

[illegible]

```
Terminal  Help  api.php - rental_mobil - Visual Studio Code

petugas.php  PetugasController.php app\Http\Controllers  api.php  PetugasController.php C:\...laundry\...

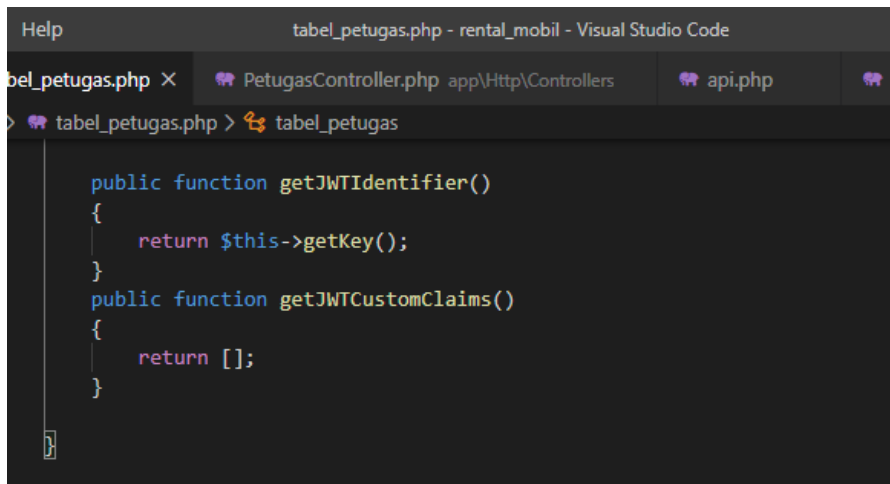
routes > api.php
1  <?php
2
3  use Illuminate\Http\Request;
4
5  /*
6  |-----
7  | API Routes
8  |-----
9  |
10 | Here is where you can register API routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | is assigned the "api" middleware group. Enjoy building your API!
13 |
14 | */
15
16 Route::middleware('auth:api')->get('/user', function (Request $request) {
17     return $request->user();
18 });
19
20 Route::post('register', 'PetugasController@register');
21 Route::post('login', 'PetugasController@login');
22
23 |
```

Model Tabel Petugas

```
terminal  Help  tabel_petugas.php - rental_mobil - Visual Studio Code

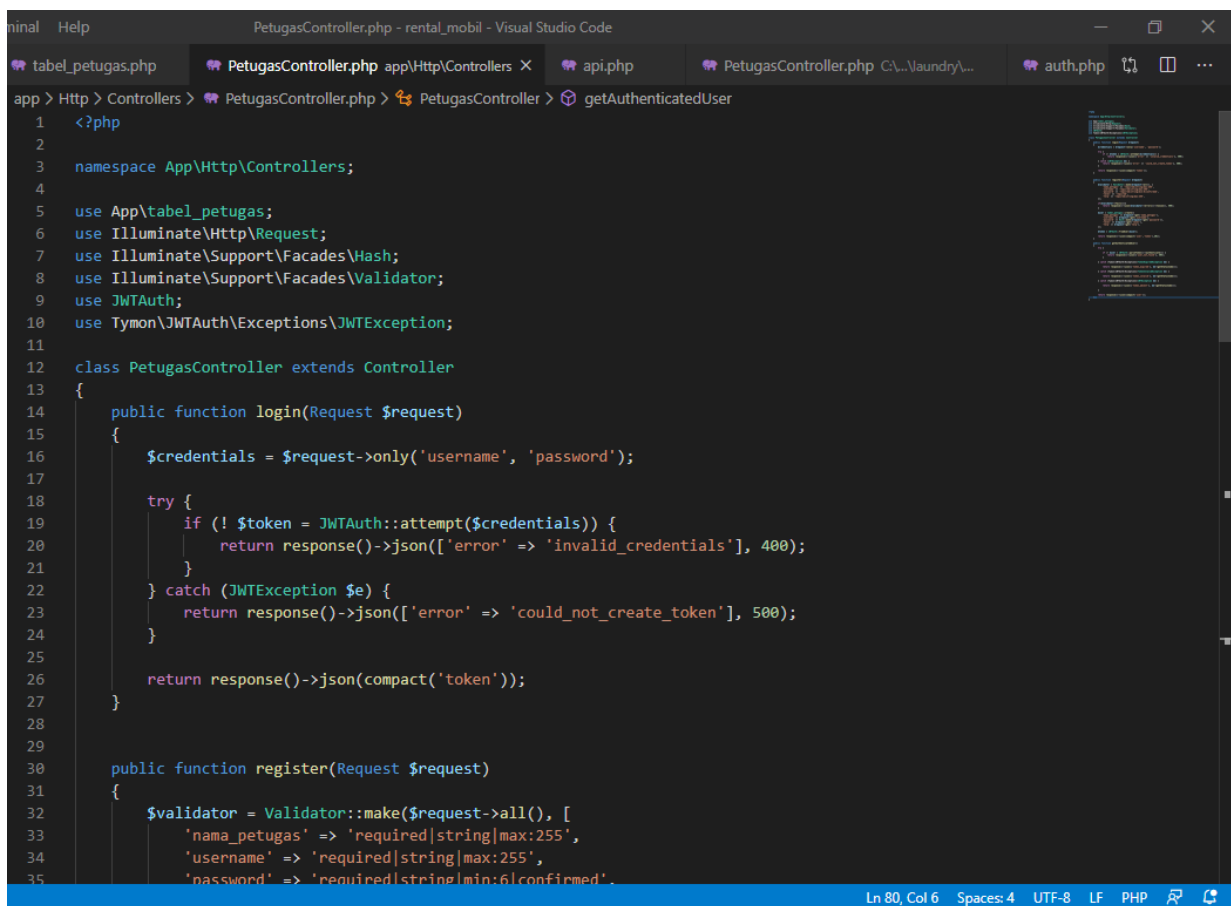
tabel_petugas.php  PetugasController.php app\Http\Controllers  api.php  PetugasController.php C:\...laundry\...  auth.php  ...

app > tabel_petugas.php > tabel_petugas
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6  use Illuminate\Notifications\Notifiable;
7  use Illuminate\Contracts\Auth\MustVerifyEmail;
8  use Illuminate\Foundation\Auth\User as Authenticatable;
9  use Tymon\JWTAuth\Contracts\JWTSubject;
10
11
12 class tabel_petugas extends Authenticatable implements JWTSubject
13 {
14     use Notifiable;
15     protected $table = 'petugas';
16     protected $primaryKey = 'id';
17     /**
18      * The attributes that are mass assignable.
19      *
20      * @var array
21      */
22     protected $fillable = [
23         'nama_petugas', 'username', 'password', 'level', 'telp'
24     ];
25
26     /**
27      * The attributes that should be hidden for arrays.
28      *
29      * @var array
30      */
31     protected $hidden = [
32         'password', 'remember_token',
33     ];
34
35     public function getJWTIdentifier()
```



```
public function getJWTIdentifier()
{
    return $this->getKey();
}
public function getJWTCustomClaims()
{
    return [];
}
```

