Model

- 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'));
}
```

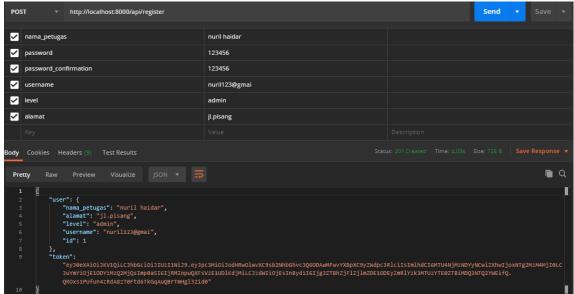
b. Register

```
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



## b. Login

