CRUD Buku dan Anggota

1. API

```
Route::post('tambah_buku', 'BukuController@store')->middleware('jwt.verify');

Route::put('edit_buku/{id}', 'BukuController@update')->middleware('jwt.verify');

Route::delete('hapus_buku/{id}', 'BukuController@destroy')->middleware('jwt.verify');

Route::get('tampil_buku', 'BukuController@index')->middleware('jwt.verify');

Route::post('tambah_anggota', 'AnggotaController@store')->middleware('jwt.verify');

Route::put('edit_anggota/{id}', 'AnggotaController@update')->middleware('jwt.verify');

Route::delete('hapus_anggota/{id}', 'AnggotaController@destroy')->middleware('jwt.verify');

Route::get('tampil_anggota', 'AnggotaController@index')->middleware('jwt.verify');
```

2. Model Buku

3. Controller Buku

```
namespace App\Http\Controllers;
use App\ModelBuku;
use Illuminate\Http\Request;
use Validator;
use JWTAuth;
use Auth;
class BukuController extends Controller
     * @return \Illuminate\Http\Response
    public function index()
        if(Auth::user()->level="admin") {
        $data=ModelBuku::all();
        return response()->json($data);
          echo "Mohon maaf, data hanya bisa diakses oleh Admin";
```

```
public function store(Request $request)
   if(Auth::user()->level=="admin") {
   $validator=Validator::make($request->all(),[
      'judul'=>'required',
     'penerbit'=>'required',
      'pengarang'=>'required',
     'foto'=>'required'
   if($validator->fails()){
     return response()->json($validator->errors()->toJson(),400);
     $insert=ModelBuku::insert([
        'judul'=>$request->judul,
       'penerbit'=>$request->penerbit,
        'pengarang'=>$request->pengarang,
       'foto'=>$request->foto
     if($insert){
       $status="gagal";
     return response()->json(compact('status'));
   echo "Mohon maaf, data hanya bisa diakses oleh Admin";
```

```
public function update(Request $request, $id)
   if(Auth::user()->level=="admin") {
   $validator=Validator::make($request->all(),
     'judul'=>'required',
     'penerbit'=>'required',
     'pengarang'=>'required',
     'foto'=>'required'
 if($validator->fails()) {
   return response()->json($validator->errors());
 $ubah=ModelBuku::where('id', $id)->update([
    'judul'=>$request->judul,
   'penerbit'=>$request->penerbit,
    'pengarang'=>$request->pengarang,
   'foto'=>$request->foto
 if($ubah){
   return response()->json(['status'=>1]);
   return response()->json(['status'=>0]);
 echo "Mohon maaf, data hanya bisa diakses oleh Admin";
```

```
public function destroy($id)

{

//

if(Auth::user()->level=="admin") {

$hapus=ModelBuku::where('id',$id)->delete();

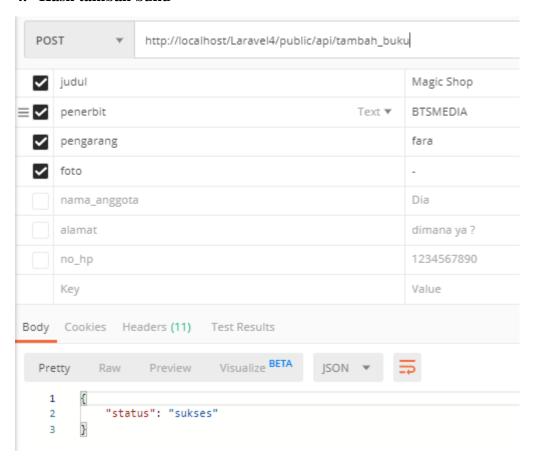
if($hapus){
    return response()->json(['status'=>1]);

}

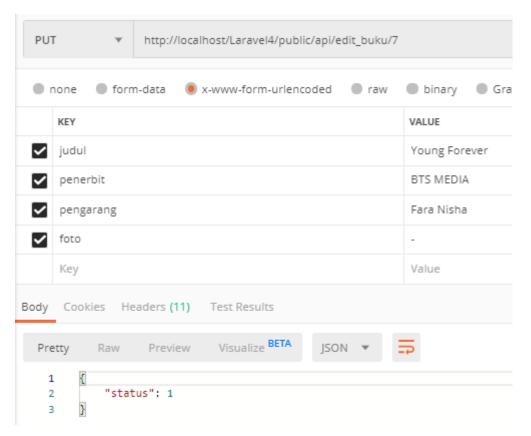
else {
    return response()->json(['status'=>0]);

}
```

