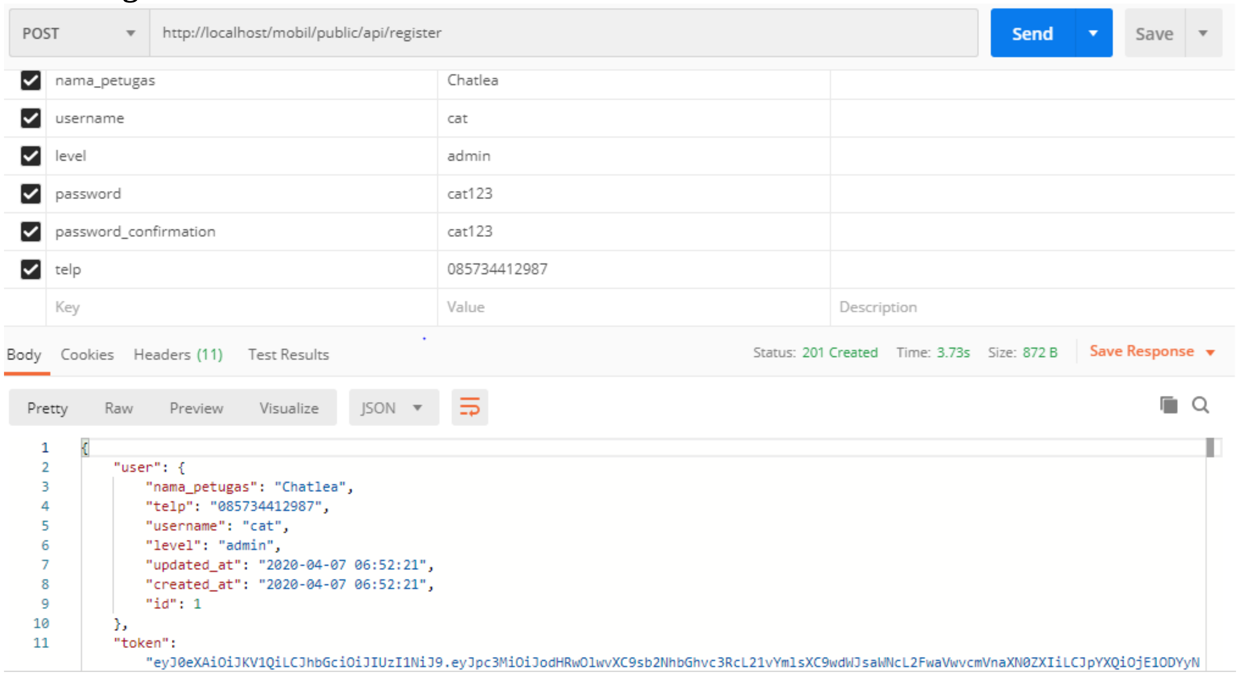
Nama : Chatlea Cinta Putri Widyanto

Kelas : Laravel

No. Absen : 09

Tugas API Authentication Rental Mobil Menggunakan JWT

1. Hasil Register Admin di Postman



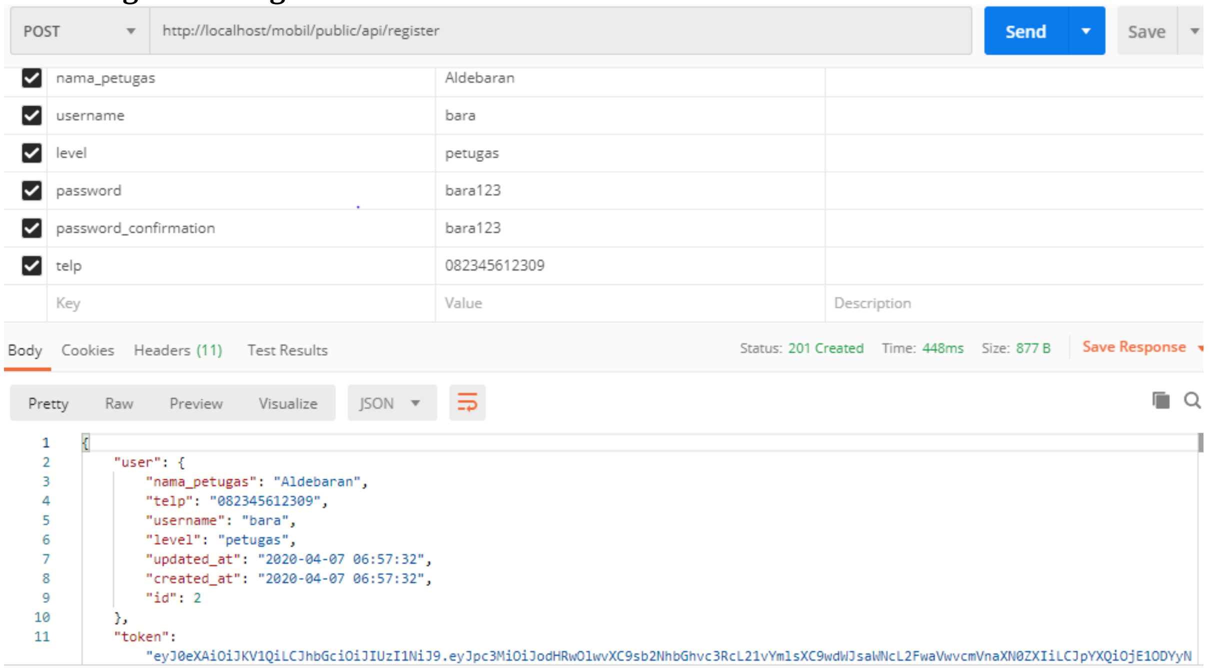
POST http://localhost/mobil/public/api/register

	Key	Value	Description
✓	nama_petugas	Chatlea	
✓	username	cat	
✓	level	admin	
✓	password	cat123	
✓	password_confirmation	cat123	
✓	telp	085734412987	

Status: 201 Created Time: 3.73s Size: 872 B

```
