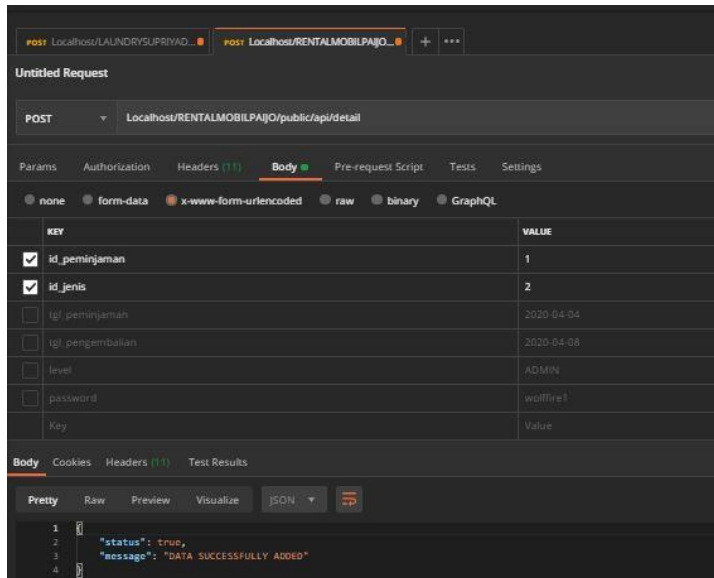


Nama : Ananda Rafi Amanullah

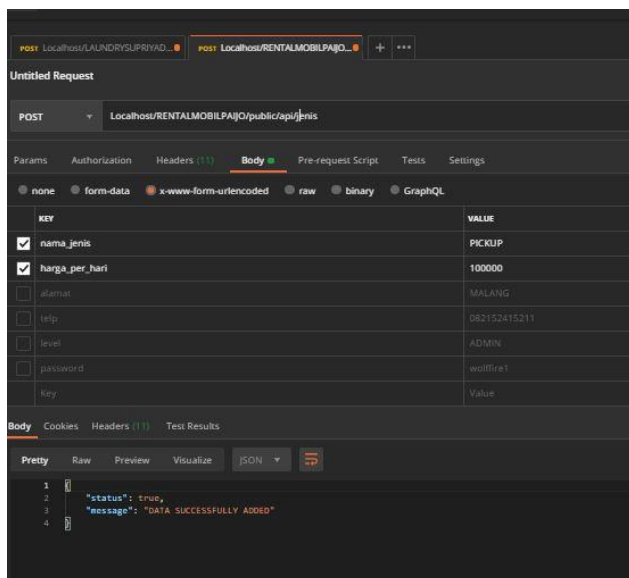
Kelas : XI PRL 7 / 06

## SCREENSHOOT

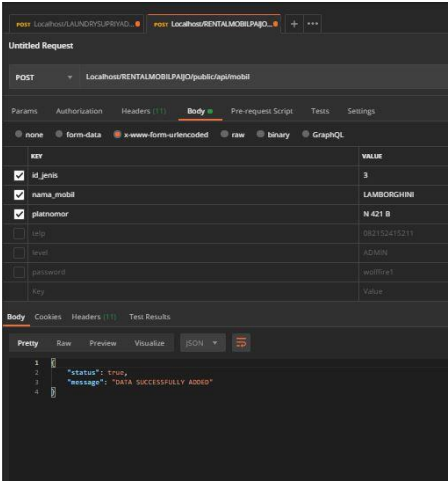
### Tambah Detail



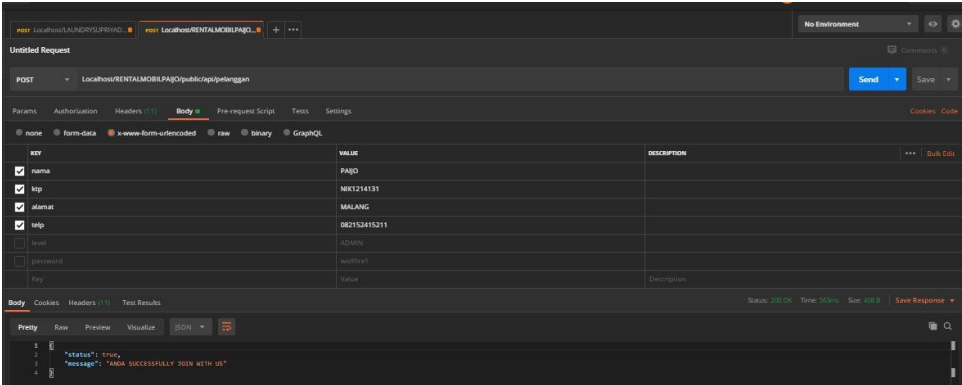
### Tambah Jenis



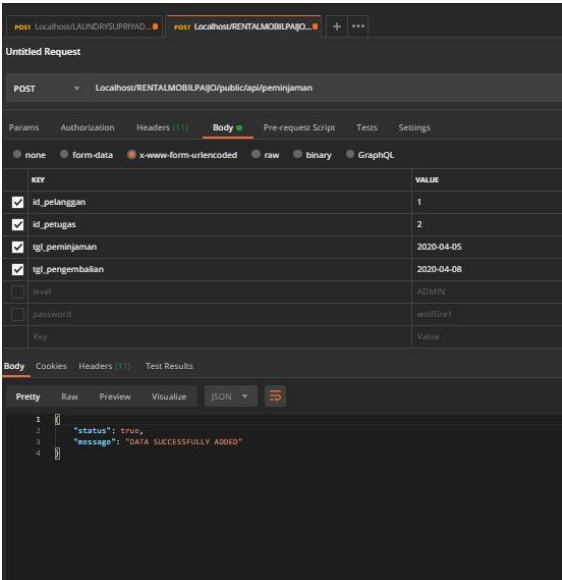
TAMBAH MOBIL



TAMBAH PELANGGAN



TAMBAH PEMINJAMAN



## UBAH DETAIL

POST Localhost/LAUNDRI/SUPRIYAD... PUT Localhost/RENTALMOBILPAJO/...

Untitled Request

PUT Localhost/RENTALMOBILPAJO/public/api/detail/1

Params Authorization Headers (1) Body Pre-request Script Tests Settings

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

KEY	VALUE
<input checked="" type="checkbox"/> id_peminjaman	1
<input checked="" type="checkbox"/> id_jenis	3
<input type="checkbox"/> tgl_peminjaman	2020-04-04
<input type="checkbox"/> tgl_pengembalian	2020-04-08
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
Key	Value

Body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY CHANGED"
4 }
```

## UBAH JENIS

POST Localhost/LAUNDRI/SUPRIYAD... PUT Localhost/RENTALMOBILPAJO/...

Untitled Request

PUT Localhost/RENTALMOBILPAJO/public/api/jenis/1

Params Authorization Headers (1) Body Pre-request Script Tests Settings

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

KEY	VALUE
<input checked="" type="checkbox"/> nama_jenis	PICKUP
<input checked="" type="checkbox"/> harga_per_hari	90000
<input type="checkbox"/> status	MALANG
<input type="checkbox"/> trip	082152415211
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
Key	Value

Body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY CHANGED"
4 }
```

## UBAH MOBIL

The screenshot shows a Postman interface with a PUT request to `localhost/RENTALMOBILPAJO/public/api/mobil/7`. The request body is in JSON format, containing the following data:

KEY	VALUE
<input checked="" type="checkbox"/> id_jenis	3
<input checked="" type="checkbox"/> nama_mobil	LAMBORGHINI AVENTADOR
<input checked="" type="checkbox"/> platnomor	N 421 B
<input type="checkbox"/> telp	082152415211
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
Key	Value

