

Transaksi Peminjaman Buku

Heni Yunar Larasati

XI RPL 6 / 16

1. Script Controller
 - a. CRUD Peminjaman
 1. Create (POST)

```
14 public function store(Request $req)
15 {
16     if(Auth::user()->level=="admin"){
17         $validator = Validator::make($req->all(),
18         [
19             'tgl' => 'required',
20             'id_anggota' => 'required',
21             'id_petugas' => 'required',
22             'tgl_deadline' => 'required',
23             'denda' => 'required'
24         ]);
25         if($validator->fails()){
26             return Response()->json($validator->errors());
27         }
28         $simpan = Peminjaman_model::create([
29             'tgl' => $req->tgl,
30             'id_anggota' => $req->id_anggota,
31             'id_petugas' => $req->id_petugas,
32             'tgl_deadline' => $req->tgl_deadline,
33             'denda' => $req->denda,
34
35         ]);
36         return Response()->json(['status'=> 'Data peminjaman berhasil dimasukkan']);
37     } else {
38         return Response()->json(['status'=> 'Data peminjaman gagal dimasukkan, anda bukan admin']);
39     }
40 }
```

2. Read (GET)

```

68:     public function tampil(){
69:         if(Auth::user()->level=="admin"){
70:             $datas = Peminjaman_model::get();
71:             $count = $datas->count();
72:             $anggota = array();
73:             $status = 1;
74:             foreach ($datas as $dt_sw){
75:                 $anggota[] = array(
76:                     'id' => $dt_sw->id,
77:                     'tgl' => $dt_sw->tgl,
78:                     'id_anggota' => $dt_sw->id_anggota,
79:                     'id_petugas' => $dt_sw->id_petugas,
80:                     'tgl_deadline' => $dt_sw->tgl_deadline,
81:                     'denda' => $dt_sw->denda,
82:                     'created_at' => $dt_sw->created_at,
83:                     'updated_at' => $dt_sw->updated_at,
84:                 );
85:             }
86:             return Response()->json(compact('count','anggota'));
87:         } else {
88:             return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
89:         }
90:     }

```

3. Update (PUT)

```

132 public function update($id, Request $req)
133 {
134     if(Auth::user()->level=="admin"){
135         $validator=Validator::make($req->all(),
136         [
137             'tgl' => 'required',
138             'id_anggota' => 'required',
139             'id_petugas' => 'required',
140             'tgl_deadline' => 'required',
141             'denda' => 'required',
142         ]);
143         if($validator->fails()){
144             return Response()->json($validator->errors());
145         }
146         $ubah=Peminjaman_model::where('id',$id)->update([
147             'tgl' => $req->tgl,
148             'id_anggota' => $req->id_anggota,
149             'id_petugas' => $req->id_petugas,
150             'tgl_deadline' => $req->tgl_deadline,
151             'denda' => $req->denda,
152         ]);
153         return Response()->json(['status'=>1,
154                                 'message'=>'Data peminjaman berhasil diubah']);
155     } else {
156         return Response()->json(['status'=>0,
157                                 'message'=>'Data peminjaman gagal diubah, anda bukan admin']);
158     }
159 }

```

4. Delete (DELETE)

```

187 public function delete($id)
188 {
189     if(Auth::user()->level=="admin"){
190         $hapus=Peminjaman_model::where('id',$id)->delete();
191         return Response()->json([
192             'message'=>'Data peminjaman berhasil dihapus']);
193     } else {
194         return Response()->json([
195             'message'=>'Data peminjaman gagal dihapus, anda bukan admin']);
196     }
197 }
198 }

```

b. CRUD Detail_peminjaman

1. Create (POST)

```

44 public function storedetail(Request $req)
45 {
46     if(Auth::user()->level=="admin"){
47         $validator = Validator::make($req->all(),
48         [
49             'id_pinjam' => 'required',
50             'id_buku' => 'required',
51             'qty' => 'required'
52         ]);
53         if($validator->fails()){
54             return Response()->json($validator->errors());
55         }
56         $simpan = Detail_model::create([
57             'id_pinjam' => $req->id_pinjam,
58             'id_buku' => $req->id_buku,
59             'qty' => $req->qty
60         ]);
61         return Response()->json(['status'=> 'Data detail berhasil dimasukkan']);
62     } else {
63         return Response()->json(['status'=> 'Data detail gagal dimasukkan, anda bukan admin']);
64     }
65 }
66 }

