

Nama : Kandiya Asoka Kristiwi

Absen : 14

1. Controller

a. Anggota

```
Anggotacontroller.php X Anggota.php api.php Buku.php Bukucontroller.php ... Anggotacontroller.php X
app > Http > Controllers > Anggotacontroller.php
10 public function store(Request $req)
11 {
12     $validator=Validator::make($req->all(),
13     [
14         'nama_anggota'=>'required',
15         'alamat'=>'required',
16         'telp'=>'required',
17     ]
18     );
19     if($validator->fails()){
20         return Response()->json($validator->errors());
21     }
22
23     $simpan=Anggota::create([
24         'nama_anggota'=>$req->nama_anggota,
25         'alamat'=>$req->alamat,
26         'telp'=>$req->telp,
27     ]);
28     if($simpan){
29         $status='sukses';
30         $message='Data Anggota Berhasil Ditambahkan';
31     } else {
32         $status='gagal';
33         $message='Data Siswa Anggota Ditambahkan';
34     }
35     return Response()->json(compact('status','message'));
36 }
37 public function update($id, Request $req)
38 {
39     $validator=Validator::make($req->all(),
40     [
41         'nama_anggota'=>'required',
42         'alamat'=>'required',
43         'telp'=>'required',
44     ]
45     );
46     if($validator->fails()){
47         return Response()->json($validator->errors());
48     }
49
50     $ubah=Anggota::where('id',$id)->update([
51         'nama_anggota'=>$req->nama_anggota,
52         'alamat'=>$req->alamat,
53         'telp'=>$req->telp,
54     ]);
55     if($ubah){
56         $status='sukses';
57         $message='Data Anggota Berhasil Diubah';
58     } else {
59         $status='gagal';
60         $message='Data Anggota Gagal Diubah';
61     }
62     return Response()->json(compact('status','message'));
63 }
64 public function destroy($id)
65 {
66     $hapus=Anggota::where('id',$id)->delete();
67     if($hapus){
68         $status='sukses';
69         $message='Data Anggota Berhasil Dihapus';
70     } else {
71         $status='gagal';
72         $message='Data Anggota Gagal Dihapus';
73     }
74     return Response()->json(compact('status','message'));
75 }
76 public function tampil()
77 {
78     $data_anggota=Anggota::get();
79     $count=$data_anggota->count();
80     $arr_data=array();
81     foreach($data_anggota as $dt_ag){
82         $arr_data[]=array(
83             'id'=>$dt_ag->id,
84             'nama_anggota'=>$dt_ag->nama_anggota,
85             'alamat'=>$dt_ag->alamat,
86             'telp'=>$dt_ag->telp
87         );
88     }
89     $status=1;
90     return Response()->json(compact('status','count','arr_data'));
91 }
```

