_at": null
163  },
164  {
165     "user_id": 19,
166     "level_id": 19,
167     "username": "staff19",
168     "nama": "Staff 19",
169     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
170     "created_at": null,
171     "updated_at": null
172  },
173  {
174     "user_id": 20,
175     "level_id": 20,
176     "username": "staff20",
177     "nama": "Staff 20",
178     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
179     "created_at": null,
180     "updated_at": null
181  },
182  {
183     "user_id": 21,
184     "level_id": 21,
185     "username": "staff21",
186     "nama": "Staff 21",
187     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
188     "created_at": null,
189     "updated_at": null
190  },
191  {
192     "user_id": 22,
193     "level_id": 22,
194     "username": "staff22",
195     "nama": "Staff 22",
196     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
197     "created_at": null,
198     "updated_at": null
199  },
200  {
201     "user_id": 23,
202     "level_id": 23,
203     "username": "staff23",
204     "nama": "Staff 23",
205     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
206     "created_at": null,
207     "updated_at": null
208  },
209  {
210     "user_id": 24,
211     "level_id": 24,
212     "username": "staff24",
213     "nama": "Staff 24",
214     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
215     "created_at": null,
216     "updated_at": null
217  },
218  {
219     "user_id": 25,
220     "level_id": 25,
221     "username": "staff25",
222     "nama": "Staff 25",
223     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
224     "created_at": null,
225     "updated_at": null
226  },
227  {
228     "user_id": 26,
229     "level_id": 26,
230     "username": "staff26",
231     "nama": "Staff 26",
232     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
233     "created_at": null,
234     "updated_at": null
235  },
236  {
237     "user_id": 27,
238     "level_id": 27,
239     "username": "staff27",
240     "nama": "Staff 27",
241     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
242     "created_at": null,
243     "updated_at": null
244  },
245  {
246     "user_id": 28,
247     "level_id": 28,
248     "username": "staff28",
249     "nama": "Staff 28",
250     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
251     "created_at": null,
252     "updated_at": null
253  },
254  {
255     "user_id": 29,
256     "level_id": 29,
257     "username": "staff29",
258     "nama": "Staff 29",
259     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
260     "created_at": null,
261     "updated_at": null
262  },
263  {
264     "user_id": 30,
265     "level_id": 30,
266     "username": "staff30",
267     "nama": "Staff 30",
268     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
269     "created_at": null,
270     "updated_at": null
271  },
272  {
273     "user_id": 31,
274     "level_id": 31,
275     "username": "staff31",
276     "nama": "Staff 31",
277     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
278     "created_at": null,
279     "updated_at": null
280  },
281  {
282     "user_id": 32,
283     "level_id": 32,
284     "username": "staff32",
285     "nama": "Staff 32",
286     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
287     "created_at": null,
288     "updated_at": null
289  },
290  {
291     "user_id": 33,
292     "level_id": 33,
293     "username": "staff33",
294     "nama": "Staff 33",
295     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
296     "created_at": null,
297     "updated_at": null
298  },
299  {
300     "user_id": 34,
301     "level_id": 34,
302     "username": "staff34",
303     "nama": "Staff 34",
304     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
305     "created_at": null,
306     "updated_at": null
307  },
308  {
309     "user_id": 35,
310     "level_id": 35,
311     "username": "staff35",
312     "nama": "Staff 35",
313     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
314     "created_at": null,
315     "updated_at": null
316  },
317  {
318     "user_id": 36,
319     "level_id": 36,
320     "username": "staff36",
321     "nama": "Staff 36",
322     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
323     "created_at": null,
324     "updated_at": null
325  },
326  {
327     "user_id": 37,
328     "level_id": 37,
329     "username": "staff37",
330     "nama": "Staff 37",
331     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
332     "created_at": null,
333     "updated_at": null
334  },
335  {
336     "user_id": 38,
337     "level_id": 38,
338     "username": "staff38",
339     "nama": "Staff 38",
340     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
341     "created_at": null,
342     "updated_at": null
343  },
344  {
345     "user_id": 39,
346     "level_id": 39,
347     "username": "staff39",
348     "nama": "Staff 39",
349     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
350     "created_at": null,
351     "updated_at": null
352  },
353  {
354     "user_id": 40,
355     "level_id": 40,
356     "username": "staff40",
357     "nama": "Staff 40",
358     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
359     "created_at": null,
360     "updated_at": null
361  },
362  {
363     "user_id": 41,
364     "level_id": 41,
365     "username": "staff41",
366     "nama": "Staff 41",
367     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
368     "created_at": null,
369     "updated_at": null
370  },
371  {
372     "user_id": 42,
373     "level_id": 42,
374     "username": "staff42",