The response body is in JSON format:

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY CHANGED"
4 }
```

## UBAH PELANGGAN

The screenshot shows a Postman interface with a PUT request to `localhost/RENTALMOBILPAJO/public/api/pelanggan/3`. The request body is in JSON format, containing the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama	SUPRI	
<input checked="" type="checkbox"/> tlp	NR1214131	
<input checked="" type="checkbox"/> alamat	MALANG	
<input checked="" type="checkbox"/> telp	082152415211	
<input type="checkbox"/> level	ADMIN	
<input type="checkbox"/> password	wolffire1	
Key	Value	Description

The response body is in JSON format:

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY CHANGED"
4 }
```

## UBAH PEMINJAMAN

The screenshot shows a Postman interface with a PUT request to `localhost/RENTALMOBILPAJO/public/api/peminjaman/2`. The request body is in JSON format, containing the following data:

KEY	VALUE
<input checked="" type="checkbox"/> id_pelanggan	1
<input checked="" type="checkbox"/> id_petugas	2
<input checked="" type="checkbox"/> tgl_peminjaman	2020-04-04
<input checked="" type="checkbox"/> tgl_pengembalian	2020-04-08
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
Key	Value

The response body is in JSON format:

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY CHANGED"
4 }
```

DELETE UBAH

POST Localhost/LAUNDARYSUPRIYAD...DELETE Localhost/RENTALMOBILPAJO/...

Untitled Request

DELETE Localhost/RENTALMOBILPAJO/public/api/jenis/6

ParamsAuthorizationHeaders (11)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE
<input checked="" type="checkbox"/> nama_jenis	PICKUP
<input checked="" type="checkbox"/> harga_per_hari	90000
<input type="checkbox"/> alamat	MALANG
<input type="checkbox"/> telepon	082152415211
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	walffire1
<input type="checkbox"/> Key	Value

BodyCookiesHeaders (11)Test Results

PrettyRawPreviewVisualizeJSON

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY DELETED"
4 }
```

DETELE DETAIL

POST Localhost/LAUNDARYSUPRIYAD...DELETE Localhost/RENTALMOBILPAJO/...

