

Nama : Muhammad Bagas Ramadhan

Absen : 19

## Controller Anggota

```
namespace App\Http\Controllers;
use Auth;
use App\Buku_model;
use App\User;
use App\Anggota_model;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use JWTAuth;
use Tymon\JWTAuth\Exceptions\JWTException;

class anggota extends Controller
{
    public function simpan(Request $req)
    {
        $validator=Validator::make($req->all(),
        [
            'nama_anggota'=>'required',
            'alamat'=>'required',
            'telp'=>'required'
        ]);
        if($validator->fails()){
            return Response()->json($validator->errors());
        }
        $simpan=Anggota_model::create([
            'nama_anggota'=>$req->nama_anggota,
            'alamat'=>$req->alamat,
            'telp'=>$req->telp
        ]);
        if($simpan){
            $pesan['message'] = 'berhasil tambah anggota broh';
            return Response()->json($pesan);
        } else {
            $pesan['message'] = 'gagal tambah anggota broh';
            return Response()->json($pesan);
        }
    }
    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());
        }
        $ubah=Anggota_model::where('id',$id)->update([
            'nama_anggota'=>$req->nama_anggota,
```

```

        'alamat'=>$req->alamat,
        'telp'=>$req->telp
    ));
    if($ubah){
        $data['message'] = 'Anggota Berhasil diubah';
        return Response()->json($data);
    } else {
        $data['message'] = 'Anggota gagal diubah';
        return Response()->json($data);
    }
}

public function destroy($id)
{
    $hapus=Anggota_model::where('id',$id)->delete();
    if($hapus){
        $pesan['message'] = 'Anggota berhasil dihapus';
        return Response()->json($pesan);
    }else{
        $pesan['message'] = 'Anggota gagal dihapus';
        return Response()->json($pesan);
    }
}

public function tampil_anggota()
{
    if(Auth::User()->level=="admin"){
        $dt_buku=Anggota_model::get();
        return Response()->json($dt_buku);
    } else {
        $pesan['message'] = 'kamu buka admin bro maaf';
        return Response()->json($pesan);
    }
}
}

```

## Controller Buku

```
class buku extends Controller
{
    public function store(Request $req)
    {
        $validator=Validator::make($req->all(),
        [
            'judul'=>'required',
            'penerbit'=>'required',
            'pengarang'=>'required',
            'foto'=>'required',
        ]);
        if($validator->fails()){
            return Response()->json($validator->errors());
        }
        $simpan=Buku_model::create([
            'judul'=>$req->judul,
            'penerbit'=>$req->penerbit,
            'pengarang'=>$req->pengarang,
            'foto'=>$req->foto
        ]);
        if($simpan){
            $pesan['message'] = 'berhasil tambah buku broh';
            return Response()->json($pesan);
        } else {
            $pesan['message'] = 'gagal tambah buku broh';
            return Response()->json($pesan);
        }
    }
    public function update($id,Request $req)
    {
        $validator=Validator::make($req->all(),
        [
            'judul'=>'required',
            'pengarang'=>'required',
            'penerbit'=>'required',
            'foto'=>'required',
        ]);
        if($validator->fails()){
            return Response()->json($validator->errors());
        }
        $ubah=Buku_model::where('id',$id)->update([
            'judul'=>$req->judul,
            'pengarang'=>$req->pengarang,
            'penerbit'=>$req->penerbit,
            'foto'=>$req->foto
        ]);
    }
}
```

```

    }
    if($ubah){
        $data['message'] = 'Buku Berhasil diubah';
        return Response()->json($data);
    } else {
        $data['message'] = 'Buku gagal diubah';
        return Response()->json($data);
    }
}
public function destroy($id)
{
    $hapus=Buku_model::where('id',$id)->delete();
    if($hapus){
        $pesan['message'] = 'Buku berhasil dihapus';
        return Response()->json($pesan);
    }else{
        $pesan['message'] = 'Buku gagal dihapus';
        return Response()->json($pesan);
    }
}
public function index()
{
    if(Auth::User()->level=="admin"){
        $dt_buku=Buku_model::get();
        return Response()->json($dt_buku);
    } else {
        $pesan['message'] = 'kamu buka admin bro maaf';
        return Response()->json($pesan);
    }
}
}

```

## Model Anggota

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Anggota_model extends Model
{
    protected $table="anggota";
    protected $primaryKey="id";
    public $timestamps=false;
    protected $fillable = [
        'nama_anggota', 'alamat', 'telp'
    ];
}

```

## Model Buku

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Buku_model extends Model
{
    protected $table="buku";
    protected $primaryKey="id";
    public $timestamps=false;
    protected $fillable = [
        'judul', 'penerbit', 'pengarang', 'foto'
    ];
}

```

## Test postman

### 1. tambah anggota

The screenshot shows the Postman interface for an 'Untitled Request'. The request method is 'POST' and the URL is 'http://localhost/perpustakaan/public/api/simpan\_anggota'. The 'Body' tab is selected, and the 'form-data' radio button is chosen. A table lists the form data fields: 'nama\_anggota' with value 'tes', 'alamat' with value 'jl tess', and 'telp' with value '081263'. The response status is '200 OK' with a time of '942ms' and size of '394 B'. The response body is displayed in JSON format: 

```
{  "message": "berhasil tambah anggota broh"}
```