```

2. Read (GET)

```

92 public function tampilDetail(){
93     if(Auth::user()->level=="admin"){
94         $peminjaman=DB::table('peminjaman')
95         ->join('anggota','anggota.id','=','peminjaman.id_anggota')
96         ->join('petugas','petugas.id','=','peminjaman.id_petugas')
97         ->select('peminjaman.id','peminjaman.id_anggota','anggota.nama_anggota',
98         'peminjaman.id_petugas','peminjaman.tgl','peminjaman.tgl_deadline',
99         'peminjaman.denda')
100         ->get();
101
102         $data=array();
103         foreach ($peminjaman as $peminjaman){
104             $detail=Detail_model::where('id_pinjam',$peminjaman->id)->get();
105             $arr_detail=array();
106             foreach($detail as $detail){
107                 $arr_detail[]=array(
108                     'id_peminjaman' => $detail->id_pinjam,
109                     'id_buku' => $detail->id_buku,
110                     'qty' => $detail->qty
111                 );
112             }
113
114             $data[]=array(
115                 'id' => $peminjaman->id,
116                 'id_anggota' => $peminjaman->id_anggota,
117                 'nama_anggota' => $peminjaman->nama_anggota,
118                 'id_petugas' => $peminjaman->id_petugas,
119                 'tgl_pinjam' => $peminjaman->tgl,
120                 'tgl_deadline' => $peminjaman->tgl_deadline,
121                 'denda' => $peminjaman->denda,
122                 'detail_pinjam' => $arr_detail,
123             );
124         }
125         return Response()->json(compact('data'));
126     } else {
127         return Response()->json(['status'=>'anda bukan admin']);
128     }
129 }

```

3. Update (PUT)

```

162 public function updatedetail($id, Request $req)
163 {
164     if(Auth::user()->level=="admin"){
165         $validator=Validator::make($req->all(),
166         [
167             'id_pinjam' => 'required',
168             'id_buku' => 'required',
169             'qty' => 'required'
170         ]);
171         if($validator->fails()){
172             return Response()->json($validator->errors());
173         }
174         $ubah=Detail_model::where('id',$id)->update([
175             'id_pinjam' => $req->id_pinjam,
176             'id_buku' => $req->id_buku,
177             'qty' => $req->qty,
178         ]);
179         return Response()->json(['status'=>1,
180                                 'message'=>'Data detail berhasil diubah']);
181     } else {
182         return Response()->json(['status'=>0,
183                                 'message'=>'Data detail gagal diubah, anda bukan admin']);
184     }
185 }

```

4. Delete (DELETE)

```

200 public function deletedetail($id)
201 {
202     if(Auth::user()->level=="admin"){
203         $hapus=Detail_model::where('id',$id)->delete();
204         return Response()->json([
205             'message'=>'Data detail berhasil dihapus']);
206     } else {
207         return Response()->json([
208             'message'=>'Data detail gagal dihapus, anda bukan admin']);
209     }
210 }
211 }

```

2. Script Model

a. Model Peminjaman


```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Peminjaman_model extends Model
8  {
9      protected $table="peminjaman";
10     protected $primaryKey="id";
11     protected $fillable = [
12         'tgl', 'id_anggota', 'id_petugas', 'tgl_deadline', 'denda',
13     ];
14 }

```

b. Model Detail_peminjaman

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Detail_model extends Model
8  {
9      protected $table="detail_peminjaman";
10     protected $primaryKey="id";
11     protected $fillable = [
12         'id_pinjam', 'id_buku', 'qty'
13     ];
14 }

```

3. Hasil Postman

a. Peminjaman

1. Create (POST)

POST localhost:8000/api/peminjaman 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
<input checked="" type="checkbox"/> tgl	2020-02-03	
<input checked="" type="checkbox"/> id_anggota	3	
<input checked="" type="checkbox"/> id_petugas	4	
<input checked="" type="checkbox"/> tgl_deadline	2020-02-10	
<input checked="" type="checkbox"/> denda	-	
Key	Value	Description

Body Cookies Headers Test Results Status: 200 OK Time: 4.88s Size: 324B Save Response

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "status": "Data peminjaman berhasil dimasukkan"
3 }