Untitled Request

DELETE Localhost/RENTALMOBILPAJO/public/api/detail/1

ParamsAuthorizationHeaders (11)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE
<input checked="" type="checkbox"/> id_peminjaman	1
<input checked="" type="checkbox"/> id_jenis	3
<input type="checkbox"/> tgl_peminjaman	2020-04-01
<input type="checkbox"/> tgl_pengembalian	2020-04-01
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	walffire1
<input type="checkbox"/> Key	Value

BodyCookiesHeaders (11)Test Results

PrettyRawPreviewVisualizeJSON

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY DELETED"
4 }
```

DELETE MOBIL

POST Localhost/LAUNDARYSUPRIYAD...DELETE Localhost/RENTALMOBILPAJO/...

Untitled Request

DELETE Localhost/RENTALMOBILPAJO/public/api/mobil/7

ParamsAuthorizationHeaders (11)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE
<input checked="" type="checkbox"/> id_jenis	3
<input checked="" type="checkbox"/> nama_mobil	LAMBORGHINI AVENTADOR
<input checked="" type="checkbox"/> platnomor	N 421 B
<input type="checkbox"/> telepon	082152415211
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	walffire1
<input type="checkbox"/> Key	Value

BodyCookiesHeaders (11)Test Results

PrettyRawPreviewVisualizeJSON

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY DELETED"
4 }
```

## DELETE PELANGGAN

The screenshot shows a REST client interface with a DELETE request to `Localhost/RENTALMOBILPAJO/public/api/pelanggan/3`. The request body is a JSON object with the following fields:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nama	SUPRI	
<input checked="" type="checkbox"/> hp	NR1214131	
<input checked="" type="checkbox"/> alamat	MALANG	
<input checked="" type="checkbox"/> telp	082152415211	
<input type="checkbox"/> level	ADMIN	
<input type="checkbox"/> password	wolffire1	
Key	Value	Description

The response body is a JSON object:

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY DELETED"
4 }
```

## DELETE PEMINJAMAN

The screenshot shows a REST client interface with a DELETE request to `Localhost/RENTALMOBILPAJO/public/api/peminjaman/2`. The request body is a JSON object with the following fields:

KEY	VALUE
<input checked="" type="checkbox"/> id_pelanggan	1
<input checked="" type="checkbox"/> id_petugas	2
<input checked="" type="checkbox"/> tgl_peminjaman	2020-04-04
<input checked="" type="checkbox"/> tgl_pengembalian	2020-04-08
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
Key	Value

The response body is a JSON object:

```
1 {
2   "status": true,
3   "message": "DATA SUCCESSFULLY DELETED"
4 }
```

## GET DETAIL

POST Localhost/LAUNDRYSUPRIYAD... GET Localhost/RENTALMOBILPAJO...

### Untitled Request

GET Localhost/RENTALMOBILPAJO/public/api/detail

Params Authorization Headers (1) **Body** Pre-request Script Tests Settings

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

KEY	VALUE
<input checked="" type="checkbox"/> id_peminjaman	1
<input checked="" type="checkbox"/> id_jenis	2
<input type="checkbox"/> tgl_peminjaman	2020-04-04
<input type="checkbox"/> tgl_pengembalian	2020-04-08
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
Key	Value

body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id_detail": 1,
4     "id_jenis": 2,
5     "nama_jenis": "MINIBUS",
6     "nama_mobil": "SUUZKI",
7     "id_peminjaman": 1,
8     "harga_per_hari": "120000",
9     "peminjaman_hari": "2",
10    "total": "240000"
11  }
12 }
13
14
```

## DETAIL JENIS

POST Localhost/LAUNDRYSUPRIYAD... GET Localhost/RENTALMOBILPAJO...

### Untitled Request

GET Localhost/RENTALMOBILPAJO/public/api/jenis

Params Authorization Headers (1) **Body** Pre-request Script Tests Settings

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

KEY	VALUE
<input checked="" type="checkbox"/> nama_jenis	PICKUP
<input checked="" type="checkbox"/> harga_per_hari	90000
<input type="checkbox"/> alamat	MALANG
<input type="checkbox"/> telp	082152415211
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
Key	Value

body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "Jumlah DATA": 3,
3   "DATA": [
4     {
5       "id_jenis": 1,
6       "nama_jenis": "SEDAN",
7       "harga_per_hari": "100000"
8     },
9     {
10      "id_jenis": 2,
11      "nama_jenis": "MINIBUS",
12      "harga_per_hari": "250000"
13    },
14    {
15      "id_jenis": 3,
16      "nama_jenis": "SPORT",
17      "harga_per_hari": "400000"
18    }
19  ],
20   "status": true
21 }
```

# DETAIL MOBIL

POST Localhost/LAUNCHUP/RYAD...GET Localhost/RENTALMOBILPAJID...+...

Untitled Request

GETLocalhost/RENTALMOBILPAJID/public/api/mobil

Params

Authorization

Headers (11)

Body

Pre-request Scripts

Tests

Settings

none

form-data

selectedx-www-form-urlencoded

raw

binary

GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> id_jenis	3
<input checked="" type="checkbox"/> nama_mobil	LAMBORGHINI AVENTADOR
<input checked="" type="checkbox"/> platnomor	N 421 B
<input type="checkbox"/> telp	082152415211
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
key	Value

Body

Cookies

Headers (11)

Test Results

Pretty

Raw

Preview

Visualize

JSON

22

