Nama : Sania Tri A

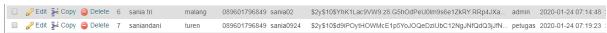
Kelas : XI RPL 6

Absen : 36

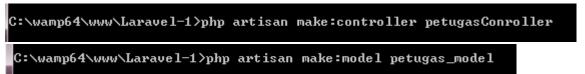
Tuga 4

## Login & Register

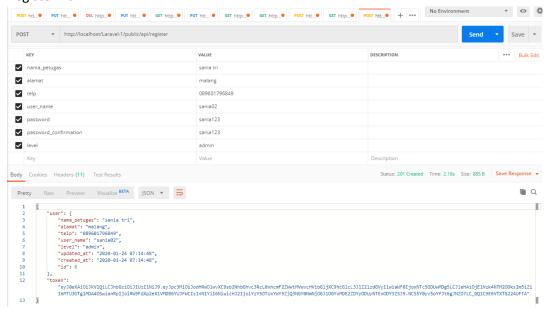
**1.** DB



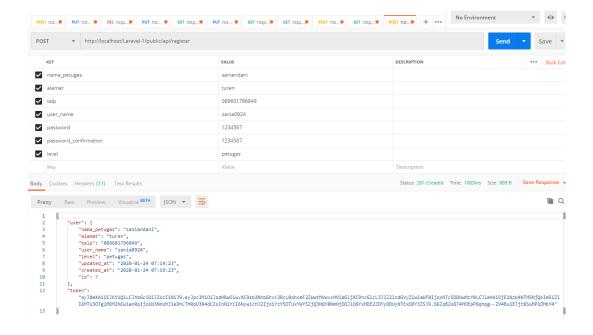
**2.** Cmd



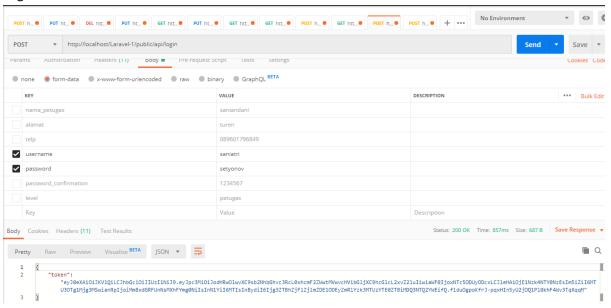
3. Register Admin



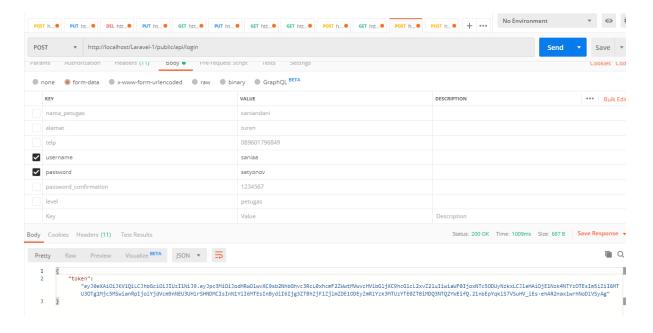
**Register Petugas** 



## 4. Login Admin



**Login Petugas** 



## 5. petugasController

```
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)
        $credential = $request->only('username','password');
        try {
            if (! $token = JWTAuth::attempt($credential)) {
                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',
        ]);
```

```
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);
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(['token_absent'], $e->getStatusCode());
     return response()->json(compact('user'));
```

6. Petugas\_model

```
namespace App;
5
    use Illuminate\Notifications\Notifiable;
    use Illuminate\Contracts\Auth\MustVerifyEmail;
    use Illuminate\Foundation\Auth\User as Authenticatable;
    use Tymon\JWTAuth\Contracts\JWTSubject;
    class petugas_model extends Authenticatable implements JWTSubject
        use Notifiable;
        protected $table = 'table petugas';
         * @var array
        protected $fillable = [
           'nama_petugas', 'alamat', 'telp', 'user_name', 'password', 'level',
        ];
        * @var array
        protected $hidden = [
           'password', 'remember_token',
        ];
         * @var array
        protected $casts = [
            'email_verified_at' => 'datetime',
            public function getJWTIdentifier()
                 return $this->getKey();
            public function getJWTCustomClaims()
            {
                 return [];
       }
```

7. Route

```
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');
```