{
    "user": {
        "nama_petugas": "Chatlea",
        "telp": "085734412987",
        "username": "cat",
        "level": "admin",
        "updated_at": "2020-04-07 06:52:21",
        "created_at": "2020-04-07 06:52:21",
        "id": 1
    },
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL21vYmlsXC9wdWJsaWNcL2FwaVwvcmVnaXN0ZXIiLCJpYXQiOjE1ODYyN
```

2. Hasil Register Petugas di Postman



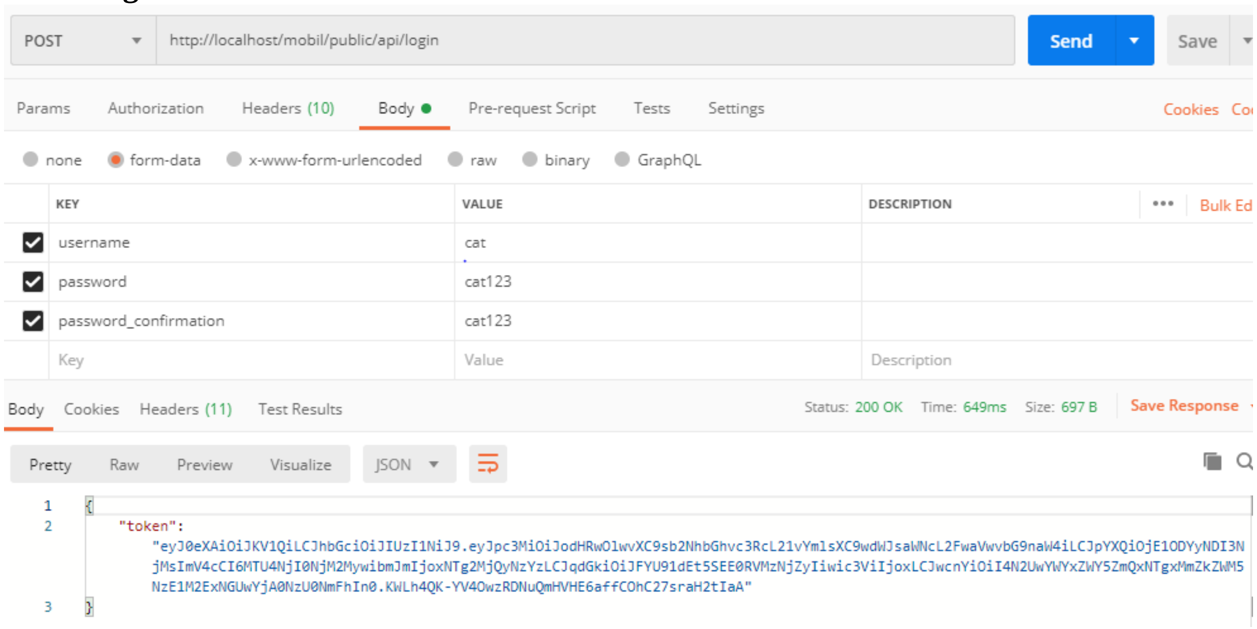
POST http://localhost/mobil/public/api/register