```
1 {
2   "Jumlah Data": 2,
3   "Data": [
4     {
5       "id_mobil": 1,
6       "id_jenis": 3,
7       "nama_mobil": "TOYOTA AVANZA",
8       "platnomor": "B 2334 LD",
9       "created_at": null,
10      "updated_at": null
11    },
12    {
13      "id_mobil": 4,
14      "id_jenis": 3,
15      "nama_mobil": "MITSUBISHI MINIRUB",
16      "platnomor": "B 2333 ASB",
17      "created_at": null,
18      "updated_at": null
19    }
20  ],
21   "status": true
22 }
```

# DETAIL PELANGGAN

POST Localhost/LAUNCHUP/RYAD...GET Localhost/RENTALMOBILPAJID...+...

Untitled Request

GETLocalhost/RENTALMOBILPAJID/public/api/pelanggan

Params

Authorization

Headers (11)

Body

Pre-request Scripts

Tests

Settings

none

form-data

selectedx-www-form-urlencoded

raw

binary

GraphQL

<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	wolffire1
key	Value

Body

Cookies

Headers (11)

Test Results

Pretty

Raw

Preview

Visualize

JSON

22

```
1 {
2   "Jumlah Data": 3,
3   "Data": [
4     {
5       "id_pelanggan": 1,
6       "nama": "PALE DIANDRA Maulana",
7       "telp": "081231512123",
8       "alamat": "MALANG",
9       "telp": "082152415211",
10      "created_at": null,
11      "updated_at": null
12    },
13    {
14      "id_pelanggan": 2,
15      "nama": "PALE BEN SURYADI",
16      "telp": "081231512123",
17      "alamat": "KEMAWAN",
18      "telp": "082152415211",
19      "created_at": null,
20      "updated_at": null
21    },
22    {
23      "id_pelanggan": 3,
24      "nama": "PALE BEN SURYADI",
25      "telp": "081231512123",
26      "alamat": "MALANG",
27      "telp": "082152415211",
28      "created_at": null,
29      "updated_at": null
30    }
31  ],
32   "status": true
33 }
```



## DETAIL PEMINJAMAN

POST Localhost/LAUNDRI/SPRING... GET Localhost/RENTALMOBILPAJID... + ...

Unlinked Request

GET Localhost/RENTALMOBILPAJID/public/api/peminjaman

Params Authorization Headers (1) Body Pre-request Script Tests Settings

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

KEY	VALUE
<input checked="" type="checkbox"/> id_pelanggan	1
<input checked="" type="checkbox"/> id_peserta	2
<input checked="" type="checkbox"/> tgl_peminjaman	2020-04-05
<input checked="" type="checkbox"/> tgl_pengembalian	2020-04-08
<input type="checkbox"/> level	ADMIN
<input type="checkbox"/> password	admin123
Any	Value

body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id_peminjaman": 1,
4     "id_pelanggan": 1,
5     "id_peserta": "Ananda Rafi Anonillah",
6     "id_peminjaman": 1,
7     "nama": "Rafli Rendra Maulana",
8     "tgl_peminjaman": "2020-04-02",
9     "tgl_pengembalian": "2020-04-05"
10  },
11  {
12    "id_peminjaman": 2,
13    "id_pelanggan": 2,
14    "id_peserta": "Rafli Rendra Maulana",
15    "id_peminjaman": 1,
16    "nama": "Rafli Rendra Maulana",
17    "tgl_peminjaman": "2020-04-05",
18    "tgl_pengembalian": "2020-04-08"
19  }
20 }
21
22 }
```

## API.PHP

```
routes > spiphp
12 | is assigned the "api" middleware group. Enjoy building your API!
13 |
14 | */
15 |
16 | Route::middleware('auth:api')->get('/user', function (Request $request) {
17 |     return $request->user();
18 | });
19 |
20 | /*
21 | |-----|
22 | |             LOGIN PETUGAS RENTAL             |
23 | |-----|
24 | */
25 | Route::post('login', 'UserController@login');
26 | Route::get('/login/check', 'UserController@getAuthenticatedUser')->middleware('jwt.verify');
27 |
28 | /*
29 | |-----|
30 | |             CRUD PETUGAS RENTAL             |
31 | |-----|
32 | */
33 | Route::post('petugas', 'UserController@store');
34 | Route::put('petugas/{id}', 'UserController@update');
35 | Route::delete('petugas/{id}', 'UserController@delete');
36 | Route::get('petugas', 'UserController@tampil');
37 |
38 | /*
39 | |-----|
40 | |             CRUD JENIS MOBIL             |
41 | |-----|
42 | */
43 | Route::post('jenis', 'JenisController@store')->middleware('jwt.verify');
44 | Route::put('jenis/{id}', 'JenisController@update')->middleware('jwt.verify');
45 | Route::delete('jenis/{id}', 'JenisController@delete')->middleware('jwt.verify');
46 | Route::get('jenis', 'JenisController@tampil')->middleware('jwt.verify');
47 |
48 | /*
49 | |-----|
50 | |             CRUD PELANGGAN             |
51 | |-----|
52 | */
53 | Route::post('pelanggan', 'PelangganController@store')->middleware('jwt.verify');
54 | Route::put('pelanggan/{id}', 'PelangganController@update')->middleware('jwt.verify');
55 | Route::delete('pelanggan/{id}', 'PelangganController@delete')->middleware('jwt.verify');
56 | Route::get('pelanggan', 'PelangganController@tampil')->middleware('jwt.verify');
57 |
58 | /*
59 | |-----|
60 | |             CRUD PEMINJAMAN             |
61 | |-----|
62 | */
63 | Route::post('peminjaman', 'PeminjamanController@store')->middleware('jwt.verify');
64 | Route::put('peminjaman/{id}', 'PeminjamanController@update')->middleware('jwt.verify');
65 | Route::delete('peminjaman/{id}', 'PeminjamanController@delete')->middleware('jwt.verify');
66 | Route::get('peminjaman', 'PeminjamanController@tampil')->middleware('jwt.verify');
67 |
68 | /*
69 | |-----|
70 | |             CRUD DETAIL TRANSAKSI             |
71 | |-----|
72 | */
73 | Route::post('detail', 'PeminjamanController@store');
74 | Route::put('detail/{id}', 'PeminjamanController@update');
75 | Route::delete('detail/{id}', 'PeminjamanController@delete');
76 | Route::get('detail', 'PeminjamanController@tampil')->middleware('jwt.verify');
77 |
78 | /*
79 | |-----|
80 | |             CRUD MOBIL             |
81 | |-----|
82 | */
83 | Route::post('mobil', 'MobilController@store')->middleware('jwt.verify');
84 | Route::put('mobil/{id}', 'MobilController@update')->middleware('jwt.verify');
85 | Route::delete('mobil/{id}', 'MobilController@delete')->middleware('jwt.verify');
86 | Route::get('mobil', 'MobilController@tampil')->middleware('jwt.verify');
```