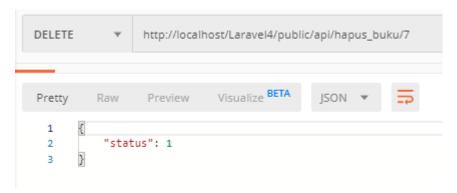
4. Hasil tambah buku



5. Hasil edit buku



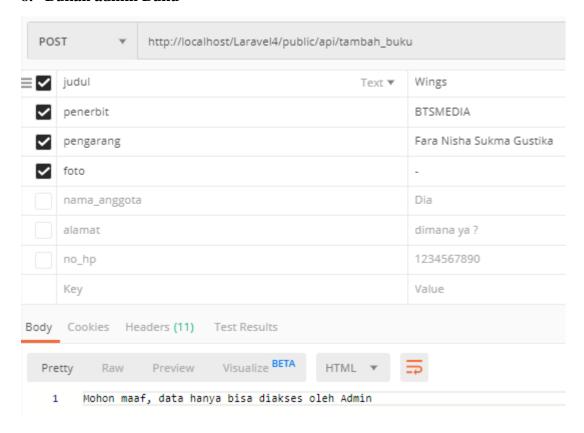
6. Hasil hapus buku



7. Hasil tampil buku

```
GET
                    http://localhost/Laravel4/public/api/tampil_buku
body cookies ricaders (11) reservesures
                                Visualize BETA
  Pretty
                    Preview
                                                 JSON
            Raw
 11
                 "id": 2,
   12
                 "judul": "Bintang",
   13
                 "penerbit": "Butterfly Media",
   14
                 "pengarang": "Fara Nisha",
   15
                 "foto": "-",
   16
                 "created_at": "2020-01-15 07:51:44",
   17
                 "updated_at": null
   18
   19
             },
   20
                 "id": 3,
   21
   22
                 "judul": "Matahari",
                 "penerbit": "Butterfly Media",
   23
                 "pengarang": "Fara Nisha",
   24
                 "foto": "-",
   25
                 "created_at": "2020-01-15 07:51:44",
   26
                 "updated_at": null
   27
   28
             },
   29
                 "id": 4,
   30
                 "judul": "Purple",
   31
                 "penerbit": "Butterfly Media",
   32
                 "pengarang": "Fara Nisha",
   33
                 "foto": "-",
   34
                 "created_at": "2020-01-15 07:51:44",
   35
                 "updated_at": null
   36
   37
```

8. Bukan admin Buku



9. Model Anggota

10. Controller Anggota

```
a namespace App\Http\Controllers;

use App\ModelAnggota;

use Illuminate\Http\Request;

use Validator;

use JWTAuth;

use Auth;

class AnggotaController extends Controller

{
    /**
    * Display a listing of the resource.

*

* @return \Illuminate\Http\Response

*/

public function index()

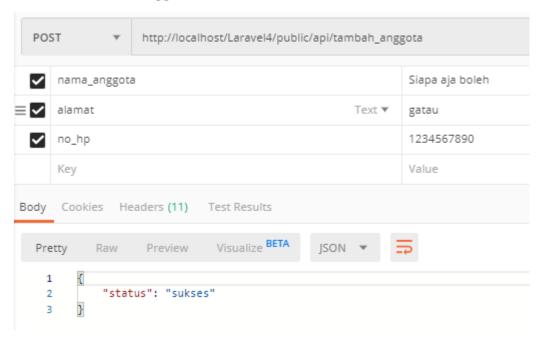
{
    //
    if(Auth::user()->level="admin") {
    $data=ModelAnggota::all();
    return response()->json($data);
}

else {
    echo "Mohon maaf, data hanya bisa diakses oleh Admin";
}
```

