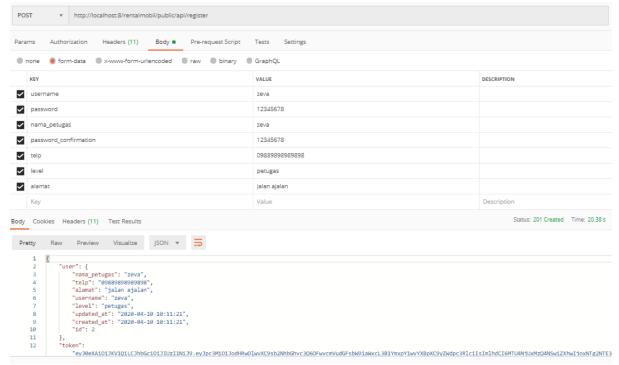
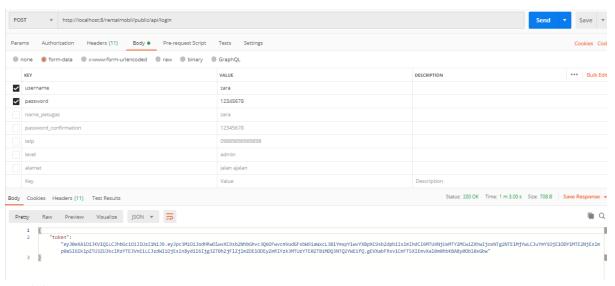
Nama: Zara Zeva Az Zurra

Kelas: XIRPL3 / 41

1. Hasil input register



2. Hasil input login



3. Model

```
namespace App;
use Illuminate\Database\Eloquent\Model;
class petugas model extends Model
  protected $table="petugas";
  protected $primaryKey="id";
  public $timestamps= false;
     'nama_petugas', 'alamat', 'telp', 'username', 'password', 'level'
   ];
2 namespace App;
3 use Illuminate\Notifications\Notifiable;
4 use Illuminate\Contracts\Auth\MustVerifyEmail;
  use Illuminate\Foundation\Auth\User as Authenticatable;
   use Tymon\JWTAuth\Contracts\JWTSubject;
   class User extends Authenticatable implements JWTSubject
   use Notifiable;
   * @var array
   protected $table = 'petugas';
   'nama_petugas', 'username', 'password', 'level', 'telp', 'alamat',
     @var array
```

```
protected $hidden = [
  'password', 'remember_token',
  ];
public function getJWTIdentifier()
  {
  return $this->getKey();
  }
  public function getJWTCustomClaims()
  {
  return [];
  }
  }
}
```

4. Controller

```
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',
         'alamat' => 'required|string|max:255',
        'username' => 'required|string|max:255|unique:petugas',
         'password' => 'required|string|min:6|confirmed',
         'level' => 'required',
     if($validator->fails()){
        return response()->json($validator->errors()->toJson(), 400);
        'nama_petugas' => $request->get('nama_petugas'),
         'telp' => $request->get('telp'),
        'alamat' => $request->get('alamat'),
         '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\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'));
}
```

5. Api

```
});
Route::post('register', 'petugas_controller@register');
Route::post('login', 'petugas_controller@login');
```