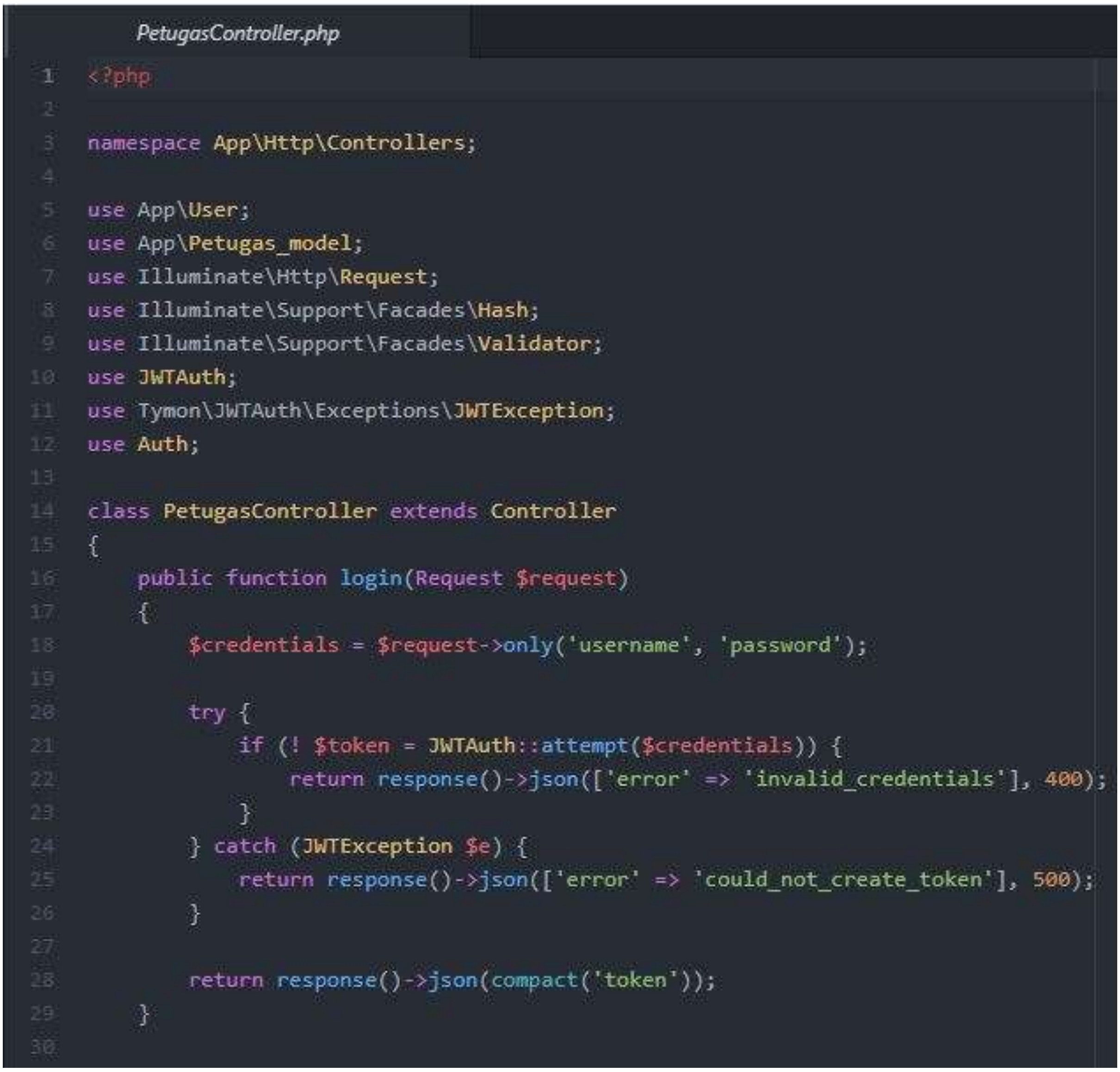
Heni Yunar Larasati’

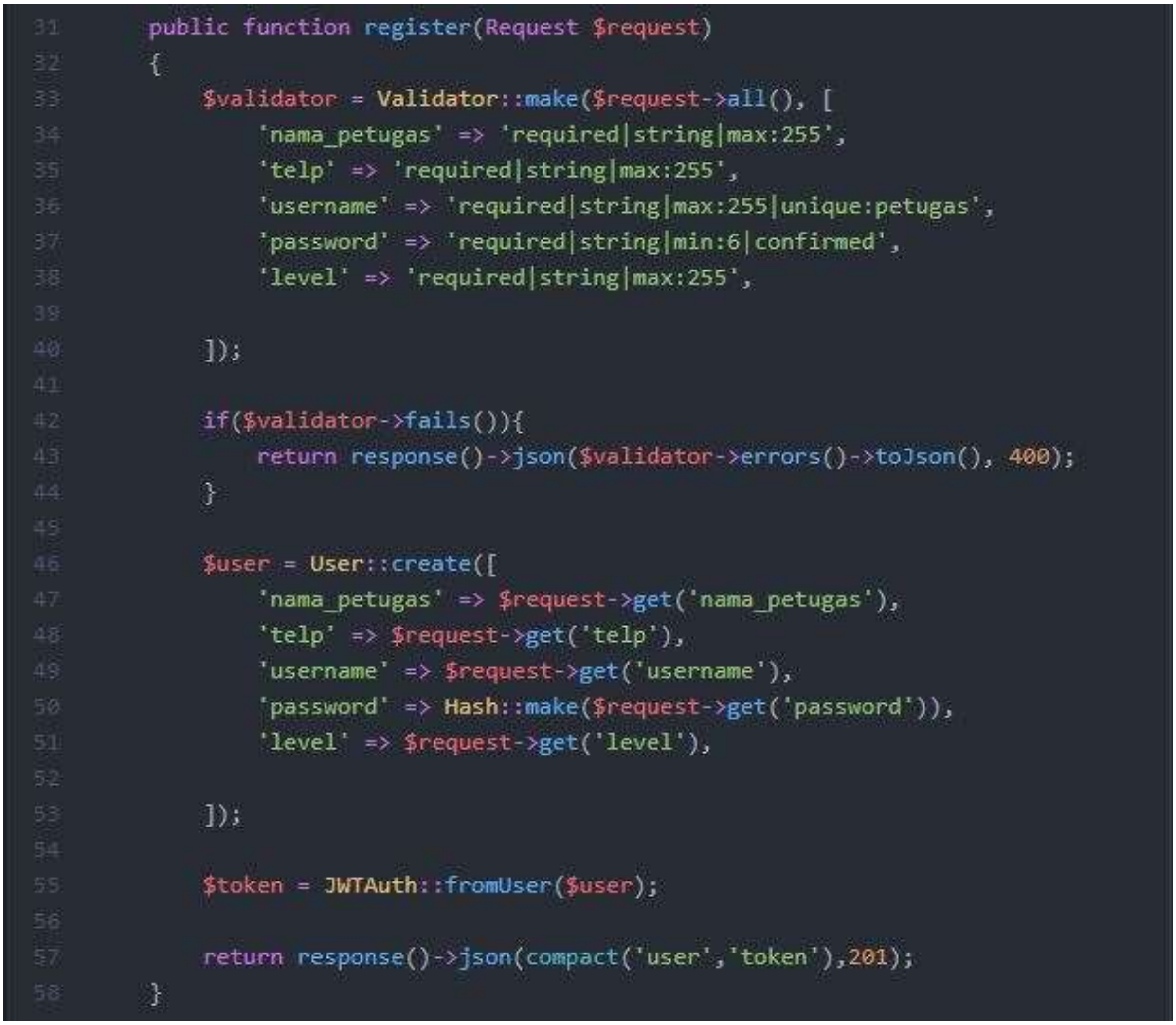
XI RPL 6 / 16

CRUD Master

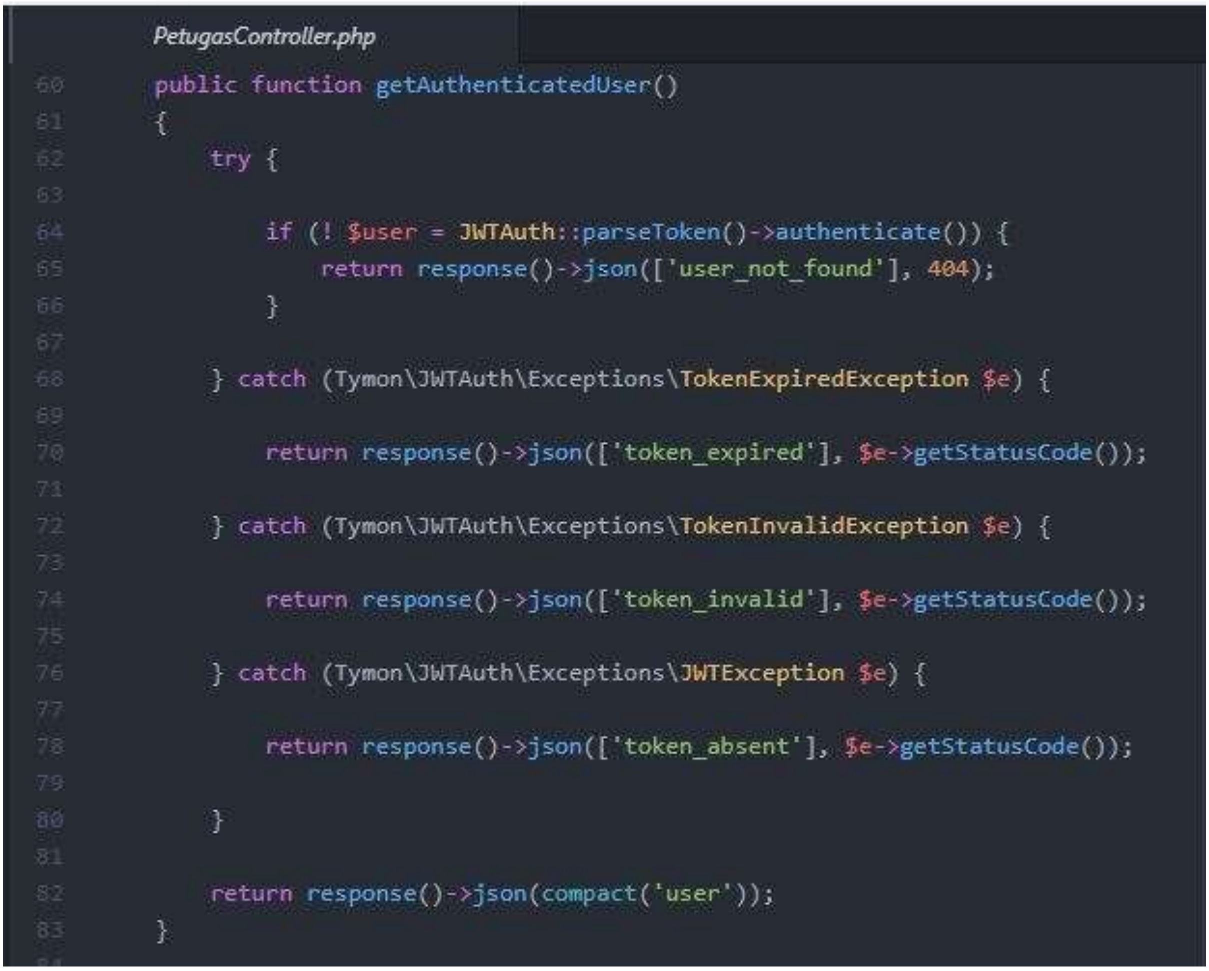
1. Controller
    a. Petugas



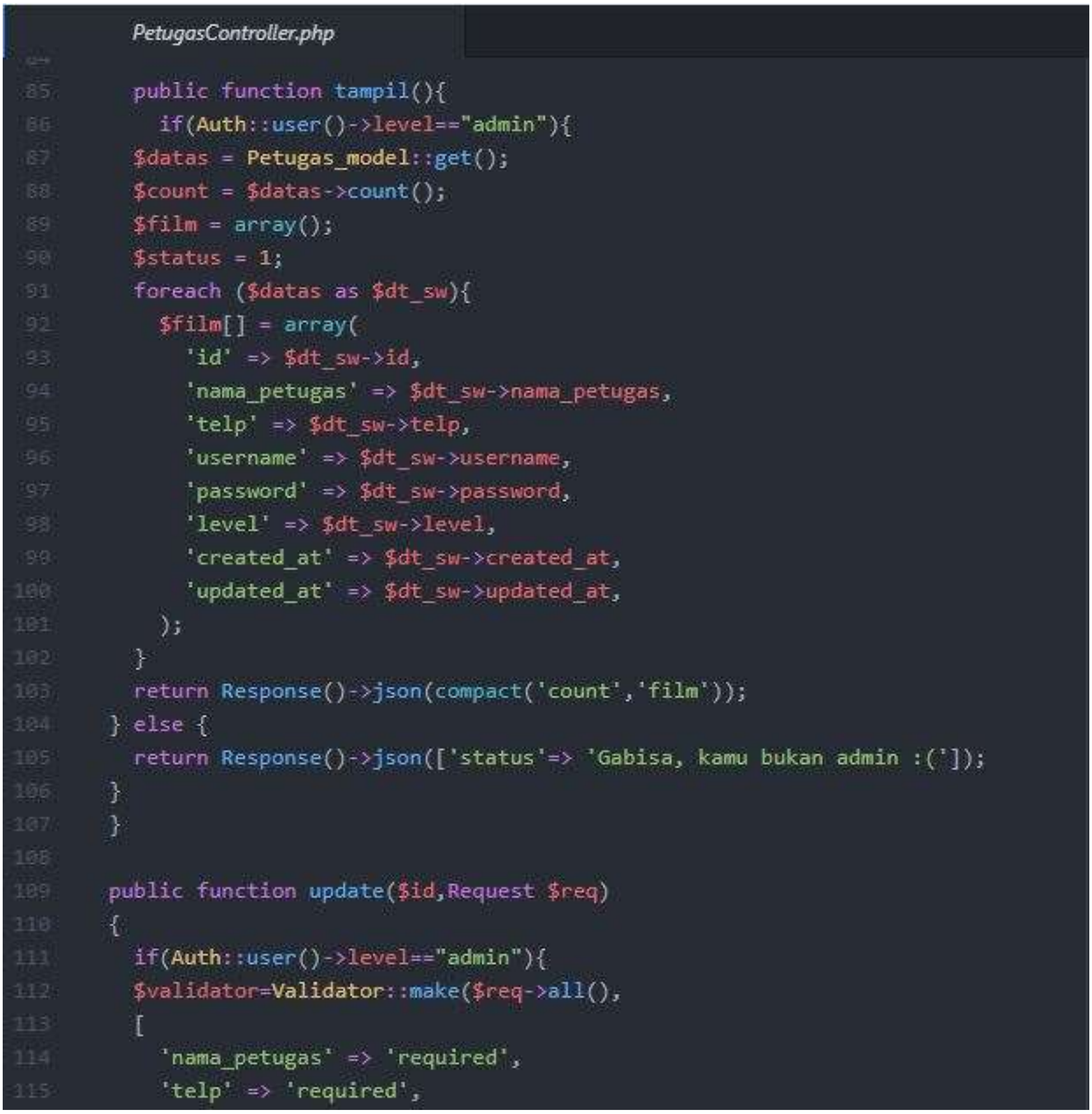
```php
<?php

namespace App\Http\Controllers;

use App\User;
use App\Petugas_model;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use JWTAuth;
use Tymon\JWTAuth\Exceptions\JWTException;
use Auth;

class PetugasController extends Controller
{
    public function login(Request $request)
