

Nama : Ananda Rafi Amanullah

Kelas : XI RPL 7 / 06 (Laravel/04)

Mohon maaf pak ini tidak ada jwt, dikarenakan authorization dari awal tidak bisa. Mohon maaf

## AnggotaModel

```
api.php  BukuController.php  Buku.php  Anggota.php  JwtMiddleware.php
app > 1 Anggota.php
2 <?php
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Anggota extends Model
8 {
9     protected $table = "anggota";
10     protected $primaryKey = "id_anggota";
11     protected $fillable = [
12         'id_anggota',
13         'nama_anggota',
14         'telp',
15         'alamat',
16     ];
17     public $timestamps = false;
18 }
19
```

## Buku Model

```
api.php  BukuController.php  Buku.php  Anggota.php  JwtMiddleware.php
app > 1 Buku.php
2 <?php
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Buku extends Model
8 {
9     protected $table = "buku";
10     protected $primaryKey = "id_buku";
11     protected $fillable = [
12         'id_buku',
13         'judul',
14         'penerbit',
15         'pengarang',
16         'foto',
17     ];
18     public $timestamps = false;
19 }
20
```

## ROUTE API

```
api.php x BukuController.php Anggota.php JethicMiddleware.php auth.php
outes > api.php
 9 |
10 | Here is where you can register API routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | is assigned the "api" middleware group. Enjoy building your API!
13 |
14 | */
15 |
16 | Route::middleware('auth:api')->get('/user', function (Request $request) {
17 |     return $request->user();
18 | });
19 |
20 | /*
21 | |-----|
22 | | SEARCH |
23 | |-----|
24 | */
25 | //Route::get('/',function(){
26 | //return Auth::user()->idlevel;
27 | //})->middleware("jwt.verify");
28 |
29 | /*
30 | |-----|
31 | | LOGIN PETUGAS TATIM |
32 | |-----|
33 | */
34 | Route::post('login','PetugasController@login');
35 | Route::get('/login/check','PetugasController@getAuthenticatedUser');
36 |
37 | /*
38 | |-----|
39 | | CRUD ANGGOTA |
40 | |-----|
41 | */
42 | Route::post('anggota','AnggotaController@store');
43 | Route::put('anggota/{id}','AnggotaController@update');
44 | Route::delete('anggota/{id}','AnggotaController@delete');
45 | Route::get('anggota','AnggotaController@tampil');
46 |
47 | /*
48 | |-----|
49 | | CRUD BUKU |
50 | |-----|
51 | */
52 | Route::post('buku','BukuController@store');
53 | Route::put('buku/{id}','BukuController@update');
54 | Route::delete('buku/{id}','BukuController@delete');
55 | Route::get('buku','BukuController@tampil');
56 |
57 | /*
58 | |-----|
59 | | CRUD PETUGAS |
60 | |-----|
61 | */
62 | Route::post('petugas','PetugasController@store');
63 | Route::put('petugas/{id}','PetugasController@update');
64 | Route::delete('petugas/{id}','PetugasController@delete');
65 | Route::get('petugas','PetugasController@tampil');
```

## DELETE ANGGOTA

[CONFLICT] Localhost/Perpustakaan

No Environment

Untitled Request

DELETE Localhost/Migrations/public/api/anggota/1

Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_anggota	RAFI HALLO	
<input checked="" type="checkbox"/> alamat	MALANG	
<input checked="" type="checkbox"/> telp	0841241241231	
<input type="checkbox"/> foto	HALP.NG	
<input type="checkbox"/> role	petugas	
Key	Value	Description

