

Heni Yunar Larasati'

XI RPL 6 / 16

CRUD Master

1. Controller
 - a. Petugas

PetugasController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\User;
6  use App\Petugas_model;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Support\Facades\Validator;
10 use JWTAuth;
11 use Tymon\JWTAuth\Exceptions\JWTException;
12 use Auth;
13
14 class PetugasController extends Controller
15 {
16     public function login(Request $request)
17     {
18         $credentials = $request->only('username', 'password');
19
20         try {
21             if (! $token = JWTAuth::attempt($credentials)) {
22                 return response()->json(['error' => 'invalid_credentials'], 400);
23             }
24         } catch (JWTException $e) {
25             return response()->json(['error' => 'could_not_create_token'], 500);
26         }
27
28         return response()->json(compact('token'));
29     }
30 }
```

```
31 public function register(Request $request)
32 {
33     $validator = Validator::make($request->all(), [
34         'nama_petugas' => 'required|string|max:255',
35         'telp' => 'required|string|max:255',
36         'username' => 'required|string|max:255|unique:petugas',
37         'password' => 'required|string|min:6|confirmed',
38         'level' => 'required|string|max:255',
39
40     ]);
41
42     if($validator->fails()){
43         return response()->json($validator->errors()->toJson(), 400);
44     }
45
46     $user = User::create([
47         'nama_petugas' => $request->get('nama_petugas'),
48         'telp' => $request->get('telp'),
49         'username' => $request->get('username'),
50         'password' => Hash::make($request->get('password')),
51         'level' => $request->get('level'),
52
53     ]);
54
55     $token = JWTAuth::fromUser($user);
56
57     return response()->json(compact('user', 'token'), 201);
58 }
```

PetugasController.php

```
60 public function getAuthenticatedUser()
61 {
62     try {
63
64         if (! $user = JWTAuth::parseToken()->authenticate()) {
65             return response()->json(['user_not_found'], 404);
66         }
67
68     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
69
70         return response()->json(['token_expired'], $e->getStatusCode());
71
72     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
73
74         return response()->json(['token_invalid'], $e->getStatusCode());
75
76     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
77
78         return response()->json(['token_absent'], $e->getStatusCode());
79
80     }
81
82     return response()->json(compact('user'));
83 }
```

PetugasController.php

```
85 public function tampil(){
86     if(Auth::user()->level=="admin"){
87         $datas = Petugas_model::get();
88         $count = $datas->count();
89         $film = array();
90         $status = 1;
91         foreach ($datas as $dt_sw){
92             $film[] = array(
93                 'id' => $dt_sw->id,
94                 'nama_petugas' => $dt_sw->nama_petugas,
95                 'telp' => $dt_sw->telp,
96                 'username' => $dt_sw->username,
97                 'password' => $dt_sw->password,
98                 'level' => $dt_sw->level,
99                 'created_at' => $dt_sw->created_at,
100                 'updated_at' => $dt_sw->updated_at,
101             );
102         }
103         return Response()->json(compact('count','film'));
104     } else {
105         return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
106     }
107 }
108
109 public function update($id,Request $req)
110 {
111     if(Auth::user()->level=="admin"){
112         $validator=Validator::make($req->all(),
113         [
114             'nama_petugas' => 'required',
115             'telp' => 'required',
```

b. Penyewa

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Penyewa_model;
7  use Illuminate\Support\Facades\Validator;
8  use Auth;
9
10 class PenyewaController extends Controller
11 {
12     // CREATE
13     public function store(Request $req)
14     {
15         if(Auth::user()->level=="admin"){
16             $validator = Validator::make($req->all(),
17             [
18                 'nama' => 'required',
19                 'no_ktp' => 'required',
20                 'alamat' => 'required',
21                 'telp' => 'required'
22             ]);
23             if($validator->fails()){
24                 return Response()->json($validator->errors());
25             }
26             $simpan = Penyewa_model::create([
27                 'nama' => $req->nama,
28                 'no_ktp' => $req->no_ktp,
29                 'alamat' => $req->alamat,
30                 'telp' => $req->telp,
31
32             ]);
33             return Response()->json(['status'=> 'Data penyewa berhasil dimasukkan']);
34         } else {
```

```
34     } else {
35         return Response()->json(['status'=> 'Data penyewa gagal dimasukkan, anda bukan admin']);
36     }
37 }
38
39 //READ
40 public function tampil(){
41     if(Auth::user()->level=="admin"){
42         $datas = Penyewa_model::get();
43         $count = $datas->count();
44         $penyewa = array();
45         $status = 1;
46         foreach ($datas as $dt_sw){
47             $penyewa[] = array(
48                 'id' => $dt_sw->id,
49                 'nama' => $dt_sw->nama,
50                 'no_ktp' => $dt_sw->no_ktp,
51                 'alamat' => $dt_sw->alamat,
52                 'telp' => $dt_sw->telp,
53                 'created_at' => $dt_sw->created_at,
54                 'updated_at' => $dt_sw->updated_at,
55             );
56         }
57         return Response()->json(compact('count','penyewa'));
58     } else {
59         return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
60     }
61 }
62
63 // UPDATE
64 public function update($id,Request $req)
65 {
66     if(Auth::user()->level=="admin"){
67         $validator=Validator::make($req->all(),
```


