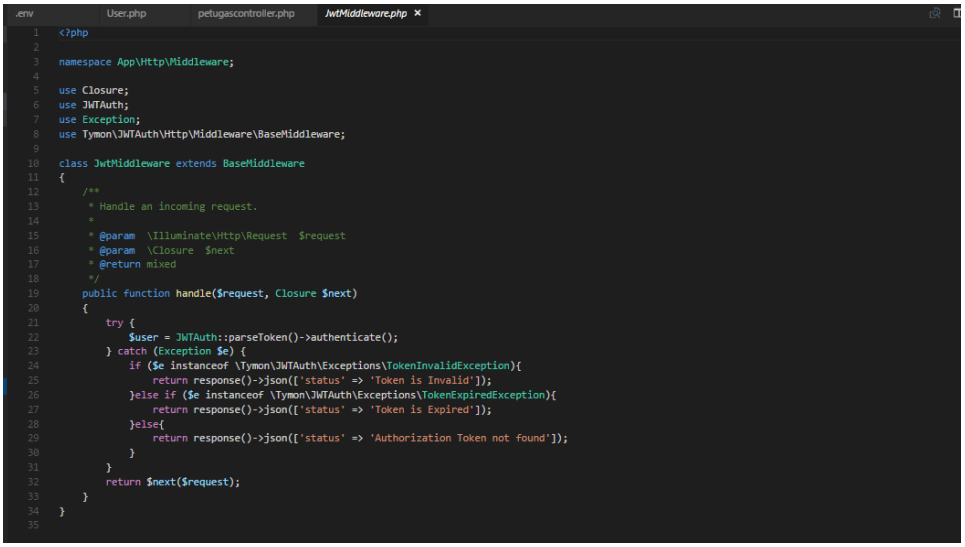
Salsabila Gemintang Kejora

- **Make controller**

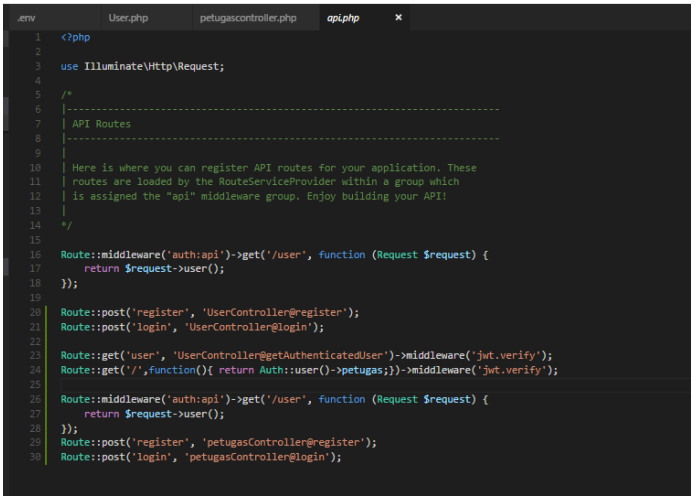
```
Command Prompt
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd c:\xampp\htdocs\perpus\

c:\xampp\htdocs\perpus>php artisan make:controller petugascontroller
```

- **Middleware**



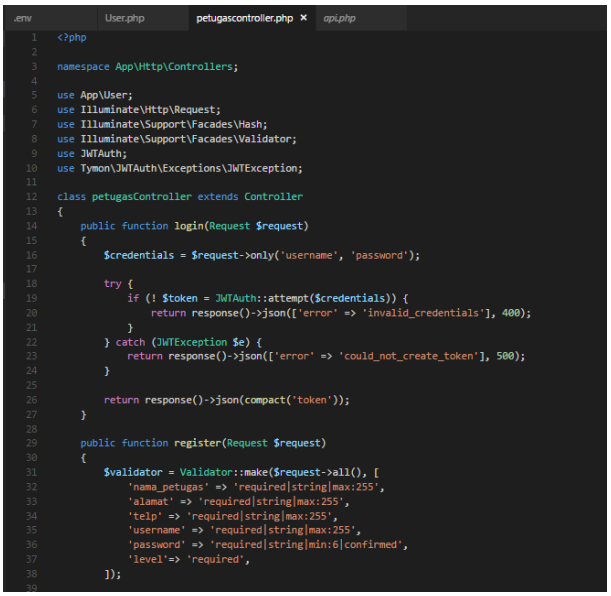
```php
<?php

namespace App\Http\Middleware;

use Closure;
use JWTAuth;
use Exception;
use Tymon\JWTAuth\Http\Middleware\BaseMiddleware;

class JwtMiddleware extends BaseMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        try {
            $user = JWTAuth::parseToken()->authenticate();
        } catch (Exception $e) {
            if ($e instanceof \Tymon\JWTAuth\Exceptions\TokenInvalidException){
                return response()->json(['status' => 'Token is Invalid']);
            }else if ($e instanceof \Tymon\JWTAuth\Exceptions\TokenExpiredException){
                return response()->json(['status' => 'Token is Expired']);
            }else{
                return response()->json(['status' => 'Authorization Token not found']);
            }
        }
        return $next($request);
    }
}
```

- **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', 'UserController@register');
Route::post('login', 'UserController@login');

Route::get('user', 'UserController@getAuthenticatedUser')->middleware('jwt.verify');
Route::get('/',function(){ return Auth::user()->petugas;})->middleware('jwt.verify');

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
Route::post('register', 'petugasController@register');
Route::post('login', 'petugasController@login');
```

- **Controller**