375     "nama": "Staff 42",
376     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
377     "created_at": null,
378     "updated_at": null
379  },
380  {
381     "user_id": 43,
382     "level_id": 43,
383     "username": "staff43",
384     "nama": "Staff 43",
385     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
386     "created_at": null,
387     "updated_at": null
388  },
389  {
390     "user_id": 44,
391     "level_id": 44,
392     "username": "staff44",
393     "nama": "Staff 44",
394     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
395     "created_at": null,
396     "updated_at": null
397  },
398  {
399     "user_id": 45,
400     "level_id": 45,
401     "username": "staff45",
402     "nama": "Staff 45",
403     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
404     "created_at": null,
405     "updated_at": null
406  },
407  {
408     "user_id": 46,
409     "level_id": 46,
410     "username": "staff46",
411     "nama": "Staff 46",
412     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
413     "created_at": null,
414     "updated_at": null
415  },
416  {
417     "user_id": 47,
418     "level_id": 47,
419     "username": "staff47",
420     "nama": "Staff 47",
421     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
422     "created_at": null,
423     "updated_at": null
424  },
425  {
426     "user_id": 48,
427     "level_id": 48,
428     "username": "staff48",
429     "nama": "Staff 48",
430     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
431     "created_at": null,
432     "updated_at": null
433  },
434  {
435     "user_id": 49,
436     "level_id": 49,
437     "username": "staff49",
438     "nama": "Staff 49",
439     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
440     "created_at": null,
441     "updated_at": null
442  },
443  {
444     "user_id": 50,
445     "level_id": 50,
446     "username": "staff50",
447     "nama": "Staff 50",
448     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
449     "created_at": null,
450     "updated_at": null
451  },
452  {
453     "user_id": 51,
454     "level_id": 51,
455     "username": "staff51",
456     "nama": "Staff 51",
457     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
458     "created_at": null,
459     "updated_at": null
460  },
461  {
462     "user_id": 52,
463     "level_id": 52,
464     "username": "staff52",
465     "nama": "Staff 52",
466     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
467     "created_at": null,
468     "updated_at": null
469  },
470  {
471     "user_id": 53,
472     "level_id": 53,
473     "username": "staff53",
474     "nama": "Staff 53",
475     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
476     "created_at": null,
477     "updated_at": null
478  },
479  {
480     "user_id": 54,
481     "level_id": 54,
482     "username": "staff54",
483     "nama": "Staff 54",
484     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
485     "created_at": null,
486     "updated_at": null
487  },
488  {
489     "user_id": 55,
490     "level_id": 55,
491     "username": "staff55",
492     "nama": "Staff 55",
493     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
494     "created_at": null,
495     "updated_at": null
496  },
497  {
498     "user_id": 56,
499     "level_id": 56,
500     "username": "staff56",
501     "nama": "Staff 56",
502     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
503     "created_at": null,
504     "updated_at": null
505  },
506  {
507     "user_id": 57,
508     "level_id": 57,
509     "username": "staff57",
510     "nama": "Staff 57",
511     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
512     "created_at": null,
513     "updated_at": null
514  },
515  {
516     "user_id": 58,
517     "level_id": 58,
518     "username": "staff58",
519     "nama": "Staff 58",
520     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
521     "created_at": null,
522     "updated_at": null
523  },
524  {
525     "user_id": 59,
526     "level_id": 59,
527     "username": "staff59",
528     "nama": "Staff 59",
529     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
530     "created_at": null,
531     "updated_at": null
532  },
533  {
534     "user_id": 60,
535     "level_id": 60,
536     "username": "staff60",
537     "nama": "Staff 60",
538     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
539     "created_at": null,
540     "updated_at": null
541  },
542  {
543     "user_id": 61,
544     "level_id": 61,
545     "username": "staff61",
546     "nama": "Staff 61",
547     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
548     "created_at": null,
549     "updated_at": null
550  },
551  {
552     "user_id": 62,
553     "level_id": 62,
554     "username": "staff62",
555     "nama": "Staff 62",
556     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
557     "created_at": null,
558     "updated_at": null
559  },
560  {
561     "user_id": 63,
562     "level_id": 63,
563     "username": "staff63",
564     "nama": "Staff 63",
565     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
566     "created_at": null,
567     "updated_at": null
568  },
569  {
570     "user_id": 64,
571     "level_id": 64,
572     "username": "staff64",
573     "nama": "Staff 64",
574     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
575     "created_at": null,
576     "updated_at": null
577  },
578  {
579     "user_id": 65,
580     "level_id": 65,
581     "username": "staff65",
582     "nama": "Staff 65",
583     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