PenyewaController.php	api.php	JenisController.php	MobilController.php	Detail
<pre> 64 public function update(\$id, Request \$req) 65 { 66 if(Auth::user()->level=="admin"){ 67 \$validator=Validator::make(\$req->all(), 68 [69 'nama' => 'required', 70 'no_ktp' => 'required', 71 'alamat' => 'required', 72 'telp' => 'required', 73]); 74 if(\$validator->fails()){ 75 return Response()->json(\$validator->errors()); 76 } 77 \$ubah=Penyewa_model::where('id',\$id)->update([78 'nama' => \$req->nama, 79 'no_ktp' => \$req->no_ktp, 80 'alamat' => \$req->alamat, 81 'telp' => \$req->telp, 82]); 83 return Response()->json([84 'message'=>'Data penyewa berhasil diubah']); 85 } else { 86 return Response()->json([87 'message'=>'Data penyewa gagal diubah, anda bukan admin']); 88 } 89 } 90 91 //DELETE 92 public function delete(\$id) 93 { 94 if(Auth::user()->level=="admin"){ 95 \$hapus=Penyewa_model::where('id',\$id)->delete(); 96 return Response()->json([97 'message'=>'Data penyewa berhasil dihapus']); 98 } else { 99 return Response()->json([100 'message'=>'Data penyewa gagal dihapus, anda bukan admin']); 101 } 102 } 103 } 104 } </pre>				

c. Mobil


```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Mobil_model;
7  use Illuminate\Support\Facades\Validator;
8  use Auth;
9
10 class MobilController extends Controller
11 {
12     // CREATE
13     public function store(Request $req)
14     {
15         if(Auth::user()->level=="admin"){
16             $validator = Validator::make($req->all(),
17             [
18                 'id_jenis' => 'required',
19                 'merk' => 'required',
20                 'plat' => 'required',
21                 'kondisi' => 'required',
22                 'keterangan' => 'required',
23             ]);
24             if($validator->fails()){
25                 return Response()->json($validator->errors());
26             }
27             $simpan = Mobil_model::create([
28                 'id_jenis' => $req->id_jenis,
29                 'merk' => $req->merk,
30                 'plat' => $req->plat,
31                 'kondisi' => $req->kondisi,
32                 'keterangan' => $req->keterangan,
33             ]);
34             return Response()->json(['status'=> 'Data mobil berhasil dimasukkan']);
```

```
35     } else {
36         return Response()->json(['status'=> 'Data mobil gagal dimasukkan,
37                                     anda bukan admin']);
38     }
39 }
40
41 //READ
42 public function tampil(){
43     if(Auth::user()->level!="admin"){
44         $datas = Mobil_model::get();
45         $count = $datas->count();
46         $jenis = array();
47         $status = 1;
48         foreach ($datas as $dt_sw){
49             $jenis[] = array(
50                 'id' => $dt_sw->id,
51                 'id_jenis' => $dt_sw->id_jenis,
52                 'merk' => $dt_sw->merk,
53                 'plat' => $dt_sw->plat,
54                 'kondisi' => $dt_sw->kondisi,
55                 'keterangan' => $dt_sw->keterangan,
56                 'created_at' => $dt_sw->created_at,
57                 'updated_at' => $dt_sw->updated_at,
58             );
59         }
60         return Response()->json(compact('count','jenis'));
61     } else {
62         return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
63     }
64 }
65
66 // UPDATE
67 public function update($id,Request $req)
68 {
```

```

67 public function update($id, Request $req)
68 {
69     if(Auth::user()->level=="admin"){
70         $validator=Validator::make($req->all(),
71         [
72             'id_jenis' => 'required',
73             'merk' => 'required',
74             'plat' => 'required',
75             'kondisi' => 'required',
76             'keterangan' => 'required',
77         ]);
78         if($validator->fails()){
79             return Response()->json($validator->errors());
80         }
81         $ubah=Mobil_model::where('id',$id)->update([
82             'id_jenis' => $req->id_jenis,
83             'merk' => $req->merk,
84             'plat' => $req->plat,
85             'kondisi' => $req->kondisi,
86             'keterangan' => $req->keterangan,
87         ]);
88         return Response()->json([
89             'message'=>'Data mobil berhasil diubah']);
90     } else {
91         return Response()->json([
92             'message'=>'Data mobil gagal diubah,
93             anda bukan admin']);
94     }
95 }
96
97 //DELETE
98 public function delete($id)
99 {
100     if(Auth::user()->level=="admin"){