	Key	Value	Description
✓	nama_petugas	Aldebaran	
✓	username	bara	
✓	level	petugas	
✓	password	bara123	
✓	password_confirmation	bara123	
✓	telp	082345612309	

Status: 201 Created Time: 448ms Size: 877 B

```
{
    "user": {
        "nama_petugas": "Aldebaran",
        "telp": "082345612309",
        "username": "bara",
        "level": "petugas",
        "updated_at": "2020-04-07 06:57:32",
        "created_at": "2020-04-07 06:57:32",
        "id": 2
    },
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL21vYmlsXC9wdWJsaWNcL2FwaVwvcmVnaXN0ZXIiLCJpYXQiOjE1ODYyN
```

3. Hasil Login Admin di Postman



POST http://localhost/mobil/public/api/login

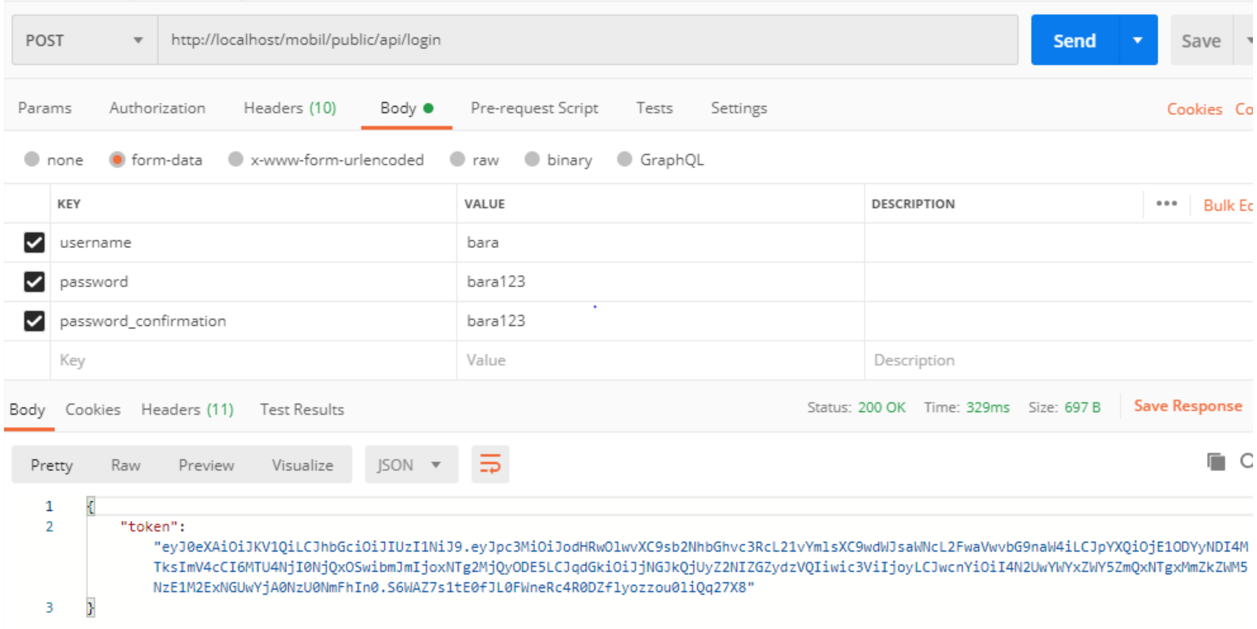
Body: form-data

KEY	VALUE	DESCRIPTION
username	cat	
password	cat123	
password_confirmation	cat123	