584     "created_at": null,
585     "updated_at": null
586  },
587  {
588     "user_id": 66,
589     "level_id": 66,
590     "username": "staff66",
591     "nama": "Staff 66",
592     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
593     "created_at": null,
594     "updated_at": null
595  },
596  {
597     "user_id": 67,
598     "level_id": 67,
599     "username": "staff67",
600     "nama": "Staff 67",
601     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
602     "created_at": null,
603     "updated_at": null
604  },
605  {
606     "user_id": 68,
607     "level_id": 68,
608     "username": "staff68",
609     "nama": "Staff 68",
610     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
611     "created_at": null,
612     "updated_at": null
613  },
614  {
615     "user_id": 69,
616     "level_id": 69,
617     "username": "staff69",
618     "nama": "Staff 69",
619     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
620     "created_at": null,
621     "updated_at": null
622  },
623  {
624     "user_id": 70,
625     "level_id": 70,
626     "username": "staff70",
627     "nama": "Staff 70",
628     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
629     "created_at": null,
630     "updated_at": null
631  },
632  {
633     "user_id": 71,
634     "level_id": 71,
635     "username": "staff71",
636     "nama": "Staff 71",
637     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
638     "created_at": null,
639     "updated_at": null
640  },
641  {
642     "user_id": 72,
643     "level_id": 72,
644     "username": "staff72",
645     "nama": "Staff 72",
646     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
647     "created_at": null,
648     "updated_at": null
649  },
650  {
651     "user_id": 73,
652     "level_id": 73,
653     "username": "staff73",
654     "nama": "Staff 73",
655     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
656     "created_at": null,
657     "updated_at": null
658  },
659  {
660     "user_id": 74,
661     "level_id": 74,
662     "username": "staff74",
663     "nama": "Staff 74",
664     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
665     "created_at": null,
666     "updated_at": null
667  },
668  {
669     "user_id": 75,
670     "level_id": 75,
671     "username": "staff75",
672     "nama": "Staff 75",
673     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
674     "created_at": null,
675     "updated_at": null
676  },
677  {
678     "user_id": 76,
679     "level_id": 76,
680     "username": "staff76",
681     "nama": "Staff 76",
682     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
683     "created_at": null,
684     "updated_at": null
685  },
686  {
687     "user_id": 77,
688     "level_id": 77,
689     "username": "staff77",
690     "nama": "Staff 77",
691     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
692     "created_at": null,
693     "updated_at": null
694  },
695  {
696     "user_id": 78,
697     "level_id": 78,
698     "username": "staff78",
699     "nama": "Staff 78",
700     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
701     "created_at": null,
702     "updated_at": null
703  },
704  {
705     "user_id": 79,
706     "level_id": 79,
707     "username": "staff79",
708     "nama": "Staff 79",
709     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
710     "created_at": null,
711     "updated_at": null
712  },
713  {
714     "user_id": 80,
715     "level_id": 80,
716     "username": "staff80",
717     "nama": "Staff 80",
718     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
719     "created_at": null,
720     "updated_at": null
721  },
722  {
723     "user_id": 81,
724     "level_id": 81,
725     "username": "staff81",
726     "nama": "Staff 81",
727     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
728     "created_at": null,
729     "updated_at": null
730  },
731  {
732     "user_id": 82,
733     "level_id": 82,
734     "username": "staff82",
735     "nama": "Staff 82",
736     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
737     "created_at": null,
738     "updated_at": null
739  },
740  {
741     "user_id": 83,
742     "level_id": 83,
743     "username": "staff83",
744     "nama": "Staff 83",
745     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
746     "created_at": null,
747     "updated_at": null
748  },
749  {
750     "user_id": 84,
751     "level_id": 84,
752     "username": "staff84",
753     "nama": "Staff 84",
754     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
755     "created_at": null,
756     "updated_at": null
757  },
758  {
759     "user_id": 85,
760     "level_id": 85,
761     "username": "staff85",
762     "nama": "Staff 85",
763     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
764     "created_at": null,
765     "updated_at": null
766  },
767  {
768     "user_id": 86,
769     "level_id": 86,
770     "username": "staff86",
771     "nama": "Staff 86",
772     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcxss3iwtUY3DSA/M/Ku87qY9hf6",
773     "created_at": null,
774     "updated_at": null
775  },
776  {
777     "user_id": 87,
778     "level_id": 87,
779     "username": "staff87",
780     "nama": "Staff 87",
781     "password": "$2y$12$G0ZleXmDBBddPP36GNZpe2RjLcx
```

GET
localhost/PWL_POS/public/api/users/29

Params
Authorization
Headers (6)
Body
Scripts
Tests
Settings

Query Params

	Key	Value	Description
--	-----	-------	-------------

Body
Cookies
Headers (11)
Test Results
Status: 200 OK
Time: 638 ms
Size: 543 B
Sa

Pretty
Raw
Preview
Visualize
JSON