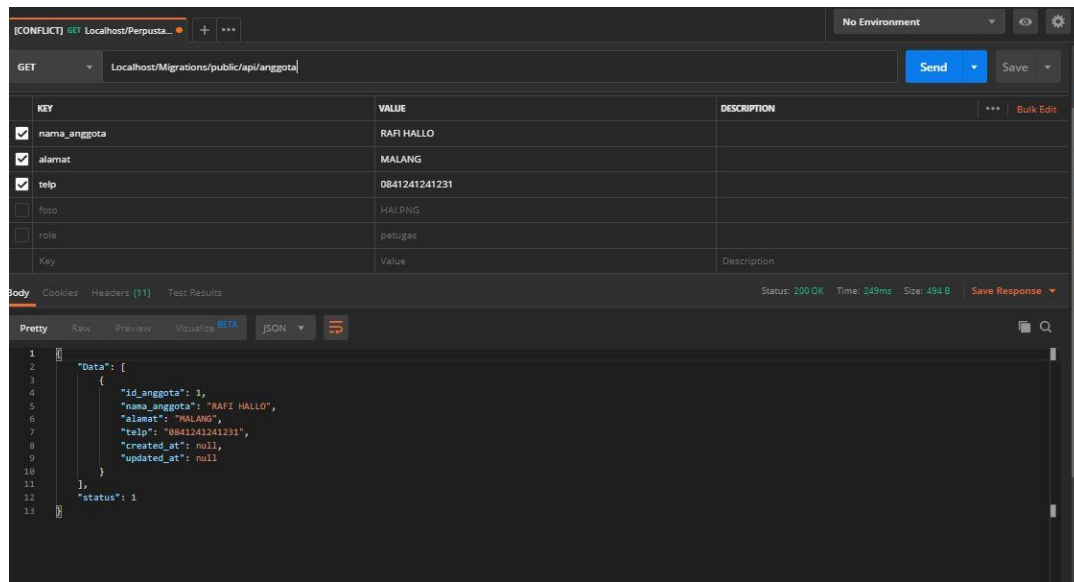
Body Cookies Headers (11) Test Results

Status: 200 OK Time: 374ms Size: 404 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": 1,
3   "message": "Data Anggota berhasil dihapus"
4 }
```

## TAMPIL ANGGOTA



[CONFLICT] GET Localhost/Perpustakaan... + ... No Environment

GET Localhost/Migrations/public/api/anggota Send Save

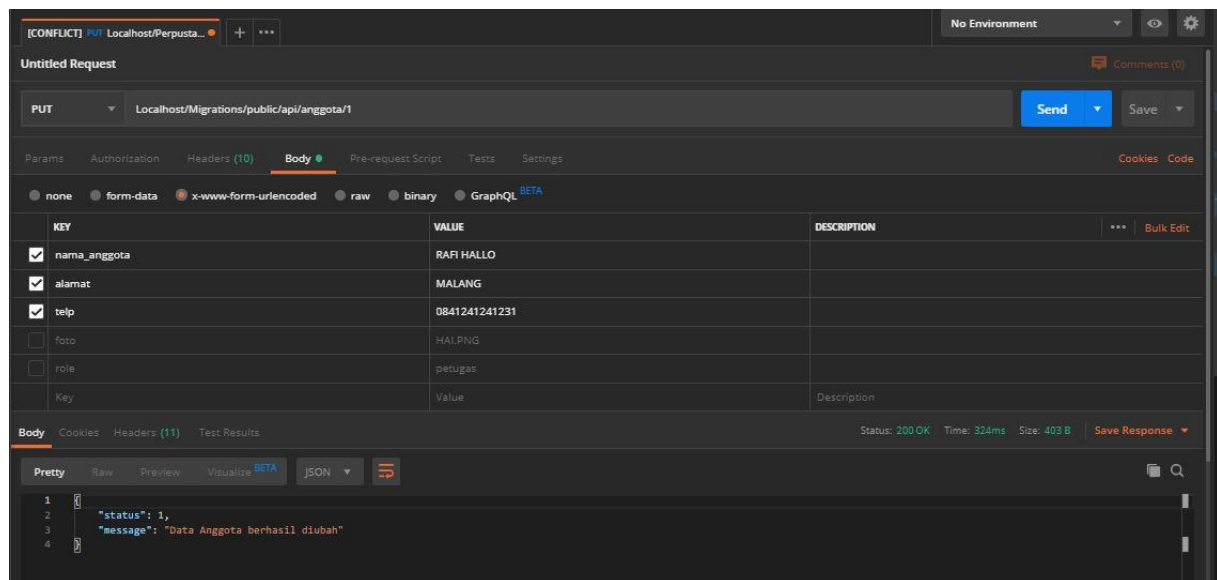
KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_anggota	RAFI HALLO	
<input checked="" type="checkbox"/> alamat	MALANG	
<input checked="" type="checkbox"/> telp	0841241241231	
<input type="checkbox"/> foto	HAL.PNG	
<input type="checkbox"/> role	petugas	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 249ms Size: 494 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "Data": [
3     {
4       "id_anggota": 1,
5       "nama_anggota": "RAFI HALLO",
6       "alamat": "MALANG",
7       "telp": "0841241241231",
8       "created_at": null,
9       "updated_at": null
10    }
11  ],
12  "status": 1
13 }
```

## UPDATE ANGGOTA



[CONFLICT] PUT Localhost/Perpustakaan... + ... No Environment

Untitled Request Comments (0)

PUT Localhost/Migrations/public/api/anggota/1 Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_anggota	RAFI HALLO	
<input checked="" type="checkbox"/> alamat	MALANG	
<input checked="" type="checkbox"/> telp	0841241241231	
<input type="checkbox"/> foto	HAL.PNG	
<input type="checkbox"/> role	petugas	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 324ms Size: 403 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": 1,
3   "message": "Data Anggota berhasil diubah"
4 }
```

## CREATE ANGGOTA

[CONFLICT] POST Localhost/Perpustakaan/anggota

Untitled Request

POST Localhost/Migrations/public/api/anggota

Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_anggota	RAFI	
<input checked="" type="checkbox"/> alamat	MALANG	
<input checked="" type="checkbox"/> telp	082222131251512	
<input checked="" type="checkbox"/> id_anggota	7	
<input type="checkbox"/> id_buku	1	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 447ms Size: 409 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": 1,
3   "message": "Data Anggota berhasil ditambahkan!"
4 }
```