Status: 200 OK Time: 649ms Size: 697 B

```
{
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL21vYmlsXC9wdWJsaWNcL2FwaVwvbG9naW4iLCJpYXQiOjE1ODYyNDI3N
        jMsImV4cCI6MTU4NjI0NjM2MywibmJmIjoxNTg2MjQyNzYzLCJqdGkiOiJFYU91dEt5SEE0RVMzNjZyIiwic3ViIjoxLCJwcnYiOiI4N2UwYWYxZWY5ZmQxNTgxMmZkZWM5
        NzE1M2ExNGUwYjA0NzU0NmFhIn0.KWLh4QK-YV4OwzRDNuQmHVHE6affCOhC27sraH2tIaA"
}
```

4. Hasil Login Petugas di Postman



POST http://localhost/mobil/public/api/login

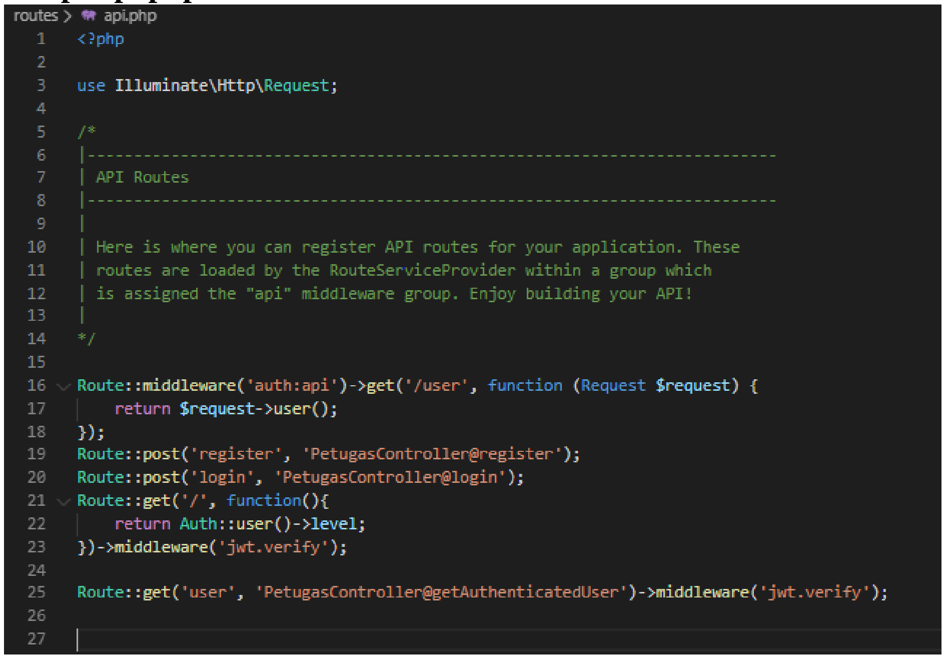
Body: form-data

KEY	VALUE	DESCRIPTION
username	bara	
password	bara123	
password_confirmation	bara123	

Status: 200 OK Time: 329ms Size: 697 B

```
{
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL21vYmlsXC9wdWJsaWNcL2FwaVwvbG9naW4iLCJpYXQiOjE1ODYyNDI4M
        TksImV4cCI6MTU4NjI0NjQxOSwibmJmIjoxNTg2MjQyODE5LCJqdGkiOiJjNGJkQjUyZ2NIZGZydzVQIiwic3ViIjoyLCJwcnYiOiI4N2UwYWYxZWY5ZmQxNTgxMmZkZWM5
        NzE1M2ExNGUwYjA0NzU0NmFhIn0.S6WAZ7s1tE0fJL0FWneRc4R0DZflyozzou0liQq27X8"
}
```