```

```

98 public function delete($id)
99 {
100     if(Auth::user()->level=="admin"){
101         $hapus=Mobil_model::where('id',$id)->delete();
102         return Response()->json([
103             'message'=>'Data mobil berhasil dihapus']);
104     } else {
105         return Response()->json([
106             'message'=>'Data mobil gagal dihapus,
107             anda bukan admin']);
108     }
109 }
110 }
111 }
112

```

d. Jenis mobil

JenisController.php	MobilController.php	DetailController.php	Mo
<pre>1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 use App\Jenis_model; 7 use Illuminate\Support\Facades\Validator; 8 use Auth; 9 10 class JenisController extends Controller 11 { 12 // CREATE 13 public function store(Request \$req) 14 { 15 if(Auth::user()->level=="admin"){ 16 \$validator = Validator::make(\$req->all(), 17 [18 'jenis' => 'required', 19]); 20 if(\$validator->fails()){ 21 return Response()->json(\$validator->errors()); 22 } 23 \$simpan = Jenis_model::create([24 'jenis' => \$req->jenis, 25]); 26 return Response()->json(['status'=> 'Data jenis berhasil dimasukkan']); 27 } else { 28 return Response()->json(['status'=> 'Data jenis gagal dimasukkan, anda bukan admin']); 29 } 30 } 31 32 //READ 33 public function tampil(){ 34 if(Auth::user()->level=="admin"){</pre>			

```
33 public function tampil(){
34     if(Auth::user()->level=="admin"){
35         $datas = Jenis_model::get();
36         $count = $datas->count();
37         $jenis = array();
38         $status = 1;
39         foreach ($datas as $dt_sw){
40             $jenis[] = array(
41                 'id' => $dt_sw->id,
42                 'jenis' => $dt_sw->jenis,
43             );
44         }
45         return Response()->json(compact('count','jenis'));
46     } else {
47         return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
48     }
49 }
50
51 // UPDATE
52 public function update($id,Request $req)
53 {
54     if(Auth::user()->level=="admin"){
55         $validator=Validator::make($req->all(),
56         [
57             'jenis' => 'required',
58         ]);
59         if($validator->fails()){
60             return Response()->json($validator->errors());
61         }
62         $ubah=Jenis_model::where('id',$id)->update([
63             'jenis' => $req->jenis,
64         ]);
65         return Response()->json([
66             'message'=>'Data jenis berhasil diubah']);
67     }
```



```

67     } else {
68         return Response()->json([
        'message'=>'Data jenis gagal diubah, anda bukan admin']);
69     }
70 }
71
72 //DELETE
73 public function delete($id)
74 {
75     if(Auth::user()->level=="admin"){
76         $hapus=Jenis_model::where('id',$id)->delete();
77         return Response()->json([
78             'message'=>'Data jenis berhasil dihapus']);
79     } else {
80         return Response()->json([
81             'message'=>'Data jenis gagal dihapus, anda bukan admin']);
82     }
83 }
84
85 }
86 }

