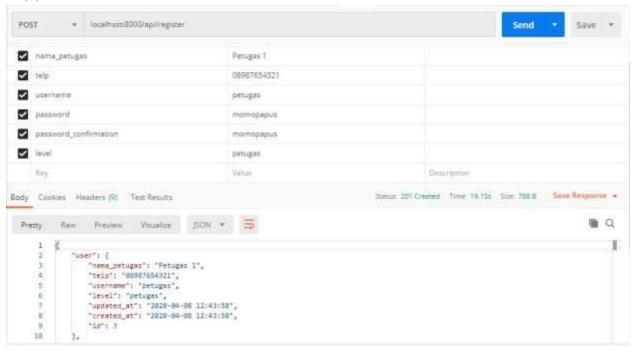
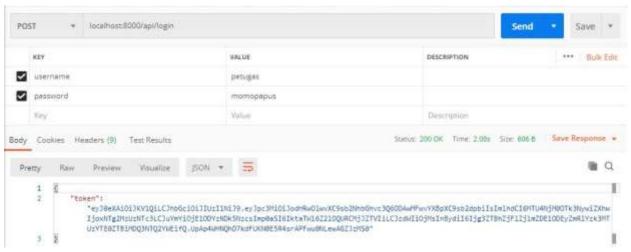
XI RPL 6/16

1. REGISTER



2. LOGIN



3. CONTROLLER

```
PetugasController.php
    namespace App\Http\Controllers;
5 use App\User;
6 use App\Petugas model;
7 use Illuminate\Http\Request;
a use Illuminate\Support\Facades\Hash;
9 use Illuminate\Support\Facades\Validator;
10 use JWTAuth;
   use Tymon\JWTAuth\Exceptions\JWTException;
   use Auth;
   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|unique:petugas',
        'password' => 'required|string|min:6|confirmed',
        'level' => 'required|string|max:255',
   ]);
    if($validator->fails()){
        return response()->json($validator->errors()->toJson(), 400);
    $user = User::create([
        'nama petugas' => $request->get('nama petugas'),
        '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);
```

```
PetugasController.php

public function getAuthenticatedUser()

{
    try {
        if (! Suser = 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(['token_absent'], $e->getStatusCode());

return response()->json(compact('user'));

}
```

```
PetugasController.php
  public function tampil(){
    if(Auth::user()->level=="admin"){
  $datas = Petugas model::get();
  $count = $datas->count();
  $film = array();
  $status = 1;
  foreach ($datas as $dt sw){
   $film[] = array(
      'id' => $dt sw->id,
      'nama petugas' => $dt sw->nama petugas,
      'telp' => $dt_sw->telp,
      'username' => $dt sw->username,
      'password' => $dt_sw->password,
      'level' => $dt sw->level,
      'created at' => $dt sw->created at,
      'updated_at' => $dt_sw->updated_at,
  return Response()->json(compact('count', 'film'));
  return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
public function update($id, Request $req)
  if(Auth::user()->level=="admin"){
  $validator=Validator::make($req->all(),
    'nama petugas' => 'required',
    'telp' => 'required',
```

4. MODEL

```
User.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 Stable = "petugas";
        @var array
       protected $fillable = [
          'nama_petugas', 'telp', 'username', 'password', 'level'
       protected $hidden = [
           'password', 'remember_token',
       public function getJWTIdentifier()
           return $this->getKey();
       public function getJWTCustomClaims()
```