

SALSABILA GEMINTANG KEJORA

XI LARAVEL / 32

- Peminjaman

❖ CREATE

The screenshot shows a Postman interface with a POST request to `localhost/perpus/public/api/simpan`. The request body is a JSON object with the following data:

Key	Value	Description
<input checked="" type="checkbox"/> <code>tgl_pinjam</code>	<code>2020-01-01</code>	
<input checked="" type="checkbox"/> <code>id_anggota</code>	<code>25</code>	
<input checked="" type="checkbox"/> <code>id_petugas</code>	<code>10</code>	
<input checked="" type="checkbox"/> <code>deadline</code>	<code>2020-01-10</code>	
<input checked="" type="checkbox"/> <code>denda</code>	<code>-</code>	

The response status is `200 OK` with a time of `785ms` and size of `408 B`. The response body is a JSON object:

```
1 {
2   "status": 1,
3   "message": "Peminjaman Berhasil Ditambahkan"
4 }
```

❖ READ

The screenshot shows a Postman interface with a GET request to `localhost/perpus/public/api/tampil`. The response status is `200 OK` with a time of `4.44s` and size of `591 B`. The response body is a JSON object:

```
1 {
2   "data": [
3     {
4       "id": 1,
5       "nama_anggota": "anatasya",
6       "alamat": "Halang",
7       "telp": "09765234567",
8       "nama_petugas": "anatasya",
9       "tgl_pinjam": "2020-01-01",
10      "deadline": "2020-01-10",
11      "denda": null
12     }
13   ],
14   "status": 1,
15   "message": "Peminjaman Berhasil ditampilkan",
16   "count": 1
17 }
```

❖ UPDATE

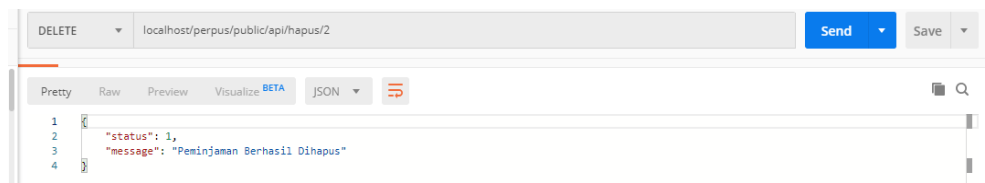
The screenshot shows a Postman interface with a PUT request to `localhost/perpus/public/api/ubah/2`. The request body is a JSON object with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> <code>tgl_pinjam</code>	<code>2020-02-10</code>	
<input checked="" type="checkbox"/> <code>id_anggota</code>	<code>23</code>	
<input checked="" type="checkbox"/> <code>id_petugas</code>	<code>7</code>	
<input checked="" type="checkbox"/> <code>deadline</code>	<code>2020-02-15</code>	
<input checked="" type="checkbox"/> <code>denda</code>	<code>-</code>	

The response status is `200 OK` with a time of `960ms` and size of `403 B`. The response body is a JSON object:

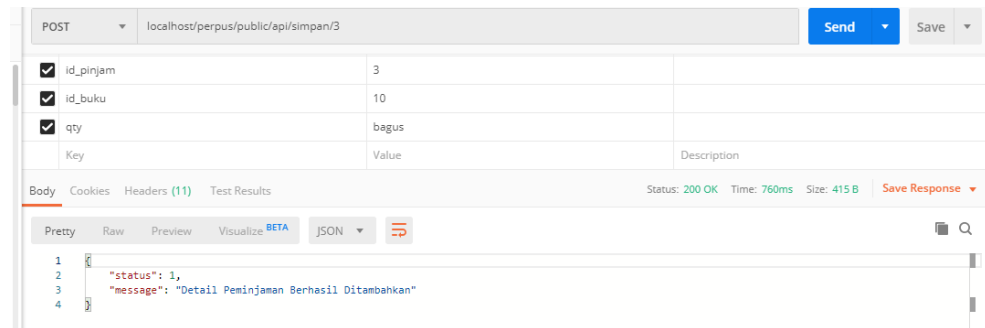
```
1 {
2   "status": 1,
3   "message": "Peminjaman Berhasil Diubah"
4 }
```