```

1 {
2   "user_id": 29,
3   "level_id": 3,
4   "username": "staff3",
5   "nama": "Staff 3",
6   "password": "$2y$12$bpPCTgTAuyGQeJ60YQvXXKuIN.2fiW3Kt2J4chttG0Zg3tcfruicyi",
7   "created_at": null,
8   "updated_at": null
9 }

```

PUT
localhost/PWL_POS/public/api/users/29?nama=Staff Tiga

Params
Authorization
Headers (7)
Body
Scripts
Tests
Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description
<input checked="" type="checkbox"/>	nama	Staff Tiga	
	Key	Value	Description

Body
Cookies
Headers (11)
Test Results
Status: 200 OK
Time: 611 ms
Size: 546 B
Sa

Pretty
Raw
Preview
Visualize
JSON

```

1 {
2   "user_id": 29,
3   "level_id": 3,
4   "username": "staff3",
5   "nama": "Staff Tiga",
6   "password": "$2y$12$bpPCTgTAuyGQeJ60YQvXXKuIN.2fiW3Kt2J4chttG0Zg3tcfruicyi",
7   "created_at": null,
8   "updated_at": null
9 }

```

DELETE
localhost/PWL_POS/public/api/users/29

Params
Authorization
Headers (6)
Body
Scripts
Tests
Settings

Body
Cookies
Headers (11)
Test Results
Status: 200 OK
Time: 611 ms
Size: 411 B

Pretty
Raw
Preview
Visualize
JSON

```

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

```

b. m_kategori

```

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

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

```



```

routes > api.php > ...
46 Route::get('kategoris', [KategoriController::class, 'index']);
47 Route::post('kategoris', [KategoriController::class, 'store']);
48 Route::get('kategoris/{kategori}', [KategoriController::class, 'show']);
49 Route::put('kategoris/{kategori}', [KategoriController::class, 'update']);
50 Route::delete('kategoris/{kategori}', [KategoriController::class, 'destroy']);

```

```

app > Http > Controllers > Api > KategoriController.php > ...
9 class KategoriController extends Controller
10 {
11     public function index()
12     {
13         return KategoriModel::all();
14     }
15
16     public function store(Request $request)
17     {
18         $kategori = KategoriModel::create($request->all());
19         return response()->json($kategori, 201);
20     }
21
22     public function show(KategoriModel $kategori)
23     {
24         return KategoriModel::find($kategori);
25     }

```

