

REGISTER LOGIN RENTAL MOBIL

1. API Register Login

```
20 Route::post('register', 'PetugasController@register');
21 Route::post('login', 'PetugasController@login');
22 Route::get('/', function() {
23     return Auth::user()->level;
24 }->middleware('jwt.verify');
```

2. Model

```
1  <?php
2
3  namespace App;
4
5  use Illuminate\Notifications\Notifiable;
6  use Illuminate\Contracts\Auth\MustVerifyEmail;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Tymon\JWTAuth\Contracts\JWTSubject;
9
10 class User extends Authenticatable implements JWTSubject
11 {
12     use Notifiable;
13     protected $table = 'petugas';
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var array
19      */
20     protected $fillable = [
21         'nama_petugas', 'username', 'password', 'level',
22     ];
23
24     /**
25      * The attributes that should be hidden for arrays.
26      *
27      * @var array
28      */
29     protected $hidden = [
30         'password', 'remember_token',
31     ];
32
33     public function getJWTIdentifier()
34     {
35         return $this->getKey();
36     }
37     public function getJWTCustomClaims()
38     {
39         return [];
40     }
41 }
```

3. Controller

```
5 use App\User;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Support\Facades\Validator;
9 use JWTAuth;
10 use Tymon\JWTAuth\Exceptions\JWTException;
11
12 class PetugasController extends Controller
13 {
14     public function login(Request $request)
15     {
16         $credentials = $request->only('username', 'password');
17
18         try {
19             if (! $token = JWTAuth::attempt($credentials)) {
20                 return response()->json(['error' => 'invalid_credentials'], 400);
21             }
22         } catch (JWTException $e) {
23             return response()->json(['error' => 'could_not_create_token'], 500);
24         }
25
26         return response()->json(compact('token'));
27     }
28
29     public function register(Request $request)
30     {
31         $validator = Validator::make($request->all(), [
32             'nama_petugas' => 'required|string|max:255',
33             'username' => 'required|string|max:255',
34             'password' => 'required|string|min:6|confirmed',
35             'level' => 'required',
36         ]);
37
38         if($validator->fails()){
39             return response()->json($validator->errors()->toJson(), 400);
40         }
41     }
42 }
```

```

        $user = User::create([
            'nama_petugas' => $request->get('nama_petugas'),
            'alamat' => $request->get('alamat'),
            'no_hp' => $request->get('no_hp'),
            'username' => $request->get('username'),
            'password' => Hash::make($request->get('password')),
            'level' => $request->get('level'),
        ]);

        $token = JWTAuth::fromUser($user);

        return response()->json(compact('user','token'),201);
    }

    public function getAuthenticatedUser()
    {
        try {

            if (! $user = JWTAuth::parseToken()->authenticate()) {
                return response()->json(['user_not_found'], 404);
            }

        } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {

            return response()->json(['token_expired'], $e->getStatusCode());

        } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {

            return response()->json(['token_invalid'], $e->getStatusCode());

        } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {

            return response()->json(['token_absent'], $e->getStatusCode());

        }

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

```

4. Hasil Register

POST

http://localhost/Rental2/public/api/register

Send

Save

<input checked="" type="checkbox"/>	nama_petugas	Nisha	
<input checked="" type="checkbox"/>	username	nisha	
<input checked="" type="checkbox"/>	password	123456	
<input checked="" type="checkbox"/>	password_confirmation	123456	
<input checked="" type="checkbox"/>	level	admin	
	Key	Value	Description

Body

Cookies

Headers (10)

Test Results

Status: 201 Created Time: 924ms Size: 832 B Save Response

PrettyRawPreviewVisualizeJSON

```
{
  "user": {
    "nama_petugas": "Nisha",
    "username": "nisha",
    "level": "admin",
    "updated_at": "2020-04-07 12:21:04",
    "created_at": "2020-04-07 12:21:04",
    "id": 3
  },
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTB1bnRhbmQjZm9udHwvYXBpcyZhdpc3RlciIsIm1ldCI6MTU4NjI2MjA2NCwiZXhwIjoxNTg2MjY1NDYyOTk1ODYyYWljZWljQSImp0aSI6ImxzTG9reDN4c2JlZVBDbDhlcjdlWl0jMSInBydDI6IjgzZTBhZjZfZjZlmDE0DEyZmRlYzk3MTUzYTE0ZTBTMDQ3NTQZYWEiFQ_VrDCSURpdKdx2tbRSKdbFF9e7EwUtHgYjHdmD4KeC0"
```

5. Hasil Login

POST

http://localhost:8080/api/login

Send

Save

<input type="checkbox"/>	nama_petugas	Text	Nisha	
<input checked="" type="checkbox"/>	username		nisha	
<input checked="" type="checkbox"/>	password		123456	
<input type="checkbox"/>	password_confirmation		123456	
<input type="checkbox"/>	level		admin	
	Key		Value	Description

Body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 653ms

Size: 679 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "token":
3     "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi1wvXc9sb2NhbgHvc3RcLj1lbnRhbDJcL3B1YmxpY1wvYXBpXC9sb2dpbiIsIm1hdCI6MjU4NjI2MjI4NiwiZXhwIjoxNTg2MjY1ODg2LCJ1wYmYiOiJlODYyNjIyODYsImp0aSI6IjE1THhFVFREdU1KU0UxTXkiLCJ2dWUiOiJMsInBydi6Ijg3ZTBhZjF1ZjlmZDE1ODEyZmRlYzY0MTUzYTE0ZTB1MDQ3NTQ2YWEiFQ.wwRgvQVF0p939Y3H1prjGNatTdC19KKHm2-L4UzcZu"
```