## JENIS CONTROLLER

```
Detail.php JenisController.php X
app > Http > Controllers > JenisController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Support\Facades\Validator;
6 use Illuminate\Http\Request;
7 use App\Jenis;
8 use App\User;
9 use Auth;
10 use DB;
11
12 class JenisController extends Controller
13 {
14     //CEK LEVEL / POSISI PETUGAS
15     public function cek(){
16         if(Auth::user()->level=='admin'){
17             $petugas=DB::table('petugas')->get();
18             return response()->json($petugas);
19         }else{
20             return response()->json(['== ANDA BUKAN ADMIN ==']);
21         }
22     }
23
24     public function store(Request $req)
25     {
26         if(Auth::user()->level=='admin'){
27             $validator = Validator::make($req->all(),
28             [
29                 'nama_jenis' => 'required',
30                 'harga_per_hari' => 'required',
31             ]);
32
33             if($validator->fails()){
34                 return Response()->json($validator->errors());
35             }
36
37             $jenis = Jenis::create([
38                 'nama_jenis' => $req->nama_jenis,
39                 'harga_per_hari' => $req->harga_per_hari,
40             ]);
41             if($jenis){
42                 return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY ADDED']);
43             }
44             else{
45                 return Response()->json(['status'=>false,'message'=>'DATA FAILED TO ADDED']);
46             }
47         }
48     }
49
50     public function update($id, Request $req)
51     {
52         if(Auth::user()->level=='admin'){
53             $validator=Validator::make($req->all(),
54             [
55                 'nama_jenis' => 'required',
56                 'harga_per_hari' => 'required',
57             ]);
58             if($validator->fails()){
59                 return Response()->json($validator->errors());
60             }
61             $jenis=Jenis::where('id_jenis',$id)->update([
62                 'nama_jenis' => $req->nama_jenis,
63                 'harga_per_hari' => $req->harga_per_hari,
64             ]);
65             if($jenis){
66                 return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY CHANGED']);
67             }
68             else{
69                 return Response()->json(['status'=>false,'message'=>'DATA FAILED TO CHANGE']);
70             }
71         }
72     }
73 }
```

```

74 public function delete($id)
75 {
76     if(Auth::user()->level=='admin'){
77         $jenis=jenis::where('id_jenis',$id)->delete();
78         if($jenis){
79             return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY CHANGED']);
80         }
81         else{
82             return Response()->json(['status'=>false,'message'=>'DATA FAILED TO CHANGE']);
83         }
84     }
85 }
86 public function tampil()
87 {
88     if(Auth::user()->level=='admin'){
89         $jenis =jenis::all();
90         $jenis = jenis::count();
91         if($jenis){
92             return Response()->json(['Jumlah DATA'=>$jenis,'DATA'=>$jenis,'status'=>true]);
93         }
94         else{
95             return Response()->json(['status'=>false]);
96         }
97     }
98 }
99 }
100

```

## JENIS MODEL

```

app > jenis.php
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Jenis extends Model
8 {
9     protected $table = "jenis_mobil";
10    protected $primaryKey="id_jenis";
11    protected $fillable =[
12        'id_jenis',
13        'nama_jenis',
14        'harga_per_hari',
15    ];
16    public $timestamps = false;
17 }
18