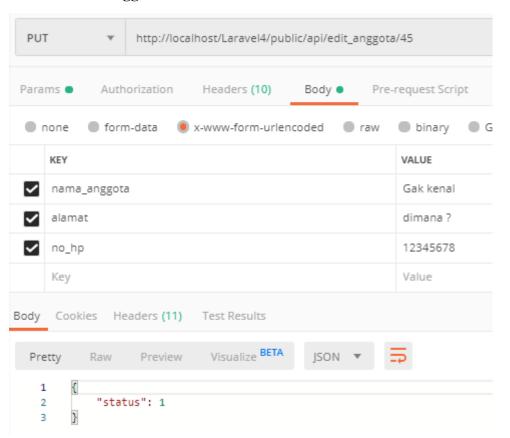
```
public function store(Request $request)
   if(Auth::user()->level=="admin") {
   $validator=Validator::make($request->all(),[
      'nama_anggota'=>'required',
      'alamat'=>'required',
      'no_hp'=>'required'
   1);
    if($validator->fails()){
     return response()->json($validator->errors()->toJson(),400);
     $insert=ModelAnggota::insert([
        'nama_anggota'=>$request->nama_anggota,
        'alamat'=>$request->alamat,
        'no_hp'=>$request->no_hp
     ]);
     if($insert){
       $status="sukses";
        $status="gagal";
     return response()->json(compact('status'));
   echo "Mohon maaf, data hanya bisa diakses oleh Admin";
```

```
public function update(Request $request, $id)
   if(Auth::user()->level=="admin") {
   $validator=Validator::make($request->all(),
      'nama_anggota'=>'required',
      'alamat'=>'required',
      'no hp'=>'required'
 if($validator->fails()) {
    return response()->json($validator->errors());
 $ubah=ModelAnggota::where('id', $id)->update([
    'nama_anggota'=>$request->nama_anggota,
    'alamat'=>$request->alamat,
    'no hp'=>$request->no hp
 ]);
 if($ubah){
    return response()->json(['status'=>1]);
   return response()->json(['status'=>0]);
 echo "Mohon maaf, data hanya bisa diakses oleh Admin";
public function destroy($id)
    if(Auth::user()->level=="admin") {
    $hapus=ModelAnggota::where('id',$id)->delete();
    if($hapus){
      return response()->json(['status'=>1]);
      return response()->json(['status'=>0]);
    echo "Mohon maaf, data hanya bisa diakses oleh Admin";
```

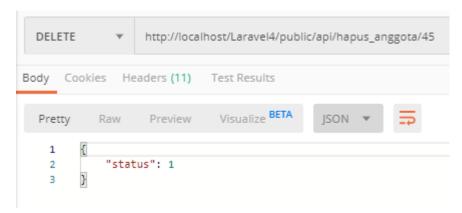
11. Hasil tambah Anggota



12. Hasil edit Anggota



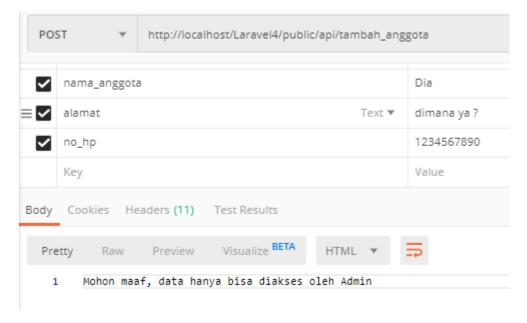
13. Hasil hapus Anggota



14. Hasil tampil Anggota

```
GET
                    http://localhost/Laravel4/public/api/tampil_anggota
Body Cookies Headers (11) Test Results
                                 Visualize BETA
  Pretty
            Raw
                     Preview
                                                  JSON
    2
                  "id": 38,
    3
    4
                  "nama_anggota": "Kim Taehyung",
    5
                  "alamat": "Daegu",
                  "no_hp": "0987654321",
    6
                  "created_at": "2020-01-15 08:04:48",
                  "updated_at": null
    8
   10
                  "id": 39,
   11
                 "nama_anggota": "Fara Nisha",
   12
                 "alamat": "Daegu",
   13
                 "no_hp": "0987612345",
   14
                  "created_at": "2020-01-15 08:04:48",
   15
                  "updated_at": null
   16
   17
             },
   18
                  "id": 40,
   19
                  "nama_anggota": "Jeon Jungkook",
   20
                  "alamat": "Seoul",
   21
                  "no_hp": "0981234567",
   22
                  "created_at": "2020-01-15 08:04:48",
   23
                  "updated_at": null
   24
             },
   25
   26
             {
                  "id": 41,
   27
                  "nama_anggota": "Park Jimin",
   28
                  "alamat": "Seoul",
   29
```

15. Bukan admin Anggota



Nama : Fara Nisha Sukma Gustika

Kelas : Laravel