```

2. Read (GET)

GET localhost:8000/api/peminjaman Send Save

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "count": 4,
3   "anggota": [
4     {
5       "id": 1,
6       "tgl": "2021-01-01",
7       "id_anggota": 3,
8       "id_petugas": 2,
9       "tgl_deadline": "2021-01-08",
10      "denda": "-",
11      "created_at": "2020-01-28T04:12:43.000000Z",
12      "updated_at": "2020-01-28T04:19:46.000000Z"
13    },
14    {
15      "id": 2,
16      "tgl": "2020-01-28",
17      "id_anggota": 2,
18      "id_petugas": 2,
19      "tgl_deadline": "2020-02-04",
20      "denda": "-",
21      "created_at": "2020-01-28T04:13:26.000000Z",
22      "updated_at": "2020-01-28T04:13:26.000000Z"
23    },
24    {
25      "id": 4,

```

3. Update (PUT)

PUT localhost8000/api/peminjaman/1 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
<input checked="" type="checkbox"/> tgl	2020-02-10	
<input checked="" type="checkbox"/> id_anggota	5	
<input checked="" type="checkbox"/> id_petugas	6	
<input checked="" type="checkbox"/> tgl_deadline	2020-02-17	
<input checked="" type="checkbox"/> denda	*	
Key	Value	Description

Body Cookies Headers (0) Test Results Status: 200 OK Time: 2.58s Size: 332 B Save Response

Pretty Raw Preview Visualize BETA JSON + [-]

```

1 {
2   "status": 1,
3   "message": "Data peminjaman berhasil diubah"
4 }

```

4. Delete (DELETE)

DELETE localhost8000/api/peminjaman/4 Send Save

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

Headers (2)

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Content-Type	application/x-www-form-urlencoded	
<input checked="" type="checkbox"/> Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.ejpc3MtO...	
Key	Value	Description

Temporary Headers (0)

Body Cookies Headers (0) Test Results Status: 200 OK Time: 2.37s Size: 322 B Save Response

Pretty Raw Preview Visualize BETA JSON + [-]

```

1 {
2   "message": "Data peminjaman berhasil dihapus"
3 }

```

b. Detail_peminjaman

1. Create (POST)

POST localhost:8000/api/detail 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
<input checked="" type="checkbox"/> id_pinjam	1	
<input checked="" type="checkbox"/> id_buku	5	
<input checked="" type="checkbox"/> qty	1	
Key	Value	Description

Body Cookies Headers (0) Test Results Status: 200 OK Time: 13.50s Size: 320 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": "Data detail berhasil dimasukkan"
3 }
```

2. Read (GET)

GET localhost:8000/api/detail Send Save

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "data": [
3     {
4       "id": 1,
5       "id_anggota": 5,
6       "nama_anggota": "Iarasati",
7       "id_petugas": 6,
8       "tgl_pinjam": "2020-02-18",
9       "tgl_deadline": "2020-02-17",
10      "senda": "-",
11      "detail_pinjam": [
12        {
13          "id_peminjaman": 1,
14          "id_buku": 4,
15          "qty": "3"
16        },
17        {
18          "id_peminjaman": 1,
19          "id_buku": 5,
20          "qty": "1"
21        }
22      ]
23    },
24    {
25      "id": 2,
```

3. Update (PUT)

PUT localhost:8000/api/detail/5 Send Save

Params Authorization Headers (10) Body Pre-request Scripts Tests Settings Cookies Code

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

KEY	VALUE	DESCRIPTION	*** Bulk Edit
<input checked="" type="checkbox"/> id_pinjam	1		
<input checked="" type="checkbox"/> id_buku	2		
<input checked="" type="checkbox"/> qty	3		
Key	Value	Description	

Body Cookies Headers (9) Test Results Status: 200 OK Time: 3.06s Size: 328 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": 1,
3   "message": "Data detail berhasil diubah"
4 }
```

4. Delete (DELETE)

DELETE localhost:8000/api/detail/5 Send Save

Params Authorization Headers (10) Body Pre-request Scripts Tests Settings Cookies Code

▼ Headers (2)

KEY	VALUE	DESCRIPTION	*** Bulk Edit Presets
<input checked="" type="checkbox"/> Content-Type	application/x-www-form-urlencoded		
<input checked="" type="checkbox"/> Authorization	bearer eyJ0eXAvOjklKV1QlLQjhbGciOiJIUzI1Ni9yYjpc3M/O...		
Key	Value	Description	

▼ Temporary Headers (0)

Body Cookies Headers (5) Test Results Status: 200 OK Time: 2.12s Size: 318 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "message": "Data detail berhasil dihapus"
3 }
```