```

## MOBIL CONTROLLER

```
Detail.php MobilController.php X
app > Http > Controllers > MobilController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Support\Facades\Validator;
6  use Illuminate\Http\Request;
7  use App\Mobil;
8  use App\User;
9  use Auth;
10 use DB;
11
12 class MobilController extends Controller
13 {
14     //CEK LEVEL / POSISI PETUGAS
15     public function cek(){
16         if(Auth::user()->level=='admin'){
17             $petugas=DB::table('petugas')->get();
18             return response()->json($petugas);
19         }else{
20             return response()->json(['== ANDA BUKAN ADMIN ==']);
21         }
22     }
23
24     public function store(Request $req)
25     {
26         if(Auth::user()->level=='admin'){
27             $validator = Validator::make($req->all(),
28             [
29                 'id_jenis' => 'required',
30                 'nama_mobil' => 'required',
31                 'platnomor' => 'required',
32             ]);
33
34             if($validator->fails()){
35                 return Response()->json($validator->errors());
36             }
37
38             $mobil = mobil::create([
39                 'id_jenis' => $req->id_jenis,
40                 'nama_mobil' => $req->nama_mobil,
41                 'platnomor' => $req->platnomor,
42             ]);
43             if($mobil){
44                 return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY ADDED']);
45             }
46             else{
47                 return Response()->json(['status'=>false,'message'=>'DATA FAILED TO ADDED']);
48             }
49         }
50     }
51
52     public function update($id, Request $req)
53     {
54         if(Auth::user()->level=='admin'){
55             $validator=Validator::make($req->all(),
56             [
57                 'id_jenis' => 'required',
58                 'nama_mobil' => 'required',
59                 'platnomor' => 'required',
60             ]);
61             if($validator->fails()){
62                 return Response()->json($validator->errors());
63             }
64             $mobil=mobil::where('id_mobil',$id)->update([
65                 'id_jenis' => $req->id_jenis,
66                 'nama_mobil' => $req->nama_mobil,
67                 'platnomor' => $req->platnomor,
68             ]);
69             if($mobil){
70                 return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY CHANGED']);
71             }
72             else{
73                 return Response()->json(['status'=>false,'message'=>'DATA FAILED TO CHANGE']);
74             }
75         }
76     }
77 }
```

```

77
78 public function delete($id)
79 {
80     if(Auth::user()->level=='admin'){
81         $mobil=mobil::where('id_mobil',$id)->delete();
82         if($mobil){
83             return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY DELETED']);
84         }
85         else{
86             return Response()->json(['status'=>false,'message'=>'DATA FAILED TO DELETED']);
87         }
88     }
89 }
90 public function tampil()
91 {
92     if(Auth::user()->level=='admin'){
93         $mobil =mobil::all();
94         $mobil = mobil::count();
95         if($mobil){
96             return Response()->json(['Jumlah DATA'=>$mobil,'DATA'=>$mobil,'status'=>true]);
97         }
98         else{
99             return Response()->json(['status'=>false]);
100         }
101     }
102 }
103 }
104

```

## MOBIL MODEL

```

app > Mobil.php
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Mobil extends Model
8 {
9     protected $table = "mobil";
10     protected $primaryKey="id_mobil";
11     protected $fillable =[
12         'id_mobil',
13         'id_jenis',
14         'nama_mobil',
15         'platnomer',
16     ];
17     public $timestamps = false;
18 }
19

```

## PELANGGAN CONTROLLER

```
Detail.php  MobilController.php  PelangganController.php X
app > Http > Controllers > PelangganController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Support\Facades\Validator;
6  use Illuminate\Http\Request;
7  use App\Pelanggan;
8  use App\User;
9  use Auth;
10 use DB;
11
12 class PelangganController extends Controller
13 {
14     //CEK LEVEL / POSISI PETUGAS
15     public function cek(){
16         if(Auth::user()->level=='admin'){
17             $petugas=DB::table('petugas')->get();
18             return response()->json($petugas);
19         }else{
20             return response()->json(['=== ANDA BUKAH ADMIN ===']);
21         }
22     }
23
24     public function store(Request $req)
25     {
26         if(Auth::user()->level=='admin'){
27             $validator = Validator::make($req->all(),
28             [
29                 'nama' => 'required',
30                 'ktp' => 'required',
31                 'alamat' => 'required',
32                 'telp' => 'required',
33             ]);
34
35             if($validator->fails()){
36                 return Response()->json($validator->errors());
37             }
38
39             $Pelanggan = Pelanggan::create([
40                 'nama' => $req->nama,
41                 'ktp' => $req->ktp,
42                 'alamat' => $req->alamat,
43                 'telp' => $req->telp,
44             ]);
45             if($Pelanggan){
46                 return Response()->json(['status'=>true,'message'=>'ANDA SUCCESSFULLY JOIN WITH US']);
47             }
48             else{
49                 return Response()->json(['status'=>false,'message'=>'ANDA FAILED TO JOIN WITH US']);
50             }
51         }
52     }
53 }
```



```

54 public function update($id, Request $req)
55 {
56     if(Auth::user()->level=='admin'){
57         $validator=Validator::make($req->all(),
58         [
59             'nama' => 'required',
60             'ktp' => 'required',
61             'alamat' => 'required',
62             'telp' => 'required',
63         ]);
64         if($validator->fails()){
65             return Response()->json($validator->errors());
66         }
67         $Pelanggan=Pelanggan::where('id_pelanggan',$id)->update([
68             'nama' => $req->nama,
69             'ktp' => $req->ktp,
70             'alamat' => $req->alamat,
71             'telp' => $req->telp,
72         ]);
73         if($Pelanggan){
74             return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY CHANGED']);
75         }
76         else{
77             return Response()->json(['status'=>false,'message'=>'DATA FAILED TO CHANGE']);
78         }
79     }
80 }
81
82 public function delete($id)
83 {
84     if(Auth::user()->level=='admin'){
85         $Pelanggan=Pelanggan::where('id_pelanggan',$id)->delete();
86         if($Pelanggan){
87             return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY DELETED']);
88         }
89         else{
90             return Response()->json(['status'=>false,'message'=>'DATA FAILED TO DELETED']);
91         }
92     }
93 }
94 public function tampil()
95 {
96     if(Auth::user()->level=='admin'){
97         $pelanggan=Pelanggan::all();
98         $pelanggan = Pelanggan::count();
99         if($Pelanggan){
100             return Response()->json(['JUMLAH DATA'=>$pelanggan,'DATA'=>$pelanggan,'status'=>true]);
101         }
102         else{
103             return Response()->json(['status'=>false]);
104         }
105     }
106 }
107 }
108

