### • PETUGAS CONTROLLER

```
petugascontroller.php × JwtMiddleware.php Kernel.php
use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Hash;
use IIIminate\Support\Facades\Validator;
use JWTAUt\Exceptions\JWTException;
                return response()->json(['error' => 'invalid_credentials'], 400);

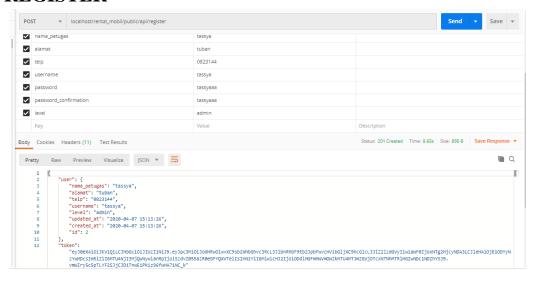
} catch (]MTException $e) {
    return response()->json(['error' => 'could_not_create_token'], 500);
}
                $validator = Validator::make($request->all(), [
    'nama_petugas' >> 'required|string|max:255',
    'alamat' >> 'required|string|max:255',
    'telp' >> 'required|string|max:250',
    'username' >> 'required|string|max:255',
    'password' >> 'required|string|min:6|confirmed',
    'level' >> 'required',
}
                 if($validator->fails()){
    return response()->json($validator->errors()->toJson(), 400);
                 Suser = User::create([
    'nama_petugas' >> Srequest->get('nama_petugas'),
    'alamst' >> Srequest->get('alamat'),
    'telp' >> Srequest->get('telp'),
    'username' >> Srequest->get('username'),
    'password' >> Mash:make(Srequest->get('password')),
    'level' >> Srequest->get('level'),
};
        public function getAuthenticatedUser()
{
                       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());
                        return response()->json(['token_absent'], $e->getStatusCode());
```

### • PETUGAS MODEL

# • Api.php

```
Route::post('register', 'petugascontroller@register');
Route::post('login', 'petugascontroller@login');
Route::get('user', 'petugascontroller@getAuthenticatedUser')->middleware('jwt.verify');
23
```

## REGISTER



## • LOGIN

