

1. Hapus anggota

Untitled Request

DELETE localhost:8000/api/hapus_anggota/7

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

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

KEY	VALUE
<input checked="" type="checkbox"/> nama_anggota	kandiya
<input checked="" type="checkbox"/> alamat	Sukun
<input checked="" type="checkbox"/> telp	123456789
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1 {  
2   "status": 1  
3 }
```

2. Hapus buku

DELETE localhost:8000/api/hapus_buku/6

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

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

KEY	VALUE
<input checked="" type="checkbox"/> judul	Uwu Bet
<input checked="" type="checkbox"/> penerbit	Gramedia
<input checked="" type="checkbox"/> pengarang	Sixty
<input checked="" type="checkbox"/> foto	-
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1 {  
2   "status": 1  
3 }
```

3. Simpan anggota

APIs BETA

Clear all

POST localhost:8000/api/simpan_anggota

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

KEY	VALUE
<input checked="" type="checkbox"/> nama_anggota	Kandia
<input checked="" type="checkbox"/> alamat	Sukun
<input checked="" type="checkbox"/> telp	0864682632
<input type="checkbox"/> foto	-
<input type="checkbox"/> password	aulia123
<input type="checkbox"/> password_confirmation	Text aulia123
<input type="checkbox"/> level	admin
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": 1
3 }
```

4. Simpan buku

APIs BETA

Clear all

POST localhost:8000/api/simpan_buku

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

KEY	VALUE
<input checked="" type="checkbox"/> judul	Gaskeun lur
<input checked="" type="checkbox"/> penerbit	Gramedia
<input checked="" type="checkbox"/> pengarang	Text Niansa
<input checked="" type="checkbox"/> foto	-
<input type="checkbox"/> password	aulia123
<input type="checkbox"/> password_confirmation	aulia123
<input type="checkbox"/> level	admin
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": 1
3 }
```

5. Tampil anggota

The screenshot shows a REST client interface with a GET request to `localhost:8000/api/tampil_anggota`. The response is a JSON object with a `count` of 5 and an `arr_data` array containing 5 member objects. The members are Jinjin, Aulia, Rizky, and Eunwoo, with the fifth member's details partially obscured by a scrollbar.

Key	Value
GET	localhost:8000/api/tampil_anggota

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "count": 5,
3   "arr_data": [
4     {
5       "id": 2,
6       "nama_anggota": "Jinjin",
7       "alamat": "Malang",
8       "telp": "0812345679"
9     },
10    {
11      "id": 3,
12      "nama_anggota": "Aulia",
13      "alamat": "Malang",
14      "telp": "0278671312"
15    },
16    {
17      "id": 4,
18      "nama_anggota": "Rizky",
19      "alamat": "Lamongan",
20      "telp": "98777371931"
21    },
22    {
23      "id": 5,
24      "nama_anggota": "Eunwoo",
25      "alamat": "Seoul",
26      "telp": "9831973191"
```

6. Tampil buku

GET localhost:8... GET localhost:8... GET localhost:8... DEL localhost:8... POS

GET localhost:8000/api/tampil_buku

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "count": 5,
3   "arr_data": [
4     {
5       "id": 1,
6       "judul": "Laskar pelangi",
7       "penerbit": "Gramedia",
8       "pengarang": "Niansa",
9       "foto": "-"
10    },
11    {
12      "id": 2,
13      "judul": "Dilan 1990",
14      "penerbit": "Gramedia",
15      "pengarang": "Aulia",
16      "foto": "-"
17    },
18    {
19      "id": 3,
20      "judul": "Harry Potter",
21      "penerbit": "Gramedia",
22      "pengarang": "Della",
23      "foto": "-"
24    },
25    {
26      "id": 4,
27      "judul": "Detektif Conan",
28      "penerbit": "Gramedia",
29      "pengarang": "Heni",
30    }
31  ]
32 }
```

7. Ubah anggota

PUT localhost:8000/api/ubah_anggota/7

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

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

	KEY	VALUE
<input checked="" type="checkbox"/>	nama_anggota	kandiya
<input checked="" type="checkbox"/>	alamat	Sukun
<input checked="" type="checkbox"/>	telp	123456789
	Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1 {  
2   "status": 1  
3 }
```

8. Ubah buku

ETA

all

Untitled Request

PUT localhost:8000/api/ubah_buku/6

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

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

	KEY	VALUE
<input checked="" type="checkbox"/>	judul	Uwu Bet
<input checked="" type="checkbox"/>	penerbit	Gramedia
<input checked="" type="checkbox"/>	pengarang	Sixty
<input checked="" type="checkbox"/>	foto	-
	Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize **BETA** JSON