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_anggota	tes	
<input checked="" type="checkbox"/> alamat	jl tess	
<input checked="" type="checkbox"/> telp	081263	
<input type="checkbox"/>	-	
Key	Value	Description

```
1 {
2   "message": "berhasil tambah anggota broh"
3 }
```

### 2. edit anggota

The screenshot shows the Postman interface for an 'Untitled Request'. The request method is 'PUT' and the URL is 'http://localhost/perpustakaan/public/api/ubah\_anggota/1'. The 'Body' tab is selected, and the 'x-www-form-urlencoded' radio button is chosen. A table lists the form data fields: 'nama\_anggota' with value 'ganti', 'alamat' with value 'jl ganti', and 'telp' with value '0894571'. The response status is '200 OK' with a time of '510ms' and size of '368 B'. The response body is displayed in JSON format: 

```
{  "message": "Anggota Berhasil diubah"}
```

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama_anggota	ganti	
<input checked="" type="checkbox"/> alamat	jl ganti	
<input checked="" type="checkbox"/> telp	0894571	
<input type="checkbox"/> foto	-	
Key	Value	Description

```
1 {
2   "message": "Anggota Berhasil diubah"
3 }
```

### 3. hapus anggota

Postman interface showing a DELETE request to `http://localhost/perpustakaan/public/api/hapus_anggota/1`. The response status is 200 OK, Time: 248ms, Size: 369 B. The response body is a JSON object:

```
{  "message": "Anggota berhasil dihapus"}
```

### 4. read anggota

Postman interface showing a GET request to `http://localhost/perpustakaan/public/api/tampil_petugas`. The request has an Authorization header with a bearer token. The response status is 200 OK, Time: 319ms, Size: 1.56 KB. The response body is a JSON object:

```
{  "id": 1,  "nama_petugas": "palako",  "alamat": "jt. kebebasan",  "telp": "120120",  "username": "palakao",  "level": "admin",  "created_at": null,  "updated_at": "2020-01-27 07:00:10"}
```

## 5. tambah buku

The screenshot shows the Postman interface for an "Untitled Request". The request method is **POST** and the URL is `http://localhost/perpustakaan/public/api/simpan_buku`. The "Body" tab is selected, and the "form-data" type is chosen. A table of form data is visible:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> judul	Gangsal	
<input checked="" type="checkbox"/> pengarang	Lawrents	
<input checked="" type="checkbox"/> penerbit	Edustar	
<input checked="" type="checkbox"/> foto	-	
Key	Value	Description

The status bar indicates a successful response: **Status: 200 OK**, Time: 10.71s, Size: 391 B. The response body is shown in the "Body" tab, displaying a JSON message: `"message": "berhasil tambah buku broh"`.

## 6. edit buku

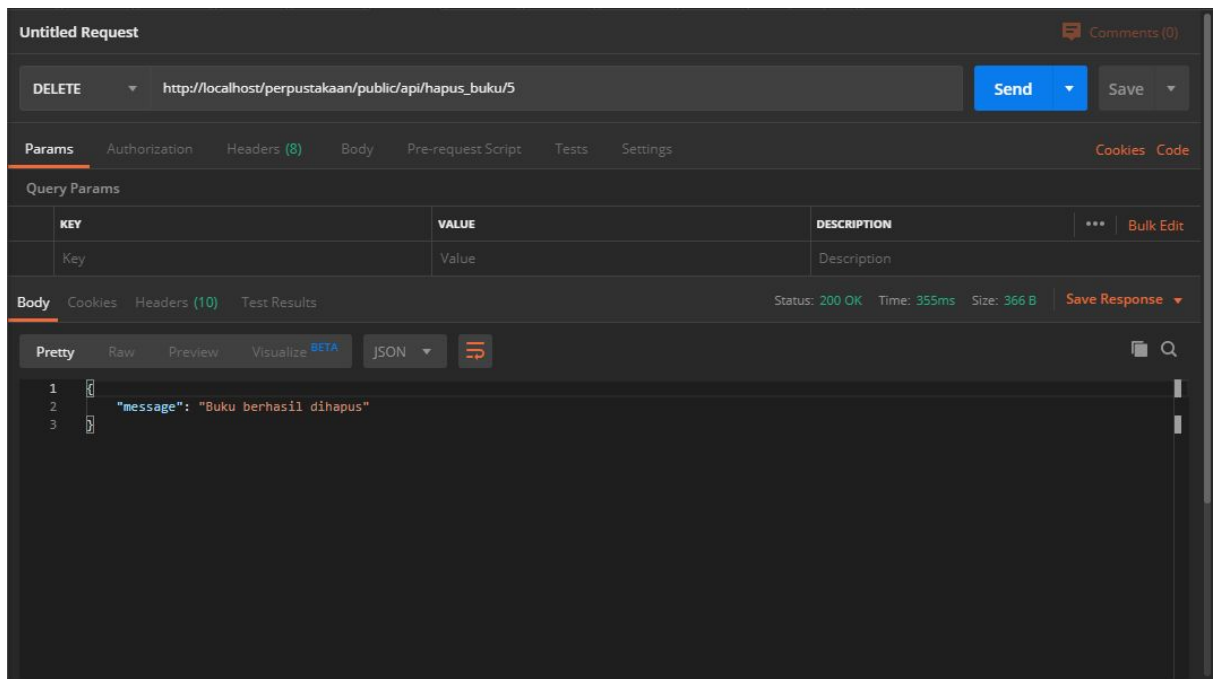
The screenshot shows the Postman interface for an "Untitled Request". The request method is **PUT** and the URL is `http://localhost/perpustakaan/public/api/ubah_buku/6`. The "Body" tab is selected, and the "x-www-form-urlencoded" type is chosen. A table of form data is visible:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> judul	Lima	
<input checked="" type="checkbox"/> penerbit	Postar	
<input checked="" type="checkbox"/> pengarang	Lajo	
<input checked="" type="checkbox"/> foto	--	
Key	Value	Description