## DELETE BUKU

[CONFLICT] DELETE Localhost/Perpustakaan/buku/1

Untitled Request

DELETE Localhost/Migrations/public/api/buku/1

Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> judul	HAI	
<input checked="" type="checkbox"/> penerbit	SULLE	
<input checked="" type="checkbox"/> pengarang	AHHA	
<input checked="" type="checkbox"/> foto	HA1.PNG	
<input type="checkbox"/> role	petugas	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 334ms Size: 401 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": 1,
3   "message": "Data Buku berhasil dihapus"
4 }
```

## UPDATE BUKU

The screenshot shows a REST client interface with a PUT request to `localhost/migrations/public/api/buku/1`. The request body is in `x-www-form-urlencoded` format with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> judul	HAI	
<input checked="" type="checkbox"/> penerbit	SULLE	
<input checked="" type="checkbox"/> pengarang	AHHA	
<input checked="" type="checkbox"/> foto	HAL.PNG	
<input type="checkbox"/> role	petugas	
Key	Value	Description

The response status is 200 OK, with a time of 418ms and a size of 400 B. The response body is in JSON format:

```
1 {
2   "status": 1,
3   "message": "Data Buku berhasil diubah"
4 }
```

## TAMPIL BUKU

The screenshot shows a REST client interface with a GET request to `localhost/migrations/public/api/buku`. The response status is 200 OK, with a time of 298ms and a size of 502 B. The response body is in JSON format:

```
1 {
2   "Data": [
3     {
4       "id_buku": 1,
5       "judul": "HELLO",
6       "penerbit": "BANGBANG",
7       "pengarang": "SUPRIYADI",
8       "foto": "HELLO.jpg",
9       "created_at": null,
10      "updated_at": null
11    }
12  ],
13   "status": 1
14 }
```

## CREATE BUKU

The screenshot shows a REST client interface with a POST request to `localhost/Migrations/public/api/buku`. The request body is in `form-data` format with the following fields:

KEY	VALUE	DESCRIPTION
judul	HELLO	
penerbit	BAMBANG	
pengarang	SUPRIYADI	
foto	HELLO.jpg	
id_buku	1	

The response is in JSON format, showing a status of 1 and a message: "Data Buku berhasil ditambahkan!".

```
{
  "status": 1,
  "message": "Data Buku berhasil ditambahkan!"
}
```

## CONTROLLER BUKU (2)

```
public function delete($id)
{
    $buku=Buku::where('id_buku',$id)->delete();
    if($buku){
        return Response()->json(['status'=>1,'message'=>'Data Buku berhasil dihapus']);
    }
    else{
        return Response()->json(['status'=>0,'message'=>'Data Buku tidak berhasil dihapus']);
    }
}

public function tampil()
{
    $buku=Buku::all();
    if($buku){
        return Response()->json(['Data'=>$buku,'status'=>1]);
    }
    else{
        return Response()->json(['status'=>0]);
    }
}
```

## BUKU CONTROLLER (1)

```
app > php > Controllers > BkController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Support\Facades\Validator;
6 use Illuminate\Http\Request;
7 use App\Buku;
8
9 class BkController extends Controller
10 {
11     public function store(Request $req)
12     {
13         $validator = Validator::make($req->all(),
14         [
15             'judul' => 'required',
16             'penulis' => 'required',
17             'penerbit' => 'required',
18             'penerbit' => 'required',
19             'foto' => 'required',
20         ]);
21     }
22     if($validator->fails()){
23         return Response()->json($validator->errors());
24     }
25     $buku = Buku::create([
26         'judul' => $req->judul,
27         'penulis' => $req->penulis,
28         'penerbit' => $req->penerbit,
29         'foto' => $req->foto,
30     ]);
31     if($buku){
32         return Response()->json(['status'=>1,'message'=>'Data Buku berhasil ditambahkan!']);
33     }
34     else{
35         return Response()->json(['status'=>0,'message'=>'Data Buku tidak berhasil ditambahkan!']);
36     }
37 }
38
39 public function update($id, Request $req)
40 {
41     $validator=Validator::make($req->all(),
42     [
43         'judul' => 'required',
44         'penulis' => 'required',
45         'penerbit' => 'required',
46         'foto' => 'required',
47     ]);
48     if($validator->fails()){
49         return Response()->json($validator->errors());
50     }
51     $buku=Buku::where('id_buku',$id)->update([
52         'judul' => $req->judul,
53         'penulis' => $req->penulis,
54         'penerbit' => $req->penerbit,
55         'foto' => $req->foto,
56     ]);
57     if($buku){
58         return Response()->json(['status'=>1,'message'=>'Data Buku berhasil diubah!']);
59     }
60     else{
61         return Response()->json(['status'=>0,'message'=>'Data Buku tidak berhasil diubah!']);
62     }
63 }
```

## ANGGOTA CONTROLLER (2)