```

2. Model
 - a. Petugas

```

User.php
1  <?php
2
3  namespace App;
4
5  use Illuminate\Notifications\Notifiable;
6  use Illuminate\Contracts\Auth\MustVerifyEmail;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Tymon\JWTAuth\Contracts\JWTSubject;
9
10 class User extends Authenticatable implements JWTSubject
11 {
12     protected $table = "petugas";
13     // public $timestamps = false;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var array
19      */
20     protected $fillable = [
21         'nama_petugas', 'telp', 'username', 'password', 'level'
22     ];
23
24     /**
25      * The attributes that should be hidden for arrays.
26      *
27      * @var array
28      */
29     protected $hidden = [
30         'password', 'remember_token',
31     ];
32
33     public function getJWTIdentifier()
34     {
35         return $this->getKey();
36     }
37     public function getJWTCustomClaims()
38     {
39         return [];
40     }
41 }

```

```

Petugas_model.php
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Petugas_model extends Model
8  {
9      protected $table="petugas";
10     protected $primaryKey="id";
11     protected $fillable = ['nama_petugas','telp','username','password','level'];
12 }

```

b. Penyewa

```
Penyewa_model.php
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Penyewa_model extends Model
8  {
9      protected $table="penyewa";
10     protected $primaryKey="id";
11     protected $fillable = [
12         'nama', 'no_ktp', 'alamat', 'telp',
13     ];
14 }
15
```

c. Mobil

```
Mobil_model.php
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Mobil_model extends Model
8  {
9      protected $table="mobil";
10     protected $primaryKey="id";
11     protected $fillable = [
12         'id_jenis', 'merk', 'plat', 'kondisi', 'keterangan',
13     ];
14 }
15
```

d. Jenis mobil

```
Jenis_model.php
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Jenis_model extends Model
8  {
9      protected $table="jenis";
10     protected $primaryKey="id";
11     protected $fillable = [
12         'jenis',
13     ];
14 }
15
```

3. API

```
16 Route::middleware('auth:api')->get('/user', function (Request $request) {
17     return $request->user();
18 });
19 Route::post('register', 'PetugasController@register');
20 Route::post('login', 'PetugasController@login');
21 Route::get('/', function() {
22     return Auth::user()->level;
23 })->middleware('jwt.verify');
24 Route::get('petugas', 'PetugasController@tampil')->middleware('jwt.verify');
25 Route::put('petugas/{id}', 'PetugasController@update')->middleware('jwt.verify');
26 Route::delete('petugas/{id}', 'PetugasController@delete')->middleware('jwt.verify');
27
28 Route::post('penyewa', 'PenyewaController@store')->middleware('jwt.verify');
29 Route::get('penyewa', 'PenyewaController@tampil')->middleware('jwt.verify');
30 Route::put('penyewa/{id}', 'PenyewaController@update')->middleware('jwt.verify');
31 Route::delete('penyewa/{id}', 'PenyewaController@delete')->middleware('jwt.verify');
32
33 Route::post('mobil', 'MobilController@store')->middleware('jwt.verify');
34 Route::get('mobil', 'MobilController@tampil')->middleware('jwt.verify');
35 Route::put('mobil/{id}', 'MobilController@update')->middleware('jwt.verify');
36 Route::delete('mobil/{id}', 'MobilController@delete')->middleware('jwt.verify');
37
38 Route::post('jenis', 'jenisController@store')->middleware('jwt.verify');
39 Route::get('jenis', 'jenisController@tampil')->middleware('jwt.verify');
40 Route::put('jenis/{id}', 'jenisController@update')->middleware('jwt.verify');
41 Route::delete('jenis/{id}', 'jenisController@delete')->middleware('jwt.verify');
```