```

## PELANGGAN MODEL

```

app > Pelanggan.php
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Pelanggan extends Model
8 {
9     protected $table = "pelanggan";
10     protected $primaryKey = "id_pelanggan";
11     protected $fillable = [
12         'id_pelanggan',
13         'nama',
14         'ktp',
15         'alamat',
16         'telp',
17     ];
18     public $timestamps = false;
19 }
20

```



## PEMINJAMAN CONTROLLER

```
app > Php > Controllers > PeminjamanController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Support\Facades\Validator;
6  use Illuminate\Http\Request;
7  use App\Peminjaman;
8  use App\Detail;
9  use App\User;
10 use App\Penis;
11 use Auth;
12 use DB;
13 use DateTime;
14
15 class PeminjamanController extends Controller
16 {
17     //CEK LEVEL / POSISI PETUGAS
18     public function cek(){
19         if(Auth::user()->level=='admin'){
20             $petugas=DB::table('petugas')->get();
21             return response()->json($petugas);
22         }else{
23             return response()->json(['== ANDA BUKAN ADMIN ==']);
24         }
25     }
26
27     public function store(Request $req)
28     {
29         if(Auth::user()->level=='admin'){
30             $validator = Validator::make($req->all(),
31             [
32                 'id_pelanggan' => 'required',
33                 'id_petugas' => 'required',
34                 'tgl_peminjaman' => 'required',
35                 'tgl_pengembalian' => 'required',
36             ]);
37
38             if($validator->fails()){
39                 return Response()->json($validator->errors());
40             }
41
42             $peminjaman = Peminjaman::create([
43                 'id_pelanggan' => $req->id_pelanggan,
44                 'id_petugas' => $req->id_petugas,
45                 'tgl_peminjaman' => $req->tgl_peminjaman,
46                 'tgl_pengembalian' => $req->tgl_pengembalian,
47             ]);
48             if($peminjaman){
49                 return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY ADDED']);
50             }
51             else{
52                 return Response()->json(['status'=>false,'message'=>'DATA FAILED TO ADDED']);
53             }
54         }
55     }
56 }
```

```

57 public function update($id, Request $req)
58 {
59     if(Auth::user()->level=='admin'){
60         $validator=Validator::make($req->all(),
61         [
62             'id_pelanggan' => 'required',
63             'id_petugas' => 'required',
64             'tgl_peminjaman' => 'required',
65             'tgl_pengembalian' => 'required',
66         ]);
67         if($validator->fails()){
68             return Response()->json($validator->errors());
69         }
70         $peminjaman=peminjaman::where('id_peminjaman',$id)->update([
71             'id_pelanggan' => $req->id_pelanggan,
72             'id_petugas' => $req->id_petugas,
73             'tgl_peminjaman' => $req->tgl_peminjaman,
74             'tgl_pengembalian' => $req->tgl_pengembalian,
75         ]);
76         if($peminjaman){
77             return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY CHANGED']);
78         }
79         else{
80             return Response()->json(['status'=>false,'message'=>'DATA FAILED TO CHANGE']);
81         }
82     }
83 }
84
85 public function delete($id)
86 {
87     if(Auth::user()->level=='admin'){
88         $peminjaman=peminjaman::where('id_peminjaman',$id)->delete();
89         if($peminjaman){
90             return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY DELETED']);
91         }
92         else{
93             return Response()->json(['status'=>false,'message'=>'DATA FAILED TO DELETED']);
94         }
95     }
96 }
97 public function tampil()
98 {
99     if(Auth::user()->level=='admin'){
100         $peminjaman =peminjaman::join('petugas','petugas.id','peminjaman.id_petugas')->join('pelanggan','pelanggan.id_pelanggan','peminjaman.id_pelanggan')->get();
101         $arr_data=array();
102         foreach ($peminjaman as $peminjaman){
103             $arr_data[]=array(
104                 'id_peminjaman'=> $peminjaman->id_peminjaman,
105                 'id_petugas'=> $peminjaman->id_petugas,
106                 'nama_petugas'=> $peminjaman->nama_petugas,
107                 'id_pelanggan'=> $peminjaman->id_pelanggan,
108                 'nama'=> $peminjaman->nama,
109                 'tgl_peminjaman'=> $peminjaman->tgl_peminjaman,
110                 'tgl_pengembalian'=> $peminjaman->tgl_pengembalian,
111             );
112         }
113         return Response()->json($arr_data);
114     }
115 }
116 }
117

```

## PEMINJAMAN CONTROLLER

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Peminjaman extends Model
8  {
9      protected $table = 'peminjaman';
10     protected $primaryKey = 'id_peminjaman';
11     protected $fillable = [
12         'id_peminjaman',
13         'id_pelanggan',
14         'id_petugas',
15         'tgl_peminjaman',
16         'tgl_pengembalian',
17     ];
18     public $timestamps = false;
19 }
20

```

## DETAIL CONTROLLER