```
public function update($id, Request $req)
{
    $validator=Validator::make($req->all(),
    [
        'nama_anggota' => 'required',
        'alamat' => 'required',
        'telp' => 'required',
    ]);
    if($validator->fails()){
        return Response()->json($validator->errors());
    }
    $anggota=Anggota::where('id_anggota',$id)->update([
        'nama_anggota' => $req->nama_anggota,
        'alamat' => $req->alamat,
        'telp' => $req->telp,
    ]);
    if($anggota){
        return Response()->json(['status'=>1,'message'=>'Data Anggota berhasil diubah!']);
    }
    else{
        return Response()->json(['status'=>0,'message'=>'Data Anggota tidak berhasil diubah!']);
    }
}

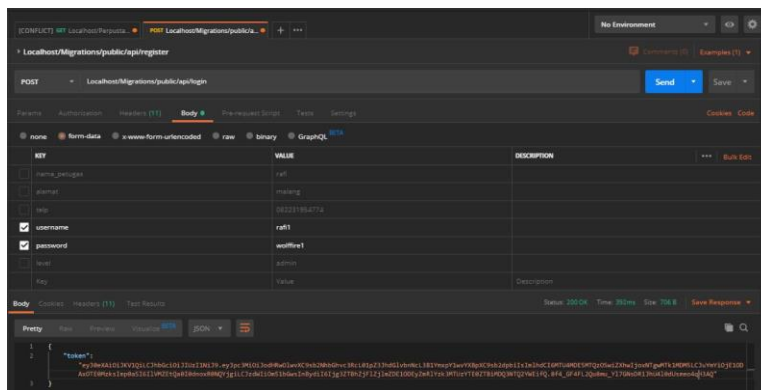
public function delete($id)
{
    $anggota=Anggota::where('id_anggota',$id)->delete();
    if($anggota){
        return Response()->json(['status'=>1,'message'=>'Data Anggota berhasil dihapus!']);
    }
    else{
        return Response()->json(['status'=>0,'message'=>'Data Anggota tidak berhasil dihapus!']);
    }
}

public function tampil()
{
    $anggota=Anggota::all();
    if($anggota){
        return Response()->json(['Data'=>$anggota,'status'=>1]);
    }
    else{
        return Response()->json(['status'=>0]);
    }
}
```

## ANGGOTA CONTROLLER

```
app > http > Controllers > AnggotaController.php
1
2
3
4 namespace App\Http\Controllers;
5
6 use Illuminate\Support\Facades\Validator;
7 use Illuminate\Http\Request;
8 use App\Anggota;
9
10 class AnggotaController extends Controller
11 {
12     public function store(Request $req)
13     {
14         $validator = Validator::make($req->all(), [
15             'nama_anggota' => 'required',
16             'alamat' => 'required',
17             'telp' => 'required',
18         ]);
19         if($validator->fails()){
20             return Response->json($validator->errors());
21         }
22         $anggota = Anggota::create([
23             'nama_anggota' => $req->nama_anggota,
24             'alamat' => $req->alamat,
25             'telp' => $req->telp,
26         ]);
27         if($anggota){
28             return Response->json(['status' => '1', 'message' => 'Data Anggota berhasil ditambahkan!']);
29         }
30         else{
31             return Response->json(['status' => '0', 'message' => 'Data Anggota tidak berhasil ditambahkan!']);
32         }
33     }
34
35     public function update($id, Request $req)
36     {
37         $validator = Validator::make($req->all(), [
38             'nama_anggota' => 'required',
39             'alamat' => 'required',
40             'telp' => 'required',
41         ]);
42         if($validator->fails()){
43             return Response->json($validator->errors());
44         }
45         $anggota = Anggota::where('id_anggota', $id)->update([
46             'nama_anggota' => $req->nama_anggota,
47             'alamat' => $req->alamat,
48             'telp' => $req->telp,
49         ]);
50         if($anggota){
51             return Response->json(['status' => '1', 'message' => 'Data Anggota berhasil diubah!']);
52         }
53         else{
54             return Response->json(['status' => '0', 'message' => 'Data Anggota tidak berhasil diubah!']);
55         }
56     }
57 }
```

## LOGIN DAPAT TOKEN



## REGISTER