4. Hasil

a. Petugas

1. Register

POST localhost:8000/api/register

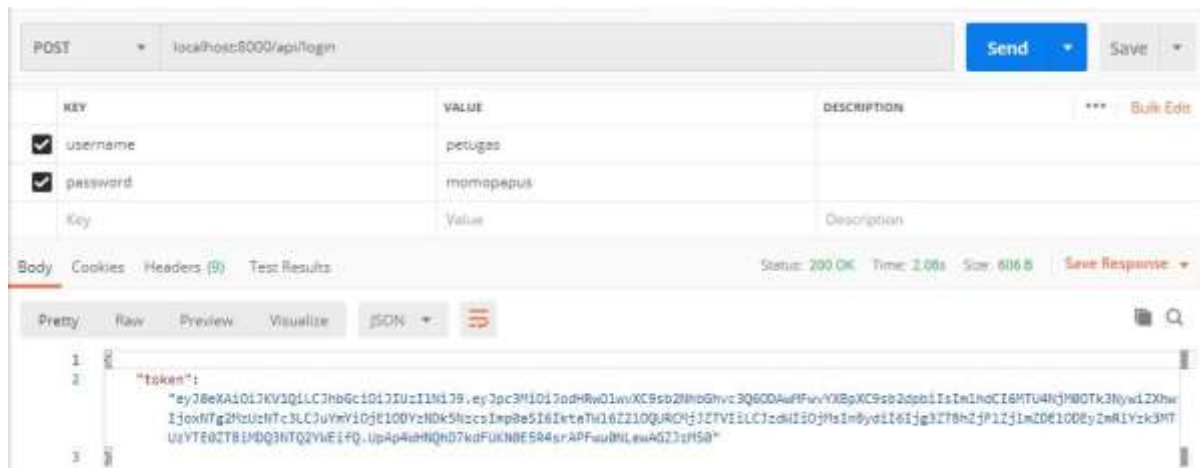
Key	Value	Description
<input checked="" type="checkbox"/> nama_petugas	Petugas 1	
<input checked="" type="checkbox"/> telp	08087654321	
<input checked="" type="checkbox"/> username	petugas	
<input checked="" type="checkbox"/> password	momopapus	
<input checked="" type="checkbox"/> password_confirmation	momopapus	
<input checked="" type="checkbox"/> level	petugas	

Body Cookies Headers (9) Test Results Status: 201 Created Time: 19.15s Size: 788 B Save Response

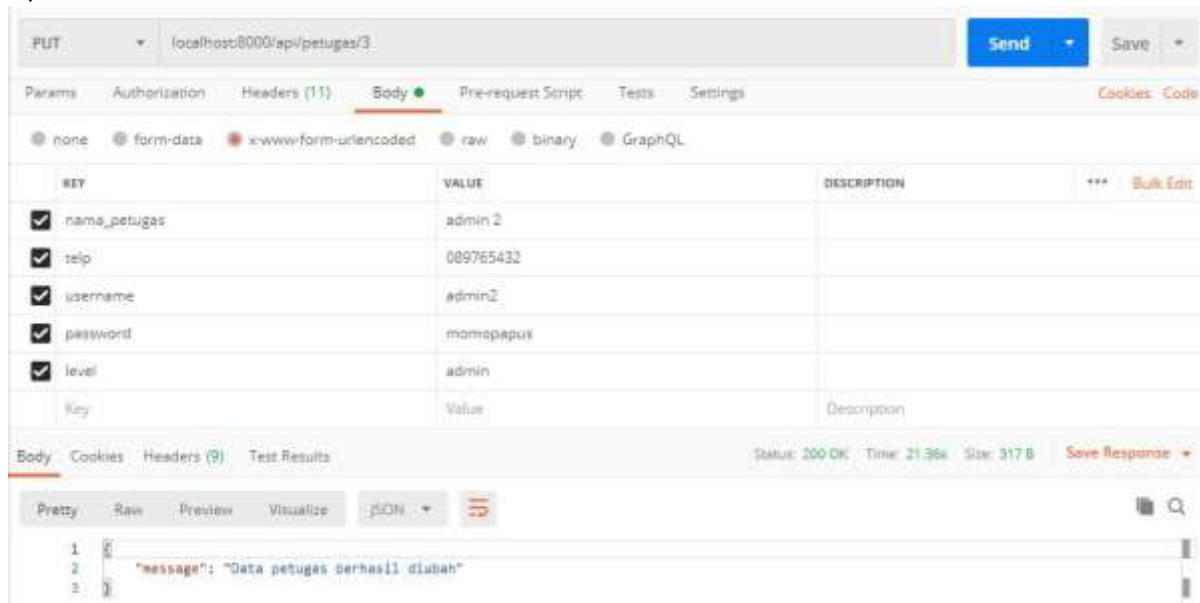
Pretty Raw Preview Visualize JSON

```
1 {
2   "user": {
3     "nama_petugas": "Petugas 1",
4     "telp": "08087654321",
5     "username": "petugas",
6     "level": "petugas",
7     "updated_at": "2020-04-08 12:43:58",
8     "created_at": "2020-04-08 12:43:58",
9     "id": 3
10  }
```