❖ DELETE

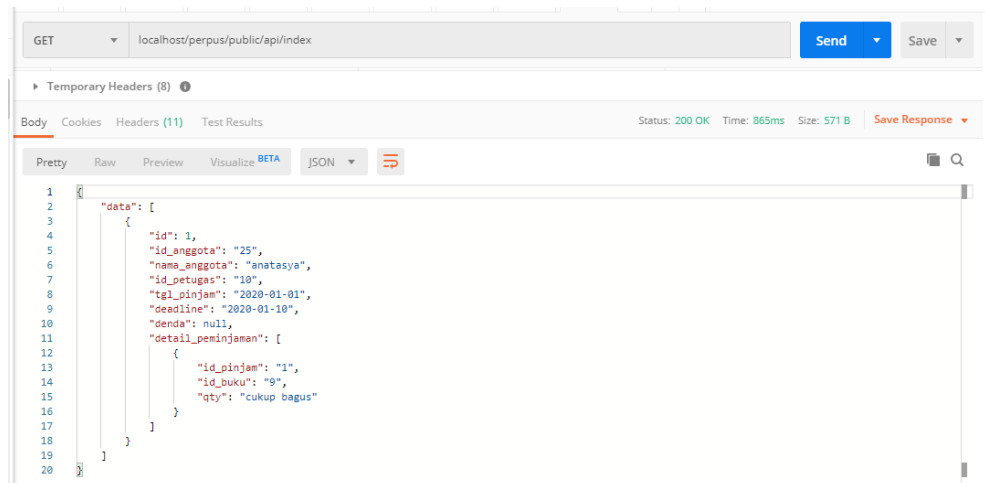


• Detail Peminjaman

❖ CREATE



❖ READ



❖ UPDATE

PUT localhost/perpus/public/api/ubah/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	*** Bulk Edit
<input checked="" type="checkbox"/> id_pinjam	1		
<input checked="" type="checkbox"/> id_buku	9		
<input checked="" type="checkbox"/> qty	cukup bagus		
Key	Value	Description	

Body Cookies Headers (11) Test Results Status: 200 OK Time: 7.62s Size: 410 B Save Response

Pretty Raw Preview Visualize BETA JSON Send

```

1 {
2   "status": 1,
3   "message": "Detail Peminjaman Berhasil Diubah"
4 }

```

❖ HAPUS

DELETE localhost/perpus/public/api/hapus/2 Send Save

Pretty Raw Preview Visualize BETA JSON Send

```

1 {
2   "status": 1,
3   "message": "Detail Peminjaman Berhasil Dihapus"
4 }

```

• SCRIPT

❖ Contoller

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Auth;
7 use App\Peminjaman_model;
8 use App\Buku_model;
9 use App\detail_pinjam;
10 use JWTAuth;
11 use DB;
12 use Illuminate\Support\Facades\Validator;
13 use Tymon\JWTAuth\Exceptions\JWTException;
14
15 class peminjamancontroller extends Controller
16 {
17     public function index(){
18         if(Auth::user()->level=="admin"){
19             $peminjaman=DB::table('peminjaman')
20                 ->join('anggota','anggota.id','=','peminjaman.id_anggota')
21                 ->join('petugas','petugas.id','=','peminjaman.id_petugas')
22                 ->select('peminjaman.id','peminjaman.id_anggota','anggota.nama_anggota','peminjaman.id_petugas','peminjaman.tgl_pinjam','peminjaman.deadline','peminjaman.denda')
23                 ->get();
24
25             $data=array();
26             foreach ($peminjaman as $dt_pj){
27                 $sok=detail_pinjam::where('id_pinjam',$dt_pj->id)->get();
28                 $sarr_detail=array();
29                 foreach ($sok as $sk) {
30                     $sarr_detail[]=array(
31                         'id_pinjam' => $sk->id_pinjam,
32                         'id_buku' => $sk->id_buku,
33                         'qty' => $sk->qty
34                     );
35                 }
36             }
37         }

```

```

.env      User.php      Anggota_model.php      Buku_model.php      petugascontroller.php      bukucontroller.php      peminjamancontroller.php x      Peminjaman_model.php      api.php
38      $data=array(
39          'id' => $dt_pj->id,
40          'id_anggota' => $dt_pj->id_anggota,
41          'nama_anggota' => $dt_pj->nama_anggota,
42          'id_petugas' => $dt_pj->id_petugas,
43          'tgl_pinjam' => $dt_pj->tgl_pinjam,
44          'deadline' => $dt_pj->deadline,
45          'denda' => $dt_pj->denda,
46          'detail_peminjaman' => $arr_detail
47      );
48      }
49      return response()->json(compact('data'));
50  } else{
51      return response()->json(['status'=>'anda bukan admin']);
52  }
53  }
54
55  public function store(Request $request){
56      $validator=Validator::make($request->all(),
57      [
58          'id_anggota'=>'required',
59          'id_petugas'=>'required',
60          'tgl_pinjam'=>'required',
61          'deadline'=>'required',
62          'denda'=>'required'
63      ]
64      );
65
66      if($validator->fails()){
67          return Response()->json($validator->errors());
68      }
69

```

```

.env      User.php      Anggota_model.php      Buku_model.php      petugascontroller.php      bukucontroller.php      peminjamancontroller.php x      Peminjaman_model.php      api.php
69
70      $simpan=Peminjaman_model::create([
71          'id_anggota'=>$request->id_anggota,
72          'id_petugas'=>$request->id_petugas,
73          'tgl_pinjam'=>$request->tgl_pinjam,
74          'deadline'=>$request->deadline,
75          'denda'=>$request->denda
76      ]);
77      $status=1;
78      $message="Peminjaman Berhasil Ditambahkan";
79      if($simpan){
80          return Response()->json(compact('status','message'));
81      }else {
82          return Response()->json(['status'=>0]);
83      }
84  }
85
86  public function update($id,Request $request){
87      $validator=Validator::make($request->all(),
88      [
89          'id_anggota'=>'required',
90          'id_petugas'=>'required',
91          'tgl_pinjam'=>'required',
92          'deadline'=>'required',
93          'denda'=>'required'
94      ]
95      );
96
97      if($validator->fails()){
98          return Response()->json($validator->errors());
99      }
100

```

