

- Model

```
class User extends Authenticatable implements JWTSubject
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $table = 'petugas';
    public $timestamps = false;
    protected $fillable = [
        'nama_petugas', 'username', 'password', 'level', 'alamat',
    ];

    /**
     * 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 [];
    }
}
```

- Controller

- a. Login

```
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'));
}
```

b. Register

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

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

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

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

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

- Routes API

```
Route::post('/register', 'Petugas@register');
Route::post('/login', 'Petugas@login');
```

- POSTMAN

a. Register

POST http://localhost:8000/api/register Send Save

<input checked="" type="checkbox"/>	nama_petugas	nuril haidar	
<input checked="" type="checkbox"/>	password	123456	
<input checked="" type="checkbox"/>	password_confirmation	123456	
<input checked="" type="checkbox"/>	username	nuril123@gmail	
<input checked="" type="checkbox"/>	level	admin	
<input checked="" type="checkbox"/>	alamat	j.l.pisang	
	Key	Value	Description

Body Cookies Headers (9) Test Results Status: 201 Created Time: 4.03s Size: 725 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "user": {
3     "nama_petugas": "nuril haidar",
4     "alamat": "j.l.pisang",
5     "level": "admin",
6     "username": "nuril123@gmail",
7     "id": 1
8   },
9   "token":
10  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IWRXbWVudC9yZWdpc3Ric2ludHkiOjEwNjZmNDYyNCwiZXhwIjoxNTg2MzM4MjE0LCJybmVmeSUiOiJ0ODYyMzQ2MjQ1ImpeasI6IjRMZnpuQXFsvZE1UDkKdmlkLCJmdWIojesInbYdiI6Ijg3ZTBhZjFlZjlmZDE0OEYzmkMTUyZTE0OTB1MDQ3NTQyMEIq.QROxs1PUFun4zRDAsZTOfdT6TKqAuQ8rTmhgl32ld8"
```

b. Login

POST http://localhost:8000/api/login Send Save

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

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

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	nuril123@gmail			
<input checked="" type="checkbox"/>	password	123456			
	Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 2.14s Size: 606 B Save Response

Pretty Raw Preview Visualize JSON ≡

```
1  {
2    "token":
3    "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbgVhc3Q6ODAwMFwvYXBpcXN5b2dpbiIsIm1hdCI6MTU4NjMzNDcwNCw1ZXhwIjoxNTg2MzQ4MzA4LCJ1bmY1OiJlODYzMzQ3MDQsImp0eSI6Inh0R2JoMzRNMGtrwmuwzZGE1LCJzdwIjOiJEsInBydiI6Ijg3ZTBhZjF1Zj1mZDE1ODUyZmR1Yz1kMTUyYTE0ZTBiMDQ3NTQ2YWUifQ.hvW4pxK3ofq3BioCudqvzy1odGDMOQ6jJupnJit9EtI"
```