```
app > Http > Controllers > PeminjamanController.php
114     return Response()->json($save_data);
115     }
116 }
117
118 public function store1(Request $req)
119 {
120
121     $validator = Validator::make($req->all(),
122     [
123         'id_peminjaman' => 'required',
124         'id_jenis'       => 'required',
125     ]);
126
127     if($validator->fails()){
128         return Response()->json($validator->errors());
129     }
130     $hari = Peminjaman::where('id_peminjaman',$req->id_peminjaman)->first();
131     $fdate = $hari->tgl_peminjaman;
132     $tdate = $hari->tgl_pengembalian;
133     $datetime1 = new DateTime($fdate);
134     $datetime2 = new DateTime($tdate);
135     $interval = $datetime1->diff($datetime2);
136     $berapa = $interval->format('%a');
137
138     $total1 = Jenis::where('id_jenis',$req->id_jenis)->first();
139     $total=$total1->harga_per_hari*$berapa;
140
141     $detail = detail::create([
142         'id_peminjaman' => $req->id_peminjaman,
143         'id_jenis'       => $req->id_jenis,
144         'peminjaman_hari' => $berapa,
145         'total'          => $total,
146     ]);
147     if($detail){
148         return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY ADDED']);
149     }
150     else{
151         return Response()->json(['status'=>false,'message'=>'DATA FAILED TO ADDED']);
152     }
153 }
154
155 public function update1($id, Request $req)
156 {
157     if(Auth::user()->level=='admin'){
158         $validator=Validator::make($req->all(),
159         [
160             'id_peminjaman' => 'required',
161             'id_jenis'       => 'required',
162         ]);
163
164         if($validator->fails()){
165             return Response()->json($validator->errors());
166         }
167         $hari = Peminjaman::where('id_peminjaman',$req->id_peminjaman)->first();
168         $fdate = $hari->tgl_peminjaman;
169         $tdate = $hari->tgl_pengembalian;
170         $datetime1 = new DateTime($fdate);
171         $datetime2 = new DateTime($tdate);
172         $interval = $datetime1->diff($datetime2);
173         $berapa = $interval->format('%a');
174
175         $total1 = Jenis::where('id_jenis',$req->id_jenis)->first();
176         $total=$req->harga_per_hari*$berapa;
177
178         $detail=detail::where('id_detail',$id)->update([
179             'id_peminjaman' => $req->id_peminjaman,
180             'id_jenis'       => $req->id_jenis,
181             'peminjaman_hari' => $berapa,
182             'total'          => $total,
183         ]);
184     }
```

```

84     if($detail){
85         return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY CHANGED']);
86     }
87     else{
88         return Response()->json(['status'=>false,'message'=>'DATA FAILED TO CHANGE']);
89     }
90 }
91 }
92 }
93
94 public function delete($id)
95 {
96     if(Auth::user()->level=='admin'){
97         $detail=$detail->where('id_detail',$id)->delete();
98     }
99     if($detail){
100         return Response()->json(['status'=>true,'message'=>'DATA SUCCESSFULLY DELETED']);
101     }
102     else{
103         return Response()->json(['status'=>false,'message'=>'DATA FAILED TO DELETED']);
104     }
105 }
106
107 public function tampil()
108 {
109     if(Auth::user()->level=='admin'){
110         $detail=$detail->join('jenis_mobil','jenis_mobil.id_jenis','detail_transaksi.id_jenis')->join('peminjaman','peminjaman.id_peminjaman','detail_transaksi.id_peminjaman')->join('mobil','mobil.id_jenis','jenis_mobil.id_jenis')->get();
111         $serv_data=array();
112         foreach ($detail as $detail){
113             $serv_data[]=array(
114                 'id_detail'=>$detail->id_detail,
115                 'id_jenis'=>$detail->id_jenis,
116                 'nama_jenis'=>$detail->nama_jenis,
117                 'nama_mobil'=>$detail->nama_mobil,
118                 'id_peminjaman'=>$detail->id_peminjaman,
119                 'nama_pem_hari'=>$detail->nama_pem_hari,
120                 'peminjaman_hari'=>$detail->peminjaman_hari,
121                 'total'=>$detail->total,
122             );
123         }
124         return Response()->json($serv_data);
125     }
126 }

```

## DETIAL MODEL

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Detail extends Model
8  {
9      protected $table = "detail_transaksi";
10     protected $primaryKey = "id_detail";
11     protected $fillable = [
12         'id_detail',
13         'id_peminjaman',
14         'id_jenis',
15         'peminjaman_hari',
16         'total',
17     ];
18     public $timestamps = false;
19 }
20
21

```

SCREENSHOOTAN DENGAN DATABASE DATANYA BERBEDA KARENA WAKTU MELAKUKAN EKSEKUSI DETAIL TRANSAKSI , TABELNYA ERROR JADI SAYA BIKIN DATABASE BARU DAN SAYA G ANGE SS ULANG. TERIMAKASIH.INI SAYA SS BESERTA TRANSAKSI(PEMINJAMAN) KARENA SUDAH TERLANJUR.