```

.env      User.php      Anggota_model.php      Buku_model.php      petugascontroller.php      bukucontroller.php      peminjamancontroller.php x      Peminjaman_model.php      api.php
100
101      $ubah=Peminjaman_model::where('id',$id)->update([
102          'id_anggota'=>$request->id_anggota,
103          'id_petugas'=>$request->id_petugas,
104          'tgl_pinjam'=>$request->tgl_pinjam,
105          'deadline'=>$request->deadline,
106          'denda'=>$request->denda
107      ]);
108      $status=1;
109      $message="Peminjaman Berhasil Diubah";
110      if($ubah){
111          return Response()->json(compact('status','message'));
112      }else {
113          return Response()->json(['status'=>0]);
114      }
115  }
116
117  public function tampil_pinjam(){
118      $data=DB::table('peminjaman')
119      ->join('anggota','anggota.id','=','peminjaman.id_anggota')
120      ->join('petugas','petugas.id','=','peminjaman.id_petugas')
121      ->select('peminjaman.id','anggota.nama_anggota','anggota.alamat','anggota.telp','petugas.nama_petugas','peminjaman.tgl_pinjam','peminjaman.deadline','peminjaman.denda')
122      ->get();
123      $count=$data->count();
124      $status=1;
125      $message="Peminjaman Berhasil ditampilkan";
126      return response()->json(compact('data','status','message','count'));
127  }
128
129  public function destroy($id){
130      $hapus=Peminjaman_model::where('id',$id)->delete();
131      $status=1;
132      $message="Peminjaman Berhasil Dihapus";
133      if($hapus){
134          return Response()->json(compact('status','message'));
135      }else {
136          return Response()->json(['status'=>0]);
137      }
138  }

```

```

140 public function simpan(Request $request){
141     $validator=Validator::make($request->all(),
142     [
143         'id_pinjam'=>'required',
144         'id_buku'=>'required',
145         'qty'=>'required'
146     ]
147 );
148
149 if($validator->fails()){
150     return Response()->json($validator->errors());
151 }
152
153 $simpan=detail_pinjam::create([
154     'id_pinjam'=>$request->id_pinjam,
155     'id_buku'=>$request->id_buku,
156     'qty'=>$request->qty
157 ]);
158 $status=1;
159 $message="Detail Peminjaman Berhasil Ditambahkan";
160 if($simpan){
161     return Response()->json(compact('status','message'));
162 }else {
163     return Response()->json(['status'=>0]);
164 }
165 }
166
167
168 public function ubah($id,Request $request){
169     $validator=Validator::make($request->all(),
170     [
171         'id_pinjam'=>'required',
172         'id_buku'=>'required',
173         'qty' =>'required'
174     ]
175 );
176

```

```

176
177 if($validator->fails()){
178     return Response()->json($validator->errors());
179 }
180
181 $ubah=detail_pinjam::where('id',$id)->update([
182     'id_pinjam'=>$request->id_pinjam,
183     'id_buku'=>$request->id_buku,
184     'qty'=>$request->qty
185 ]);
186 $status=1;
187 $message="Detail Peminjaman Berhasil Diubah";
188 if($ubah){
189     return Response()->json(compact('status','message'));
190 }else {
191     return Response()->json(['status'=>0]);
192 }
193
194
195
196 public function hapus($id){
197     $hapus=detail_pinjam::where('id',$id)->delete();
198     $status=1;
199     $message="Detail Peminjaman Berhasil Dihapus";
200     if($hapus){
201         return Response()->json(compact('status','message'));
202     }else {
203         return Response()->json(['status'=>0]);
204     }
205 }
206
207

```

❖ Model

- Peminjaman model

```

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
12    protected $fillable = [
13        'id', 'tgl_pinjam', 'id_anggota', 'id_petugas', 'deadline',
14    ];
15
16    public function Anggota_model(){
17        return $this->belongsTo('App\Anggota_model', 'id_anggota');
18    }
19
20    public function Petugas_model(){
21        return $this->belongsTo('App\Petugas_model', 'id_petugas');
22    }
23
24    public function detail(){
25        return $this->hasOne('App\detail_peminjaman', 'id');
26    }
27
28    public $timestamps=false;
29
30 }
31

```

○ Detail pinjam model

```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class detail_pinjam extends Model
8 {
9     protected $table="detail_peminjaman";
10    protected $primaryKey="id";
11    protected $fillable = [
12        'id', 'id_pinjam', 'id_buku', 'qty'
13    ];
14
15    public function buku() {
16        return $this->belongsTo('App/Buku_model', 'id_buku');
17    }
18    public function peminjaman() {
19        return $this->belongsTo('App/Peminjaman_model', 'id_pinjam');
20    }
21 }
22
```