```php
<?php

namespace App\Http\Controllers;

use App\User;
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',
        ]);
```

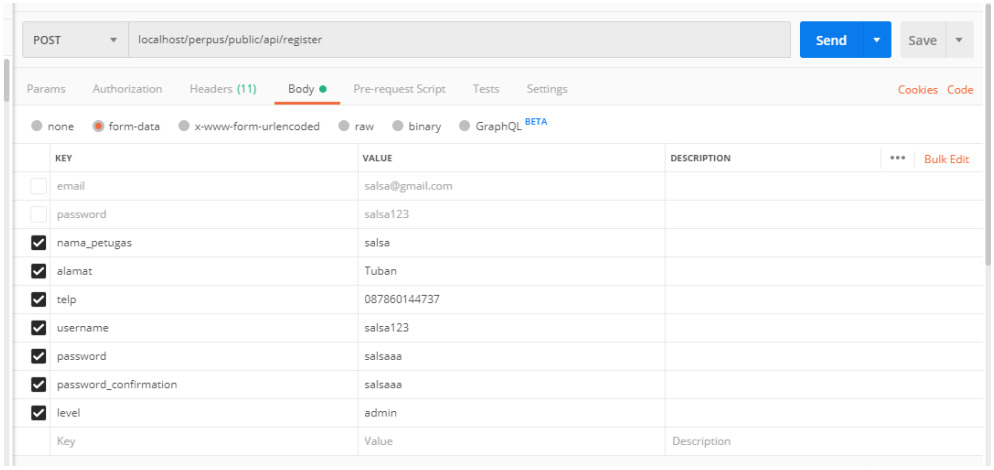
```php
        if($validator->fails()){
            return response()->json($validator->errors()->toJson(), 400);
        }

        $user = User::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);
    }
```

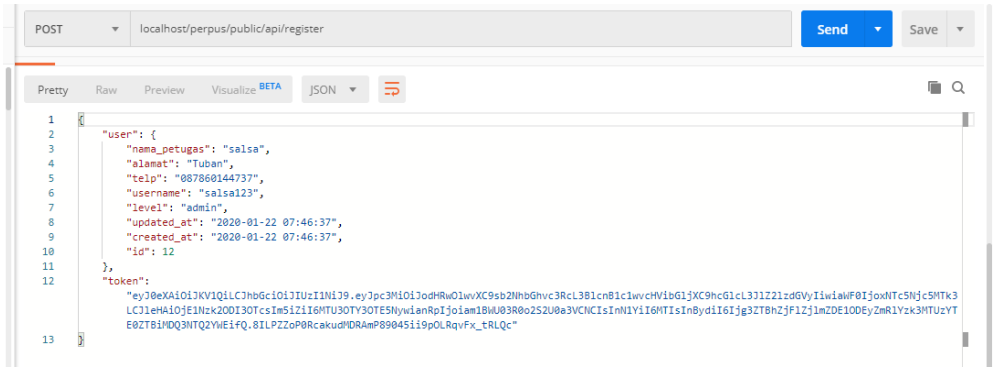
```php
    public function getAuthenticatedUser()
    {
        try {

            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**

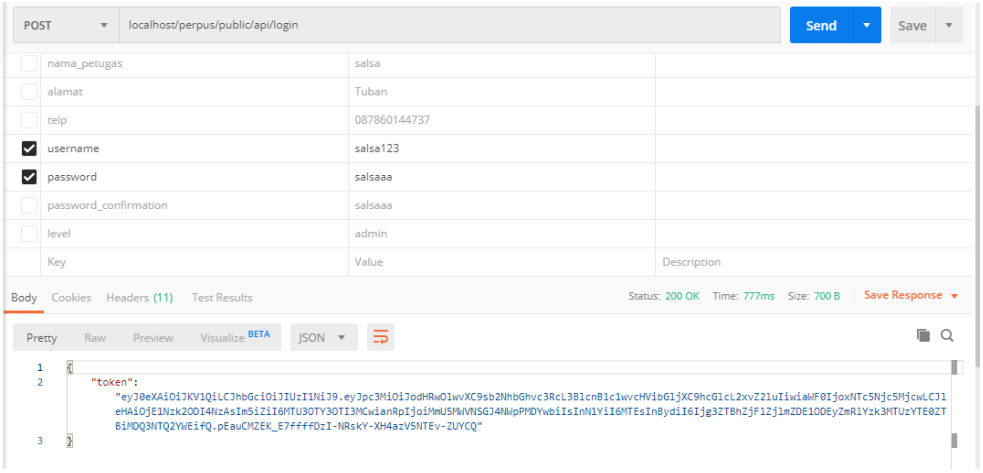
```
.env   User.php   petugascontroller.php   api.php

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
{
    protected $table = 'petugas';
        use Notifiable;

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

- **Register**





- **Login**


POST
localhost/perpus/public/api/login
Send
Save
nama_petugas
salsa
alamat
Tuban
telp
087860144737
username
salsa123
password
salsaaa
password_confirmation
salsaaa
level
admin
Key
Value
Description
Body
Cookies
Headers (11)
Test Results
Status: 200 OK
Time: 777ms
Size: 700 B
Save Response
Pretty
Raw
Preview
Visualize BETA
JSON
{
"token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL3BlcnB1c1wvcHVibGljXC9hcGlcL2xvZ2luIiwiaWF0IjoxNTc5Njc5MjcwLCJ1eHAiOjE1Nzk2ODI4NzAsIm5iZiI6MTU3OTY3OTI3MCwianRpIjoiMmU5MWVNSGJ4NWpPMDYwbiIsInN1YiI6MTEsInBydiI6Ijg3ZTBhZjF1ZjlmZDE1ODEyZmRlYzk3MTUzYTE0ZTBiMDQ3NTQ2YWEifQ.pEauCMZEK_E7ffffDzI-NRskY-XH4azV5NTEv-ZUYCQ"
}