The status bar indicates a successful response: **Status: 200 OK**, Time: 384ms, Size: 365 B. The response body is shown in the "Body" tab, displaying a JSON message: `"message": "Buku Berhasil diubah"`.

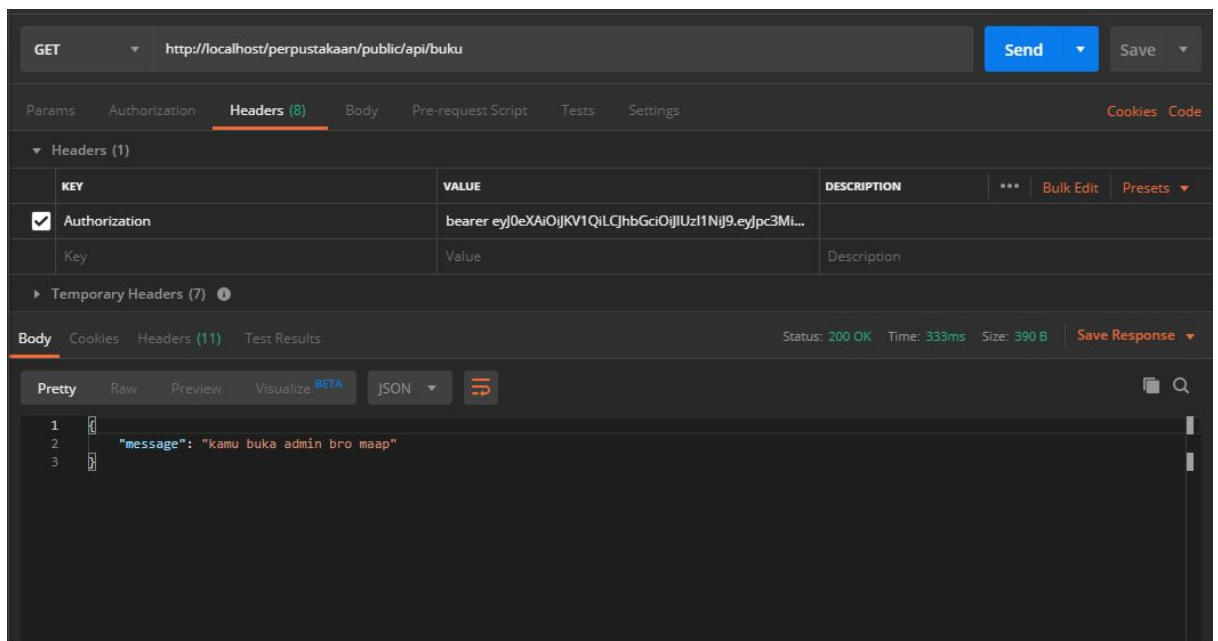


## 7. hapus buku



## 8. read buku (dengan authentication)

jika bukan admin





jika admin

Untitled Request Comments (0)

GET ▼ http://localhost/perpustakaan/public/api/buku Send Save ▼

Params Authorization **Headers (8)** Body Pre-request Script Tests Settings Cookies Cod

▼ Headers (1)

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	bearer eyJ0eXAiOiJV1QjLQjhbGciOiJIUzI1NiJ9.eyJpc3Mi...				
	Key	Value	Description			

► Temporary Headers (7) ⓘ

**Body** Cookies Headers (11) Test Results Status: 200 OK Time: 303ms Size: 943 B Save Response ▼

Pretty Raw Preview Visualize BETA JSON ▼ ≡

```
1  [
2    {
3      "id": 2,
4      "judul": "Simakalama",
5      "penerbit": "upost",
6      "pengarang": "kilua",
7      "foto": "-",
8      "created_at": null,
9      "updated_at": null
10   },
11   {
12     "id": 3,
13     "judul": "Jodo Singarung"
```