```

app > Http > Controllers > Api > KategoriController.php > ...
27 public function update(Request $request, KategoriModel $kategori)
28 {
29     $kategori->update($request->all());
30     return KategoriModel::find($kategori);
31 }
32
33 public function destroy(KategoriModel $kategori)
34 {
35     $kategori->delete();
36     return response()->json([
37         'success' => true,
38         'message' => 'Data terhapus',
39     ]);
40 }
41 }

```

GET

localhost/PWL_POS/public/api/kategoris

Params

Authorization

Headers (6)

Body

Scripts

Tests

Settings

Query Params

Key	Value	Description

Body

Cookies

Headers (11)

Test Results

Status: 200 OK

Time: 590 ms

Size: 1.19 KB

Save

Pretty

Raw

Preview

Visualize

JSON

Expand

```

1 [
2   {
3     "kategori_id": 1,
4     "kategori_kode": "MKB",
5     "kategori_nama": "Makanan Berat",
6     "created_at": null,
7     "updated_at": "2024-04-02T21:12:05.000000Z"
8   },
9   {
10    "kategori_id": 2,
11    "kategori_kode": "MNM",
12    "kategori_nama": "Minuman",

```

POSTlocalhost/PWL_POS/public/api/kategoris

ParamsAuthorizationHeaders (8)BodyScriptsTestsSettings

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

	Key		Value	Description
<input checked="" type="checkbox"/>	kategori_kode	Text	SCR	
<input checked="" type="checkbox"/>	kategori_nama	Text	Skin Care	
	Key	Text	Value	Description

BodyCookiesHeaders (11)Test Results

Status: 201 CreatedTime: 705 msSize: 528 B

PrettyRawPreviewVisualizeJSON

```
1 {
2   "kategori_kode": "SCR",
3   "kategori_nama": "Skin Care",
4   "updated_at": "2024-05-12T09:28:05.000000Z",
5   "created_at": "2024-05-12T09:28:05.000000Z",
6   "kategori_id": 15
7 }
```

GETlocalhost/PWL_POS/public/api/kategoris/15

ParamsAuthorizationHeaders (6)BodyScriptsTestsSettings

Query Params

	Key	Value	Description
--	-----	-------	-------------

BodyCookiesHeaders (11)Test Results

Status: 200 OKTime: 655 msSize: 525 B

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "kategori_id": 15,
4     "kategori_kode": "SCR",
5     "kategori_nama": "Skin Care",
6     "created_at": "2024-05-12T09:28:05.000000Z",
7     "updated_at": "2024-05-12T09:28:05.000000Z"
8   }
9 ]
```

PUTlocalhost/PWL_POS/public/api/kategoris/15?kategori_nama=Skincare

ParamsAuthorizationHeaders (7)BodyScriptsTestsSettings

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description
<input checked="" type="checkbox"/>	kategori_nama	Skincare	
	Key	Value	Description

BodyCookiesHeaders (11)Test Results

Status: 200 OKTime: 598 msSize: 524 B

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "kategori_id": 15,
4     "kategori_kode": "SCR",
5     "kategori_nama": "Skincare",
6     "created_at": "2024-05-12T09:28:05.000000Z",
7     "updated_at": "2024-05-12T09:44:39.000000Z"
8   }
9 ]
```

DELETE ▼ localhost/PWL_POS/public/api/kategoris/15

Params Authorization Headers (6) Body Scripts Tests Settings

Query Params

	Key	Value	Description
Body			

Body Cookies Headers (11) Test Results Status: 200 OK Time: 617 ms Size: 411 B

Pretty Raw Preview Visualize JSON ▼

```

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

```

c. m_barang

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

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

```
routes > api.php > ...
```

```

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

```

```
app > Http > Controllers > Api > BarangController.php > ...
```

```

9 class BarangController extends Controller
10 {
11     public function index()
12     {
13         return BarangModel::all();
14     }
15
16     public function store(Request $request)
17     {
18         $barang = BarangModel::create($request->all());
19         return response()->json($barang, 201);
20     }
21
22     public function show(BarangModel $barang)
23     {
24         return BarangModel::find($barang);
25     }

```