2. Login



3. Update



4. Delete



b. Penyewa

1. Create

POST localhost:8000/api/penyewa Send Save

<input checked="" type="checkbox"/>	nama	Text	Penyewa 3	
<input checked="" type="checkbox"/>	no_ktp		1234567892	
<input checked="" type="checkbox"/>	alamat		Jl Aroma Panden 2	
<input checked="" type="checkbox"/>	telep		089755435543	
<input type="checkbox"/>	level		admin	
<input type="checkbox"/>	password_confirmation		momopapus	
	Key		Value	Description

Body Cookies Headers [9] Test Results Status: 200 OK Time: 1451ms Size: 320 B Save Response

Pretty Raw Preview Visualize JSON ≡

```

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

```

2. Read

GET localhost:8000/api/penyewa Send Save

Pretty Raw Preview Visualize JSON ≡

```

1 {
2   "count": 3,
3   "penyewa": [
4     {
5       "id": 1,
6       "nama": "Penyewa 1",
7       "no_ktp": "1234567890",
8       "alamat": "Jl Aroma Panden",
9       "telep": "089755435543",
10      "created_at": "2020-04-17T13:24:11.000000Z",
11      "updated_at": "2020-04-17T13:24:11.000000Z"
12    },
13    {
14      "id": 2,
15      "nama": "Penyewa 2",
16      "no_ktp": "1234567891",
17      "alamat": "Jl Aroma Panden 1",
18      "telep": "089755435544",
19      "created_at": "2020-04-17T13:24:33.000000Z",
20      "updated_at": "2020-04-17T13:24:33.000000Z"
21    },
22    {
23      "id": 3,
24      "nama": "Penyewa 3",
25      "no_ktp": "1234567892",
26      "alamat": "Jl Aroma Panden 2",
27      "telep": "089755435543",
28      "created_at": "2020-04-17T13:24:55.000000Z",
29      "updated_at": "2020-04-17T13:24:55.000000Z"
30    }
31  ]
32 }

```

3. Update

PUT localhost:8000/api/penyewa/1 Send Save

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	nama	Penyewa 0
<input checked="" type="checkbox"/>	no_ktp	1234567899
<input checked="" type="checkbox"/>	alamat	Jl Baru ini
<input checked="" type="checkbox"/>	telep	087766554432
	Key	Value

Body Cookies Headers [0] Test Results Status: 200 OK Time: 1978ms Size: 317 B Save Response

Pretty Raw Preview Visualize JSON ≡

```

1 {
2   "message": "Data penyewa berhasil diubah"
3 }

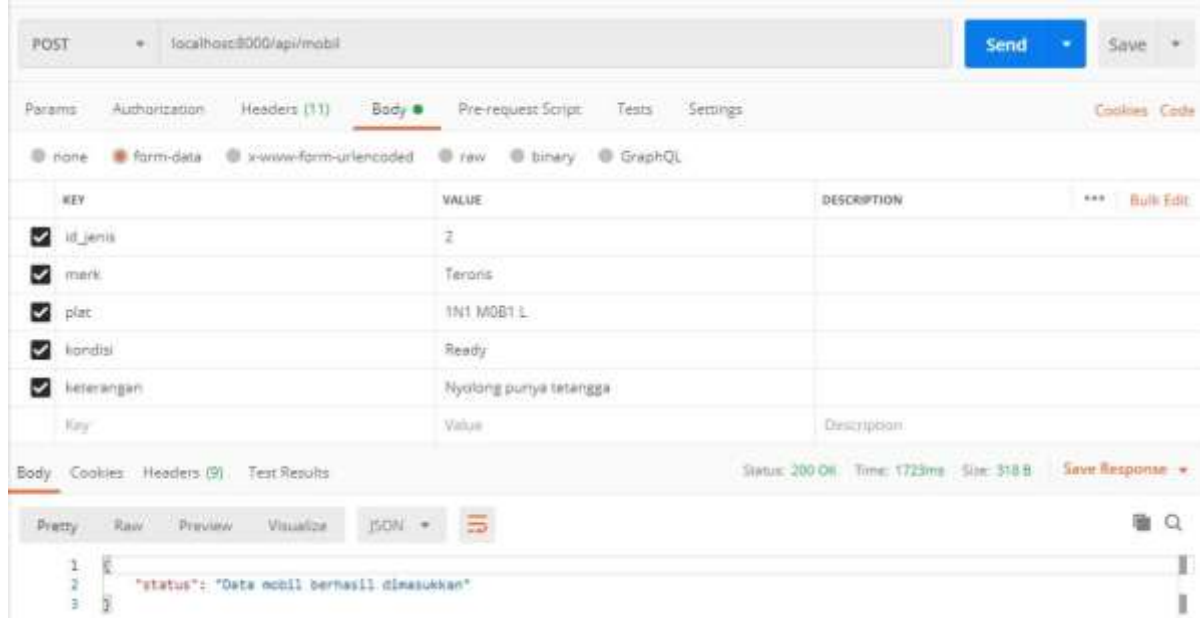
```