b. Buku

```
Anggotacontroller.php Anggota.php api.php Buku.php Bukucontroller.php X Bukucontroller.php X
app > Http > Controllers > Bukucontroller.php
10 public function store(Request $req)
11 {
12     $validator=Validator::make($req->all(),
13     [
14         'judul'=>'required',
15         'penerbit'=>'required',
16         'pengarang'=>'required',
17         'foto'=>'required',
18     ]
19     );
20     if($validator->fails()){
21         return Response()->json($validator->errors());
22     }
23
24     $simpan=Buku::create([
25         'judul'=>$req->judul,
26         'penerbit'=>$req->penerbit,
27         'pengarang'=>$req->pengarang,
28         'foto'=>$req->foto,
29     ]);
30     if($simpan){
31         $status='sukses';
32         $message='Data Buku Berhasil Ditambahkan';
33     } else {
34         $status='gagal';
35         $message='Data Buku Gagal Ditambahkan';
36     }
37     return Response()->json(compact('status','message'));
38 }
39 public function update($id, Request $req)
40 {
41     $validator=Validator::make($req->all(),
42     [
43         'judul'=>'required',
44         'penerbit'=>'required',
45         'pengarang'=>'required',
46         'foto'=>'required',
47     ]
48     );
49     if($validator->fails()){
50         return Response()->json($validator->errors());
51     }
52
53     $ubah=Buku::where('id',$id)->update([
54         'judul'=>$req->judul,
55         'penerbit'=>$req->penerbit,
56         'pengarang'=>$req->pengarang,
57         'foto'=>$req->foto,
58     ]);
59     if($ubah){
60         $status='sukses';
61         $message='Data Buku Berhasil Diubah';
62     } else {
63         $status='gagal';
64         $message='Data Buku Gagal Diubah';
65     }
66     return Response()->json(compact('status','message'));
67 }
68 public function destroy($id)
69 {
70     $hapus=Buku::where('id',$id)->delete();
71     if($hapus){
72         $status='sukses';
73         $message='Data Buku Berhasil Dihapus';
74     } else {
75         $status='gagal';
76         $message='Data Buku Gagal Dihapus';
77     }
78     return Response()->json(compact('status','message'));
79 }
80 public function tampil()
81 {
82     $data_buku=Buku::get();
83     $count=$data_buku->count();
84     $arr_data=array();
85     foreach($data_buku as $dt_bk){
86         $arr_data[]=array(
87             'id'=>$dt_bk->id,
88             'judul'=>$dt_bk->judul,
89             'penerbit'=>$dt_bk->penerbit,
90             'pengarang'=>$dt_bk->pengarang,
91             'foto'=>$dt_bk->foto
92         );
93     }
94     $status=1;
95     return Response()->json(compact('status','count','arr_data'));
96 }
```

2. Model

a. Anggota

```
app > Anggota.php
1  <?php
2
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";
11     public $timestamps = false;
12
13     protected $fillable = [
14         'nama_anggota', 'alamat', 'telp'
15     ];
16 }
```

b. Buku

```
app > Buku.php
1  <?php
2
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";
11     public $timestamps=false;
12
13     protected $fillable=[
14         'judul', 'penerbit', 'pengarang', 'foto'
15     ];
16 }
```

3. Testing Postman

a. Anggota

The image displays two Postman requests for the 'Anggota' endpoint. The first request is a PUT request to `127.0.0.1:8000/api/simpan_anggota` with a body of `{ "status": "sukses", "message": "Data Anggota Berhasil Ditambahkan" }`. The second request is a DELETE request to `127.0.0.1:8000/api/hapus_anggota/10` with a body of `{ "status": "sukses", "message": "Data Anggota Berhasil Dihapus" }`. Both requests show a 'Key' and 'Value' table for the body data.

KEY	VALUE
nama_anggota	Hendery W
alamat	Jalanmu
telp	0888000888
email	hendery@gmail.com
password	123456
password_confirmation	123456

KEY	VALUE
nama_anggota	Wong Kunhang
alamat	Jalan Macau
telp	232728738174

```
1 {
2   "status": "sukses",
3   "message": "Data Anggota Berhasil Ditambahkan"
4 }
```

```
1 {
2   "status": "sukses",
3   "message": "Data Anggota Berhasil Dihapus"
4 }
```

b. Buku

The image displays two Postman requests for the 'Buku' endpoint. The first request is a POST request to `127.0.0.1:8000/api/simpan_buku` with a body of `{ "status": "sukses", "message": "Data Buku Berhasil Ditambahkan" }`. The second request is a DELETE request to `127.0.0.1:8000/api/hapus_buku/6` with a body of `{ "status": "sukses", "message": "Data Buku Berhasil Dihapus" }`. Both requests show a 'Key' and 'Value' table for the body data.

KEY	VALUE
judul	Resep Rahasia Ayam Goreng Mail
penerbit	Tadika Mesra
pengarang	Ismail bin Mail
foto	-

KEY	VALUE
judul	Resep Ayam Goreng Mail
penerbit	Tadika Mesra
pengarang	Ismail bin Mail
foto	-

```
1 {
2   "status": "sukses",
3   "message": "Data Buku Berhasil Ditambahkan"
4 }
```

```
1 {
2   "status": "sukses",
3   "message": "Data Buku Berhasil Dihapus"
4 }
```