```
1 {  
2   "status": 1  
3 }
```

9. Buku Controller

```
app > Http > Controllers > AnggotaController.php
11 class AnggotaController extends Controller
12 {
13     public function store(Request $req)
14     {
15         $validator=Validator::make($req->all(),
16         [
17             'nama_anggota'=>'required',
18             'alamat'=>'required',
19             'telp'=>'required'
20         ]
21         );
22         if($validator->fails()){
23             return Response()->json($validator->errors());
24         }
25
26         $simpan=Anggota_model::create([
27             'nama_anggota'=>$req->nama_anggota,
28             'alamat'=>$req->alamat,
29             'telp'=>$req->telp
30         ]);
31         if($simpan){
32             return Response()->json(['status'=>1]);
33         } else {
34             return Response()->json(['status'=>0]);
35         }
36     }
37
38     public function update($id, Request $req)
39     {
40         $validator=Validator::make($req->all(),
41         [
42             'nama_anggota'=>'required',
43             'alamat'=>'required',
44             'telp'=>'required'
45         ]
46         );
47         if($validator->fails()){
48             return Response()->json($validator->errors());
49         }
50
51         $ubah=Anggota_model::where('id', $id)->update([
52             'nama_anggota'=>$req->nama_anggota,
53
54         $ubah=Anggota_model::where('id', $id)->update([
55             'nama_anggota'=>$req->nama_anggota,
56             'alamat'=>$req->alamat,
57             'telp'=>$req->telp
58         ]);
59         if($ubah){
60             return Response()->json(['status'=>1]);
61         } else {
62             return Response()->json(['status'=>0]);
63         }
64     }
65
66     public function hapus($id){
67         $hapus=Anggota_model::where('id', $id)->delete();
68         if($hapus){
69             return Response()->json(['status'=>1]);
70         } else {
71             return Response()->json(['status'=>0]);
72         }
73     }
74
75     public function tampil(){
76         $data_anggota = Anggota_model::get();
77         $count = $data_anggota->count();
78         $arr_data = array();
79         foreach ($data_anggota as $dt_ang){
80             $arr_data[] = array(
81                 'id' => $dt_ang->id,
82                 'nama_anggota' => $dt_ang->nama_anggota,
83                 'alamat' => $dt_ang->alamat,
84                 'telp' => $dt_ang->telp,
85             );
86         }
87         $status=1;
88         return Response()->json(compact('count','count', 'arr_data'));
```

10. Anggota Controller

```
File Edit Selection View Go Debug Terminal Help
BukuController.php - perpus2 - Visual Studio Code
app > Http > Controllers > BukuController.php
13 {
14     public function store(Request $req)
15     {
16         $validator=Validator::make($req->all(),
17         [
18             'judul'=>'required',
19             'penerbit'=>'required',
20             'pengarang'=>'required',
21             'foto'=>'required'
22         ]
23         );
24         if($validator->fails()){
25             return Response()->json($validator->errors());
26         }
27
28         $simpan=Buku_model::create([
29             'judul'=>$req->judul,
30             'penerbit'=>$req->penerbit,
31             'pengarang'=>$req->pengarang,
32             'foto'=>$req->foto
33         ]);
34         if($simpan){
35             return Response()->json(['status'=>1]);
36         } else {
37             return Response()->json(['status'=>0]);
38         }
39     }
40
41     public function update($id, Request $req)
42     {
43         $validator=Validator::make($req->all(),
44         [
45             'judul'=>'required',
46             'penerbit'=>'required',
47             'pengarang'=>'required',
48             'foto'=>'required'
49         ]
50         );
51         if($validator->fails()){
52             return Response()->json($validator->errors());
53         }
54
55         $ubah=Buku_model::where('id', $id)->update([
56             'judul'=>$req->judul,
57             'penerbit'=>$req->penerbit,
58             'pengarang'=>$req->pengarang,
59             'foto'=>$req->foto
60         ]);
61         if($ubah){
62             return Response()->json(['status'=>1]);
63         } else {
64             return Response()->json(['status'=>0]);
65         }
66     }
67
68     public function hapus($id){
69         $hapus=Buku_model::where('id', $id)->delete();
70         if($hapus){
71             return Response()->json(['status'=>1]);
72         } else {
73             return Response()->json(['status'=>0]);
74         }
75     }
76
77     public function tampil(){
78         $data_buku = Buku_model::get();
79         $count = $data_buku->count();
80         $arr_data = array();
81         foreach ($data_buku as $dt_buku){
82             $arr_data[] = array(
83                 'id' => $dt_buku->id,
84                 'judul' => $dt_buku->judul,
85                 'penerbit' => $dt_buku->penerbit,
86                 'pengarang' => $dt_buku->pengarang,
87                 'foto' => $dt_buku->foto,
88             );
89         }
90         $status=1;
91         return Response()->json(compact('count','count', 'arr_data'));
```