```
app > Http > Controllers > Api > BarangController.php > ...
```

```

27     public function update(Request $request, BarangModel $barang)
28     {
29         $barang->update($request->all());
30         return BarangModel::find($barang);
31     }
32
33     public function destroy(BarangModel $barang)
34     {
35         $barang->delete();
36         return response()->json([
37             'success' => true,
38             'message' => 'Data terhapus',
39         ]);
40     }
41 }

```

GET

localhost/PWL_POS/public/api/barangs

Params

Authorization

Headers (6)

Body

Scripts

Tests

Settings

Query Params

	Key	Value	Description
--	-----	-------	-------------

Body

Cookies

Headers (11)

Test Results

⌚

Status: 200 OK

Time: 741 ms

Size: 2.07 KB

📄

Sa

Pretty

Raw

Preview

Visualize

JSON

⌵

⌵

```
1 [
2   {
3     "barang_id": 1,
4     "kategori_id": 1,
5     "barang_kode": "MKN001",
6     "barang_nama": "Sari Roti",
7     "harga_beli": 3000,
8     "harga_jual": 5000,
9     "created_at": null,
10    "updated_at": null
11  },
12  {
13    "barang_id": 2,
14    "kategori_id": 1,
15    "barang_kode": "MKN002",
```

POST

localhost/PWL_POS/public/api/barangs

Params

Authorization

Headers (8)

Body

Scripts

Tests

Settings

☐ none

☒ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

	Key		Value	Description
<input checked="" type="checkbox"/>	kategori_id	Text	11	
<input checked="" type="checkbox"/>	barang_kode	Text	SPT001	
<input checked="" type="checkbox"/>	barang_nama	Text	Adidass Run	
<input checked="" type="checkbox"/>	harga_beli	Text	300000	
<input checked="" type="checkbox"/>	harga_jual	Text	350000	

Body

Cookies

Headers (11)

Test Results

⌚

Status: 201 Created

Time: 602 ms

Size: 590 B

📄

Sa

Pretty

Raw

Preview

Visualize

JSON

⌵

⌵

```
2   "kategori_id": "11",
3   "barang_kode": "SPT001",
4   "barang_nama": "Adidass Run",
5   "harga_beli": "300000",
6   "harga_jual": "350000",
7   "updated_at": "2024-05-12T09:52:43.000000Z",
8   "created_at": "2024-05-12T09:52:43.000000Z",
9   "barang id": 14
```

GETlocalhost/PWL_POS/public/api/barangs/14

ParamsAuthorizationHeaders (6)BodyScriptsTestsSettings

Query Params

	Key	Value	Description
--	-----	-------	-------------

BodyCookiesHeaders (11)Test Results

Status: 200 OKTime: 766 msSize: 581 B

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "barang_id": 14,
4     "kategori_id": 11,
5     "barang_kode": "SPT001",
6     "barang_nama": "Adidass Run",
7     "harga_beli": 300000,
8     "harga_jual": 350000,
9     "created_at": "2024-05-12T09:52:43.000000Z",
10    "updated_at": "2024-05-12T09:52:43.000000Z"
11  }
```

PUTlocalhost/PWL_POS/public/api/barangs/14?barang_nama=Adidas Running

ParamsAuthorizationHeaders (7)BodyScriptsTestsSettings

Query Params

	Key	Value	Description
<input checked="" type="checkbox"/>	Key		
<input checked="" type="checkbox"/>	barang_nama	Adidas Running	

BodyCookiesHeaders (11)Test Results

Status: 200 OKTime: 608 msSize: 584 B

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "barang_id": 14,
4     "kategori_id": 11,
5     "barang_kode": "SPT001",
6     "barang_nama": "Adidas Running",
7     "harga_beli": 300000,
8     "harga_jual": 350000,
9     "created_at": "2024-05-12T09:52:43.000000Z",
10    "updated_at": "2024-05-12T09:55:09.000000Z"
11  }
```

DELETElocalhost/PWL_POS/public/api/barangs/14

ParamsAuthorizationHeaders (6)BodyScriptsTestsSettings

Query Params

	Key	Value	Description
--	-----	-------	-------------

BodyCookiesHeaders (11)Test Results

Status: 200 OKTime: 759 msSize: 411 B

PrettyRawPreviewVisualizeJSON

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