

CRUD Anggota & Buku

1. Anggota

- Create

```
public function create(Request $req){
    $validator = Validator::make($req->all(),
        [
            'nama_anggota' => 'required',
            'alamat' => 'required',
            'telp' => 'required'
        ]
    );

    if($validator->fails()){
        return Response()->json($validator->errors());
    }

    $simpan = ModelAnggota::create([
        'nama_anggota'=>$req->nama_anggota,
        'alamat'=>$req->alamat,
        'telp'=>$req->telp,
    ]);

    if($simpan){
        return Response()->json(['status'=>1]);
    } else {
        return Response()->json(['status'=>0]);
    }
}
```

POST http://localhost:8000/api/register_anggota Send Save

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

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

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	nama_anggota	nuril			
<input checked="" type="checkbox"/>	alamat	jl.pisang			
<input checked="" type="checkbox"/>	telp	121212			
	Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 1042ms Size: 290 B Save Response

Pretty Raw Preview Visualize BETA JSON ≡

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

- Read

```
public function tampil(){
    $stampil = ModelAnggota::get();

    if($stampil){
        return Response()->json([$stampil]);
    } else {
        return Response()->json([$stampil->errors()]);
    }
}
```

GET http://localhost:8000/api/tampil_anggota Send Save

Body Cookies Headers (9) Test Results Status: 200 OK Time: 0.091ms Size: 1.01 KB Save response

Pretty Raw Preview Visualize BETA JSON

```
1  [
2
3    {
4      "id_anggota": 11,
5      "nama_anggota": "Ayam",
6      "alamat": "Malang",
7      "telp": 123456789,
8      "created_at": null,
9      "updated_at": null
10   },
11   {
12     "id_anggota": 12,
13     "nama_anggota": "Burung",
14     "alamat": "Surabaya",
15     "telp": 123456789,
16     "created_at": null,
17     "updated_at": null
18   },
19   {
20     "id_anggota": 13,
21     "nama_anggota": "Merak",
22     "alamat": "Bandung",
23     "telp": 123456789,
24     "created_at": null,
25     "updated_at": null
26   },
27   {
```

- Update

```
public function edit($id, Request $req){
    $validator = Validator::make($req->all(),
        [
            'nama_anggota'=>'required',
            'alamat'=>'required',
            'telp'=>'required'
        ]
    );

    if($validator->fails()){
        return Response()->json($validator->errors());
    }

    $edit = ModelAnggota::where('id_anggota',$id)->update([
        'nama_anggota'=>$req->nama_anggota,
        'alamat'=>$req->alamat,
        'telp'=>$req->telp
    ]);

    if($edit){
        return Response()->json(['status'=>1]);
    } else {
        return Response()->json(['status'=>0]);
    }
}
```

POST
http://localhost:8000/api/edit_anggota/16
Send
Save

Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
Code

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

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	nama_anggota	nuril haider			
<input checked="" type="checkbox"/>	alamat	jl.pisang			
<input checked="" type="checkbox"/>	telp	121212			
	Key	Value	Description		

Body
Cookies
Headers (9)
Test Results
Status: 200 OK
Time: 952ms
Size: 290 B
Save Response

Pretty
Raw
Preview
Visualize BETA
JSON

```

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

- Delete

```
public function hapus($id){  
    $hapus = ModelAnggota::where('id_anggota',$id)->delete();  
  
    $a = "Data Berhasil Dihapus";  
    $b = "Data Gagal Dihapus";  
    if($hapus){  
        return Response()->json([$a]);  
    } else {  
        return Response()->json([$b]);  
    }  
}
```

DELETE

http://localhost:8000/api/hapus_anggota/16

Send

Save

PrettyRawPreviewVisualizeBETAJSON

123

["Data Berhasil Dihapus"]

2. Buku

- Create

```
public function create(Request $req){
    $validator = Validator::make($req->all(),
        [
            'judul' => 'required',
            'penerbit' => 'required',
            'pengarang' => 'required',
            'foto' => 'required'
        ]
    );

    if($validator->fails()){
        return Response()->json($validator->errors());
    }

    $simpan = ModelBuku::create([
        'judul'=>$req->judul,
        'penerbit'=>$req->penerbit,
        'pengarang'=>$req->pengarang,
        'foto'=>$req->foto,
    ]);

    if($simpan){
        return Response()->json(['status'=>1]);
    } else {
        return Response()->json(['status'=>0]);
    }
}
```

POST http://localhost:8000/api/register_buku 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	...	Bulk Edit
<input checked="" type="checkbox"/> judul	JAJAJA			
<input checked="" type="checkbox"/> penerbit	JAJA			
<input checked="" type="checkbox"/> pengarang	JAJA			
<input checked="" type="checkbox"/> foto	1212			
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 1036ms Size: 290 B Save Response

Pretty Raw Preview Visualize BETA JSON 🔍


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

- Read

```
public function tampil(){
    $tampil = ModelBuku::get();

    if($tampil){
        return Response()->json([$tampil]);
    } else {
        return Response()->json([$tampil->errors()]);
    }
}
```

GET http://localhost:8000/api/tampil_buku Send Save

Pretty Raw Preview Visualize BETA JSON 

```
1  [
2
3  {
4      "id_buku": 1,
5      "judul": "Laskar Pelangi",
6      "penerbit": "Erlangga",
7      "pengarang": "Andrea Hirata",
8      "foto": "10290192",
9      "created_at": null,
10     "updated_at": null
11  },
12  {
13     "id_buku": 2,
14     "judul": "Sang Pemimpi",
15     "penerbit": "Erlangga",
16     "pengarang": "Andrea Hirata",
17     "foto": "10290192",
18     "created_at": null,
19     "updated_at": null
20  },
21  {
22     "id_buku": 3,
23     "judul": "Dilan 1990",
24     "penerbit": "Angkasa",
25     "pengarang": "Pidli Baiq",
26     "foto": "10290192",
27     "created_at": null,
```

- Update

```
public function edit($id, Request $req){
    $validator = Validator::make($req->all(),
        [
            'judul'=>'required',
            'penerbit'=>'required',
            'pengarang'=>'required',
            'foto'=>'required',
        ]
    );

    if($validator->fails()){
        return Response()->json($validator->errors());
    }

    $edit = ModelBuku::where('id_buku',$id)->update([
        'judul'=>$req->judul,
        'penerbit'=>$req->penerbit,
        'pengarang'=>$req->pengarang,
        'foto'=>$req->foto
    ]);

    if($edit){
        return Response()->json(['status'=>1]);
    } else {
        return Response()->json(['status'=>0]);
    }
}
```

POST http://localhost:8000/api/edit_buku/7 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	***	Bulk Edit
<input checked="" type="checkbox"/> judul	JAJAJA			
<input checked="" type="checkbox"/> penerbit	JAJA			
<input checked="" type="checkbox"/> pengarang	JAJAJA			
<input checked="" type="checkbox"/> foto	1212			
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 1133ms Size: 290 B Save Response

Pretty Raw Preview Visualize BETA **JSON** ↺

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

- Delete

```
public function hapus($id){  
    $hapus = ModelBuku::where('id_buku',$id)->delete();  
  
    $a = "Data Berhasil Dihapus";  
    $b = "Data Gagal Dihapus";  
    if($hapus){  
        return Response()->json([$a]);  
    } else {  
        return Response()->json([$b]);  
    }  
}
```

DELETE

http://localhost:8000/api/hapus_buku/7

Send

Save

PrettyRawPreviewVisualizeBETAJSON

123

"Data Berhasil Dihapus"