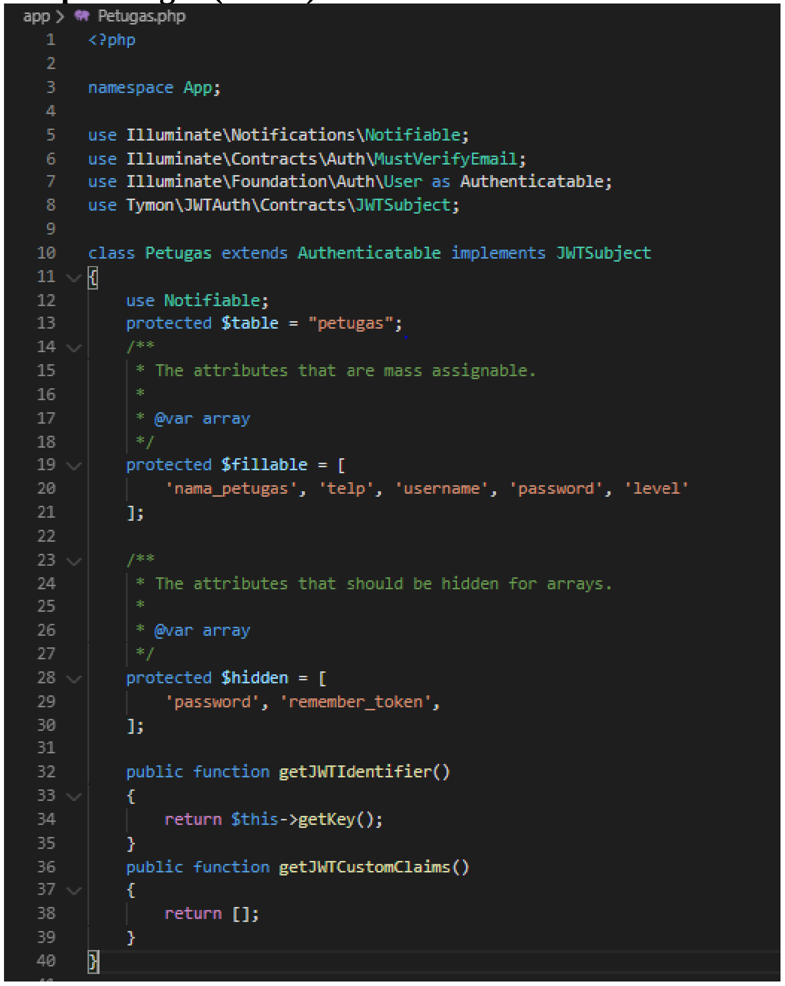
5. **Script api.php**



routes > api.php

```php
<?php

use Illuminate\Http\Request;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
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');