4. Delete

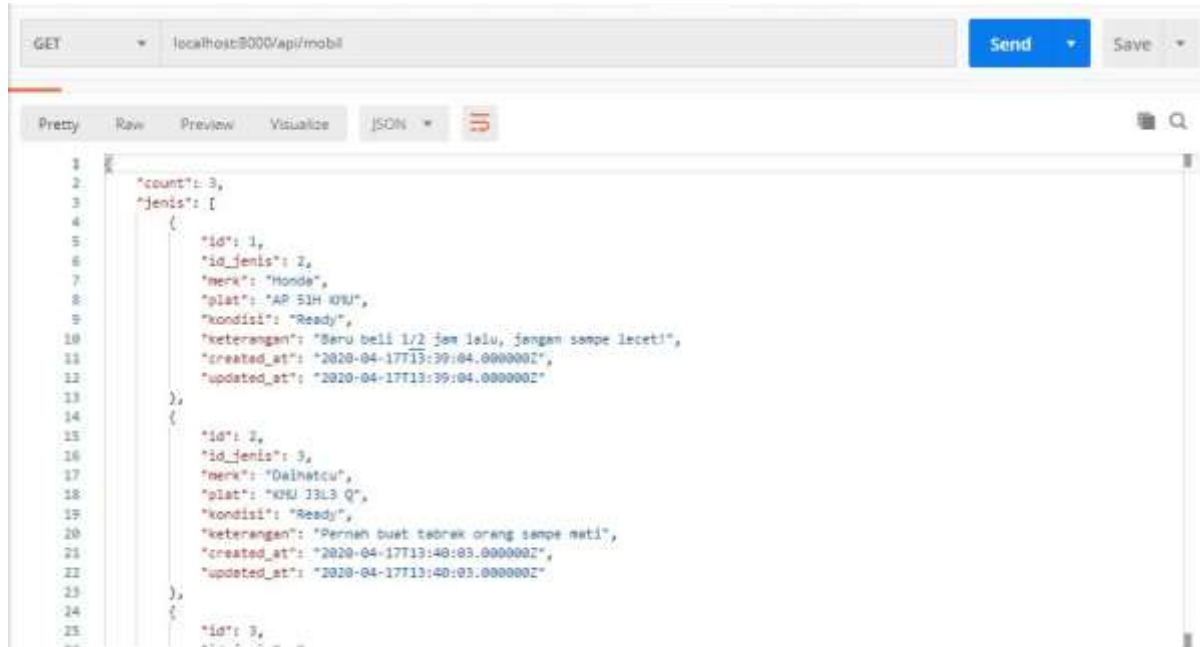


c. Mobil

1. Create



2. Read



3. Update

PUT localhost:8000/api/mobil/1 Send Save

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id_jenis	3	
<input checked="" type="checkbox"/> merk	Toyota	
<input checked="" type="checkbox"/> plat	GT W3H 3	
<input checked="" type="checkbox"/> kondisi	Not ready	
<input checked="" type="checkbox"/> keterangan	Katanya sih bagus	
Key	Value	Description

Body Cookies Headers (9) Test Results Status: 200 OK Time: 1974ms Size: 315 B Save Response

Pretty Raw Preview Visualize JSON ⋮

```

1 {
2   "message": "Data mobil berhasil diubah"
3 }

```

4. Delete

DELETE localhost:8000/api/mobil/1 Send Save

Pretty Raw Preview Visualize JSON ⋮

```

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

