

Kelas: XIRPL3 / 41

1. Hasil input register

POST

http://localhost:8/rentalmobil/public/api/register

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

KEY	VALUE	DESCRIPTION
username	zeva	
password	12345678	
nama_petugas	zeva	
password_confirmation	12345678	
telp	09889898989898	
level	petugas	
alamat	jalan ajalan	
Key	Value	Description

Body

Cookies

Headers (11)

Test Results

Status: 201 Created Time: 20.38 s

Pretty

Raw

Preview

Visualize

JSON

```

1  {
2    "user": {
3      "nama_petugas": "zeva",
4      "telp": "09889898989898",
5      "alamat": "jalan ajalan",
6      "username": "zeva",
7      "level": "petugas",
8      "updated_at": "2020-04-10 10:11:21",
9      "created_at": "2020-04-10 10:11:21",
10     "id": 2
11   },
12   "token":

```

## 2. Hasil input login

**POST** http://localhost:8/rentalmobil/public/api/login Send Save

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings Cookies Cod

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	zara	
<input checked="" type="checkbox"/> password	12345678	
<input type="checkbox"/> nama_petugas	zara	
<input type="checkbox"/> password_confirmation	12345678	
<input type="checkbox"/> telp	09889898989898	
<input type="checkbox"/> level	admin	
<input type="checkbox"/> alamat	jalan ajalan	
Key	Value	Description

**Body** Cookies Headers (11) Test Results Status: 200 OK Time: 1 m 3.00 s Size: 708 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "token":
3     "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJkbWlzdHkiLCJyb2NhdGVudGFsbGUiOiJkYXZlLjB1IiwiaWF0IjEwMTYyMmQwLzIxLjowNTg2NTEiMyJwCjUyY0Y0JFODY1ZTE2NjEsImpeasI6Ikp2TU3ZUjckIHZyTE9NeILCjdldiojSiInbydi16IjszIjlnZDl0DEyZmlzsk3MTUYzTE8TBMQ3JNTQ2WEIq. gEVxabFRsv1CnFTSXIEmvKa1emhKBADy80b18xdgw"
}
```

### 3. Model

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class petugas_model extends Model
{
    protected $table="petugas";
    protected $primaryKey="id";
    public $timestamps= false;
    protected $fillable = [
        'nama_petugas', 'alamat', 'telp', 'username', 'password', 'level'
    ];
}

```

```

1  <?php
2  namespace App;
3  use Illuminate\Notifications\Notifiable;
4  use Illuminate\Contracts\Auth\MustVerifyEmail;
5  use Illuminate\Foundation\Auth\User as Authenticatable;
6  use Tymon\JWTAuth\Contracts\JWTSubject;
7  class User extends Authenticatable implements JWTSubject
8  {
9      use Notifiable;
10     /**
11      * The attributes that are mass assignable.
12      *
13      * @var array
14      */
15     protected $table = 'petugas';
16
17     protected $fillable = [
18         'nama_petugas', 'username', 'password', 'level', 'telp', 'alamat',
19     ];
20     /**
21      * The attributes that should be hidden for arrays.
22      *
23      * @var array

```

```

protected $hidden = [
    'password', 'remember_token',
];

public function getJWTIdentifier()
{
    return $this->getKey();
}

public function getJWTCustomClaims()
{
    return [];
}
}

```

#### 4. Controller

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\User;
6
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Support\Facades\Validator;
10 use JWTAuth;
11 use Tymon\JWTAuth\Exceptions\JWTException;
12
13 class petugas_controller extends Controller
14 {
15     public function login(Request $request)
16     {
17         $credentials = $request->only('username', 'password');
18
19         try {
20             if (! $token = JWTAuth::attempt($credentials)) {
21                 return response()->json(['error' => 'invalid_credentials'], 400);
22             }
23         } catch (JWTException $e) {
24             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',
            '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);
        }

        $user = User::create([
            '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');

```