Controller Petugas



```
<?php
namespace App\Http\Controllers;

use App\tabel_petugas;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use JWTAuth;
use Tymon\JWTAuth\Exceptions\JWTException;

class PetugasController extends Controller
{
    public function login(Request $request)
    {
        $credentials = $request->only('username', 'password');

        try {
            if (! $token = JWTAuth::attempt($credentials)) {
                return response()->json(['error' => 'invalid_credentials'], 400);
            }
        } catch (JWTException $e) {
            return response()->json(['error' => 'could_not_create_token'], 500);
        }

        return response()->json(compact('token'));
    }

    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'nama_petugas' => 'required|string|max:255',
            'username' => 'required|string|max:255',
            'password' => 'required|string|min:6|confirmed',
        ]);
    }
}
```

```
inal Help PetugasController.php - rental_mobil - Visual Studio Code
tabel_petugas.php PetugasController.php app\Http\Controllers X api.php PetugasController.php C:\...laundry\... auth.php ...
app > Http > Controllers > PetugasController.php > PetugasController > getAuthenticatedUser
29
30 public function register(Request $request)
31 {
32     $validator = Validator::make($request->all(), [
33         'nama_petugas' => 'required|string|max:255',
34         'username' => 'required|string|max:255',
35         'password' => 'required|string|min:6|confirmed',
36         'level' => 'required',
37         'telp' => 'required|string|max:255',
38     ]);
39
40     if($validator->fails()){
41         return response()->json($validator->errors()->toJson(), 400);
42     }
43
44     $user = tabel_petugas::create([
45         'nama_petugas' => $request->get('nama_petugas'),
46         'username' => $request->get('username'),
47         'password' => Hash::make($request->get('password')),
48         'level' => $request->get('level'),
49         'telp' => $request->get('telp'),
50     ]);
51
52     $token = JWTAuth::fromUser($user);
53
54     return response()->json(compact('user','token'),201);
55 }
56
57 public function getAuthenticatedUser()
58 {
59     try {
60
61         if (! $user = JWTAuth::parseToken()->authenticate()) {
62             return response()->json(['user_not_found'], 404);
63         }
64     }
65 }
```

Ln 80, Col 6 Spaces: 4 UTF-8 LF PHP

```
inal Help PetugasController.php - rental_mobil - Visual Studio Code
tabel_petugas.php PetugasController.php app\Http\Controllers X api.php PetugasController.php C:\...laundry\... auth.php ...
app > Http > Controllers > PetugasController.php > PetugasController > getAuthenticatedUser
57 public function getAuthenticatedUser()
58 {
59     try {
60
61         if (! $user = JWTAuth::parseToken()->authenticate()) {
62             return response()->json(['user_not_found'], 404);
63         }
64
65     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
66
67         return response()->json(['token_expired'], $e->getStatusCode());
68
69     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
70
71         return response()->json(['token_invalid'], $e->getStatusCode());
72
73     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
74
75         return response()->json(['token_absent'], $e->getStatusCode());
76
77     }
78
79     return response()->json(compact('user'));
80 }
81
82 }
```