```

d. Jenis

1. Create

POST localhost:8000/api/jenis Send Save

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies Code

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> jenis	city car	
Key	Value	Description

Body Cookies Headers (9) Test Results Status: 200 OK Time: 1614ms Size: 313 B Save Response

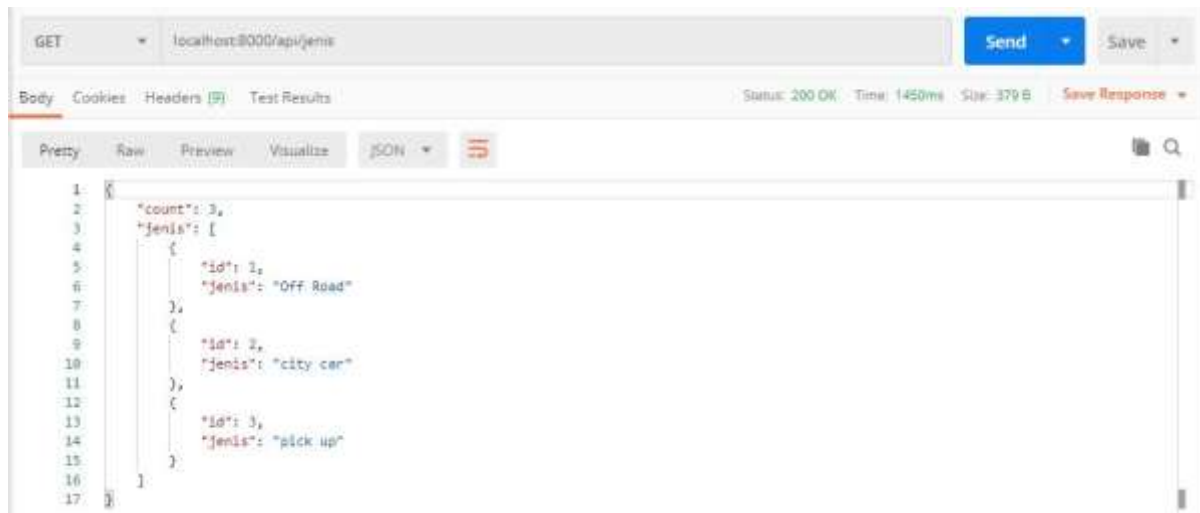
Pretty Raw Preview Visualize JSON ⋮

```

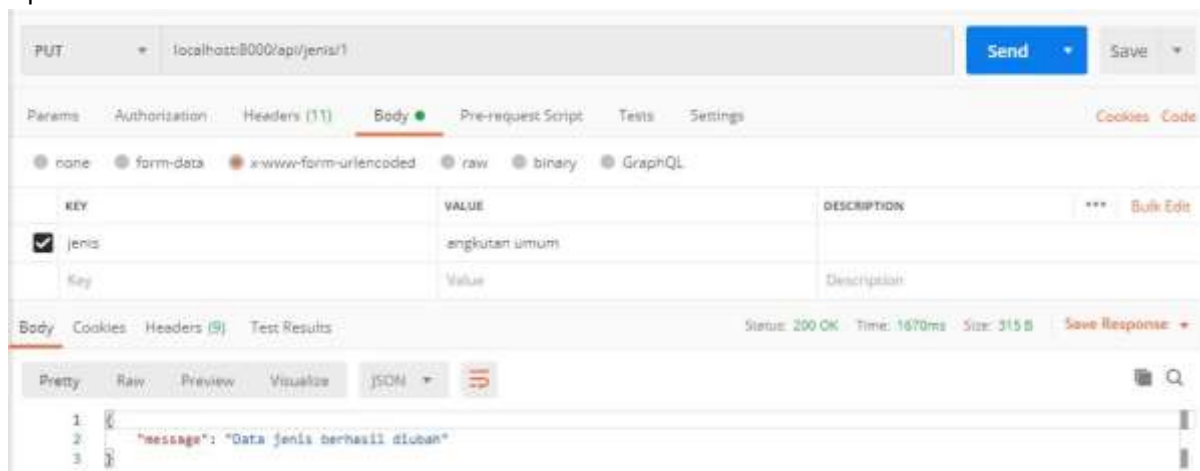
1 {
2   "status": "Data jenis berhasil dimasukkan"
3 }

```

2. Read



3. Update



4. Delete