```

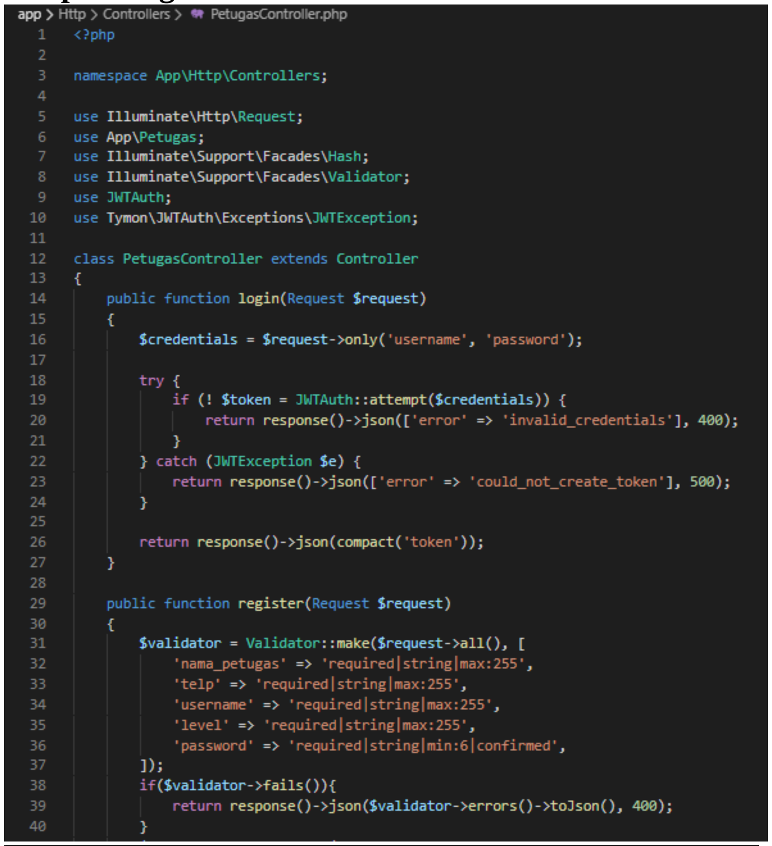
6. **Script Petugas (model)**



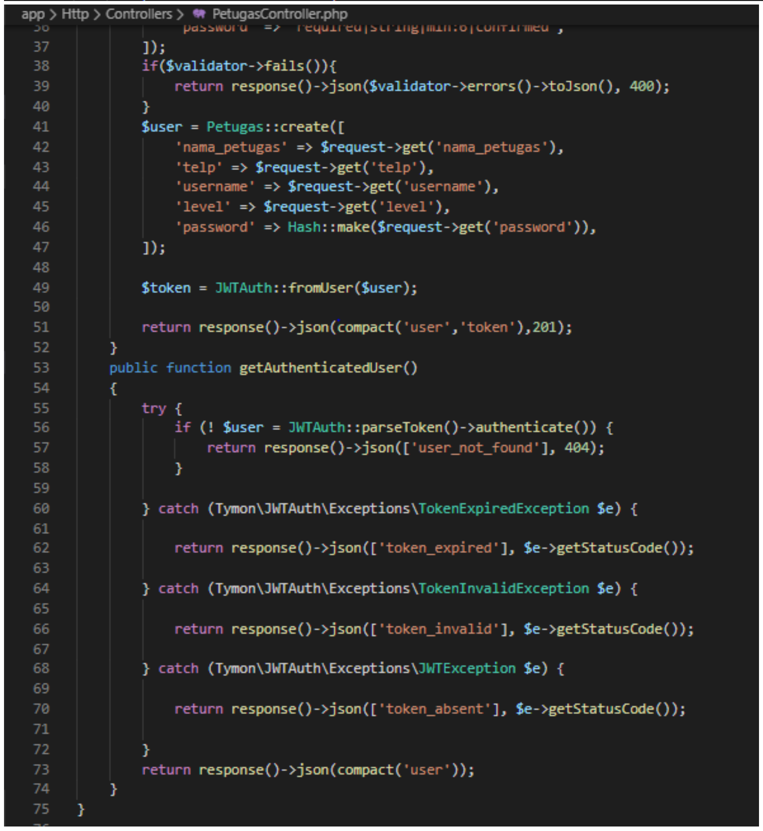
app > Petugas.php

```php
<?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', '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 [];
    }
}
```

