

Luthfarras Tresanizega / XIR6 / 17

Tugas 4 JWT login-register

-membuat controller, model, dan middleware

```
D:\xampp\htdocs\projekperpus>php artisan make:controller PetugasController  
Controller created successfully.
```

```
D:\xampp\htdocs\projekperpus>php artisan make:model Petugass  
Model created successfully.
```

```
D:\xampp\htdocs\projekperpus>php artisan make:middleware middlewarejwt  
Middleware created successfully.
```

-api.php

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

-PetugasController

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\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',
        'alamat' => 'required|string|max:255',
        'telp' => 'required|string|max:255',
        'username' => 'required|string|max:255|',
        'password' => 'required|string|min:6|confirmed',
        'level' => 'required|string|max:255',
    ]);

    if($validator->fails()){
        return response()->json($validator->errors()->toJson(), 400);
    }

    $user = Petugas::create([
        'nama_petugas' => $request->get('nama_petugas'),
        'alamat' => $request->get('alamat'),
        'telp' => $request->get('telp'),
        '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'));
}
}

```

-model petugas

```

<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Tymon\JWTAuth\Contracts\JWTSubject;

class Petugas extends Authenticatable implements JWTSubject
{

    use Notifiable;
    protected $table = 'petugas';

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'nama_petugas', 'alamat', 'telp', 'username', 'password', 'level'
    ];
}

```

```
/**
 * The attributes that should be hidden for arrays.
 *
 * @var array
 */
protected $hidden = [
    'password', 'remember_token',
];

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

/**
 * The attributes that should be cast to native types.
 *
 * @var array
 */
protected $casts = [
    'email_verified_at' => 'datetime',
];
}
```

-register postman

The screenshot shows a Postman interface for a POST request to `http://localhost/projekperpus/public/api/register`. The request body is a JSON object with the following fields: `telp` (701-855-612), `username` (farrass), `password` (123456), `password_confirmation` (123456), and `level` (petugas). The response status is 201 Created, with a time of 2.94s and a size of 885 B. The response body is a JSON object containing user details and a token.

```
POST http://localhost/projekperpus/public/api/register

{
  "telp": "701-855-612",
  "username": "farrass",
  "password": "123456",
  "password_confirmation": "123456",
  "level": "petugas"
}
```

Status: 201 Created Time: 2.94s Size: 885 B Save Response

```
{
  "user": {
    "nama_petugas": "farrass",
    "alamat": "0K Boomer",
    "telp": "701-855-612",
    "username": "farrass",
    "level": "petugas",
    "updated_at": "2020-01-22 08:04:02",
    "created_at": "2020-01-22 08:04:02",
    "id": 6
  },
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3Rlc3Byb2p1a3B1cnB1c1wvchVibG1jXC9hcG1cL3J1Z21zdGVyIiwiaWF0IjoxNTc5NTk2MjYyLjE6MTUzOTY4MDI5MiwianRpIjoieVhRVm9B5SVk5bEUYU1WeSIsInN1YiI6NiwiChJ2IjoiodD1MGFmMmVhVW9kZkMTU4MTJmZGVjOTcxNTNhMTRlMG1wIDc1NDZhYSJ9.fRTP9hE2i_7qUwQF5VcH1XIpfP0p7oqyOOK53Xnc"
```

-login postman

The screenshot shows a Postman interface for a POST request to `http://localhost/projekperpus/public/api/login`. The request body is a JSON object with the following fields: `username` (farrass) and `password` (123456). The response status is 200 OK, with a time of 870ms and a size of 685 B. The response body is a JSON object containing a token.

```
POST http://localhost/projekperpus/public/api/login

{
  "username": "farrass",
  "password": "123456"
}
```

Status: 200 OK Time: 870ms Size: 685 B Save Response

```
{
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3Rlc3Byb2p1a3B1cnB1c1wvchVibG1jXC9hcG1cL2xvZ21uIiwiaWF0IjoxNTc5NTk2MjYyLjE6MTUzOTY4MDI5MiwianRpIjoieVhRVm9B5SVk5bEUYU1WeSIsInN1YiI6NiwiChJ2IjoiodD1MGFmMmVhVW9kZkMTU4MTJmZGVjOTcxNTNhMTRlMG1wIDc1NDZhYSJ9.fRTP9hE2i_7qUwQF5VcH1XIpfP0p7oqyOOK53Xnc"
```

-level postman

GET

http://localhost/projekperpus/public/api

Send

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

Code

▼ Headers (1)

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets ▼
<input checked="" type="checkbox"/>	Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...				
	Key	Value	Description			

► Temporary Headers (9) ⓘ

Body

Cookies

Headers (11)

Test Results

Status: 200 OK

Time: 213ms

Size: 366 B

Save Response ▼

Pretty

Raw

Preview

Visualize BETA

HTML ▼

1

petugas