7. **Script Petugas Controller**

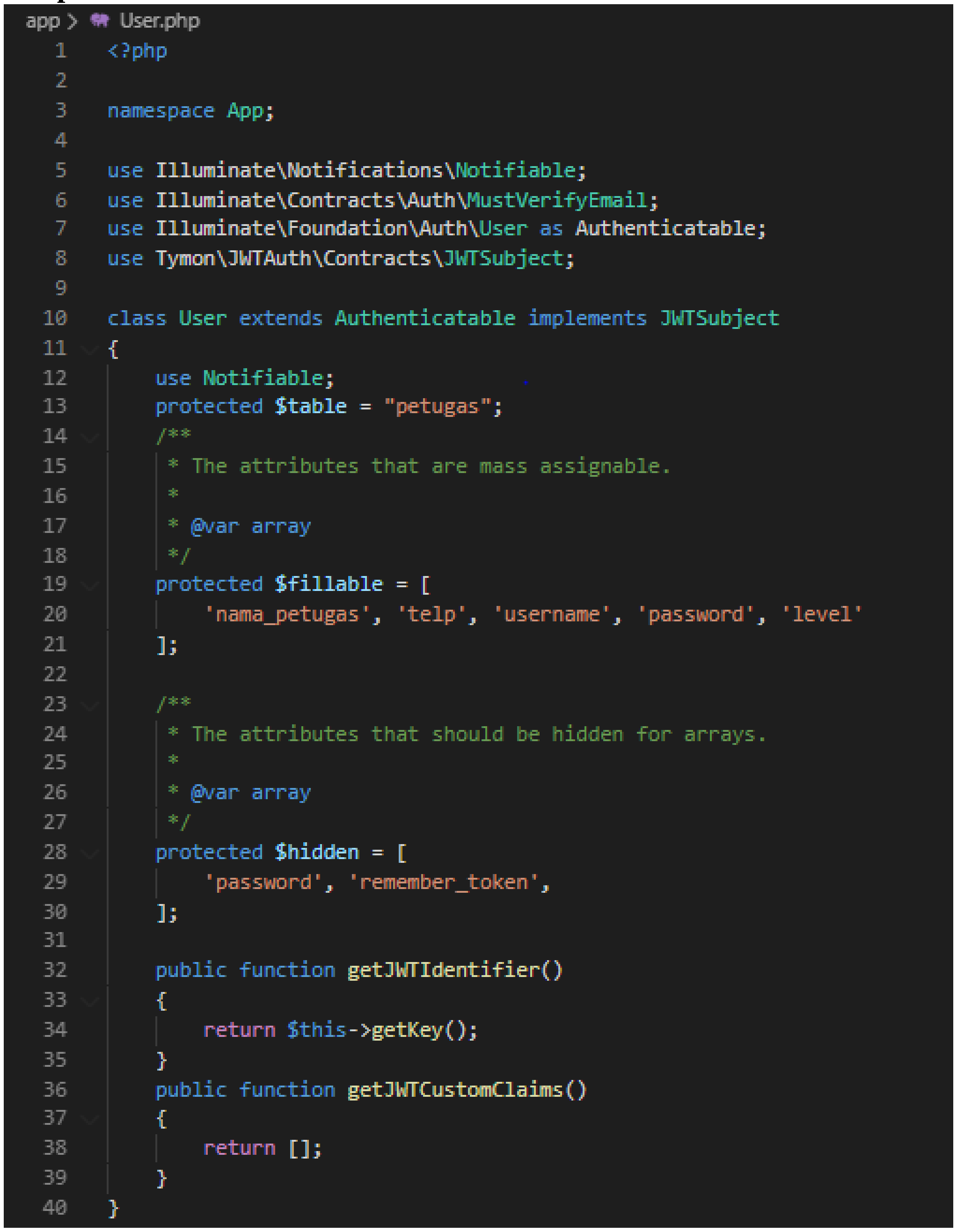

app > Http > Controllers > PetugasController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Petugas;
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',
            'telp' => 'required|string|max:255',
            'username' => 'required|string|max:255',
            'level' => 'required|string|max:255',
            'password' => 'required|string|min:6|confirmed',
        ]);
        if($validator->fails()){
            return response()->json($validator->errors()->toJson(), 400);
        }
```


app > Http > Controllers > PetugasController.php

```php
        ]);
        if($validator->fails()){
            return response()->json($validator->errors()->toJson(), 400);
        }
        $user = Petugas::create([
            'nama_petugas' => $request->get('nama_petugas'),
            'telp' => $request->get('telp'),
            'username' => $request->get('username'),
            'level' => $request->get('level'),
            'password' => Hash::make($request->get('password')),
        ]);

        $token = JWTAuth::fromUser($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'));
    }
}
```

8. Script Model User



app > User.php

```php
<?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 User extends Authenticatable implements JWTSubject
{
    use Notifiable;
    protected $table = "petugas";
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'nama_petugas', '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 [];
    }
}
```