    {
        $credentials = $request->only('username', 'password');

        try {
            if (! $token = JWTAuth::attempt($credentials)) {
                return response()->json(['error' => 'invalid_credentials'], 400);
            }
        } catch (JWTException $e) {
            return response()->json(['error' => 'could_not_create_token'], 500);
        }

        return response()->json(compact('token'));
    }

```



```php
    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'nama_petugas' => 'required|string|max:255',
            'telp' => 'required|string|max:255',
            'username' => 'required|string|max:255|unique:petugas',
            'password' => 'required|string|min:6|confirmed',
            'level' => 'required|string|max:255',

        ]);

        if($validator->fails()){
            return response()->json($validator->errors()->toJson(), 400);
        }

        $user = User::create([
            'nama_petugas' => $request->get('nama_petugas'),
            'telp' => $request->get('telp'),
            'username' => $request->get('username'),
            'password' => Hash::make($request->get('password')),
            'level' => $request->get('level'),

        ]);

        $token = JWTAuth::fromUser($user);

        return response()->json(compact('user','token'),201);
    }
```



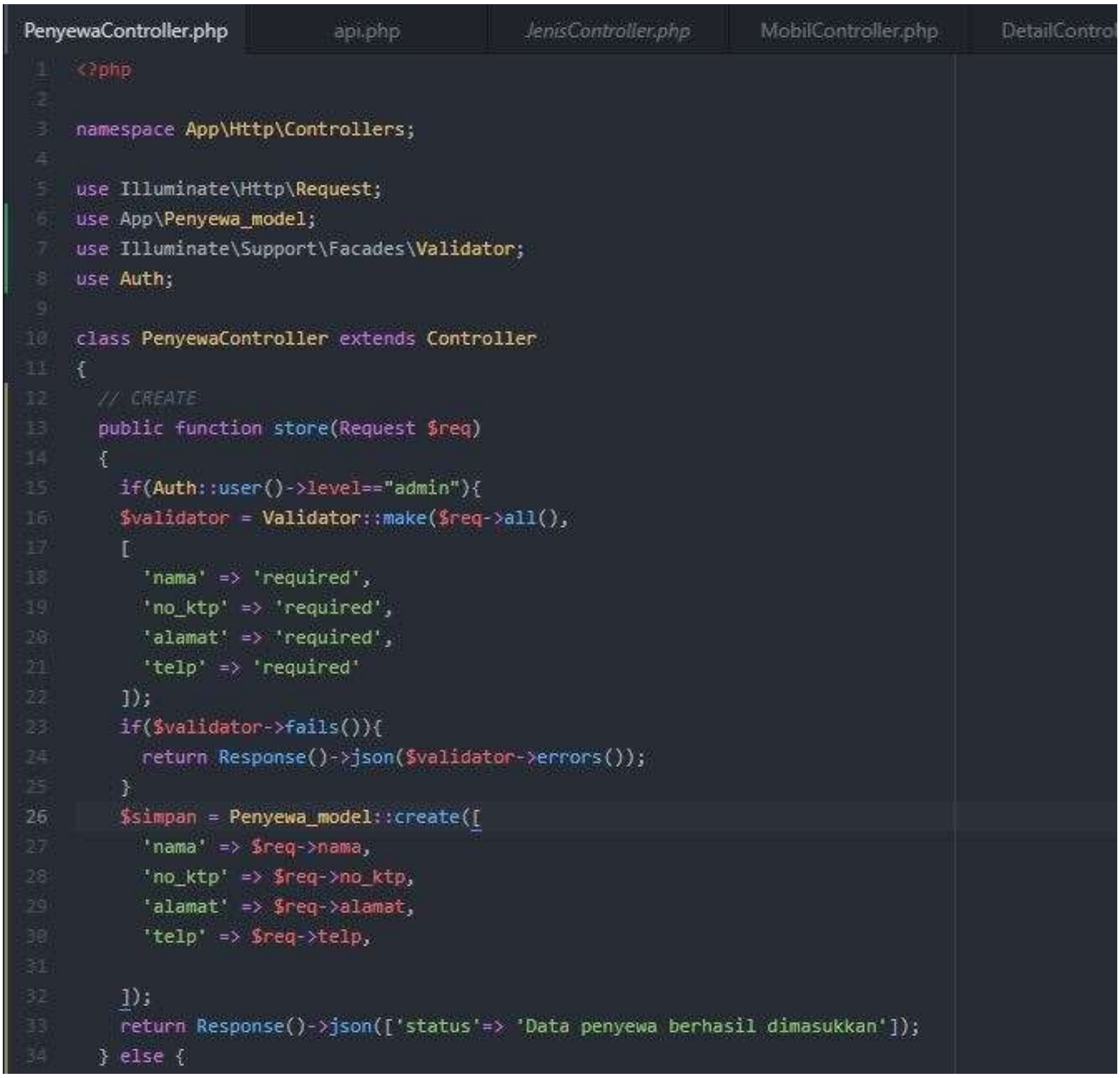
```php
    public function getAuthenticatedUser()
    {
        try {

            if (! $user = JWTAuth::parseToken()->authenticate()) {
                return response()->json(['user_not_found'], 404);
            }

        } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {

            return response()->json(['token_expired'], $e->getStatusCode());

        } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {

            return response()->json(['token_invalid'], $e->getStatusCode());

        } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {

            return response()->json(['token_absent'], $e->getStatusCode());

        }

        return response()->json(compact('user'));
    }
```



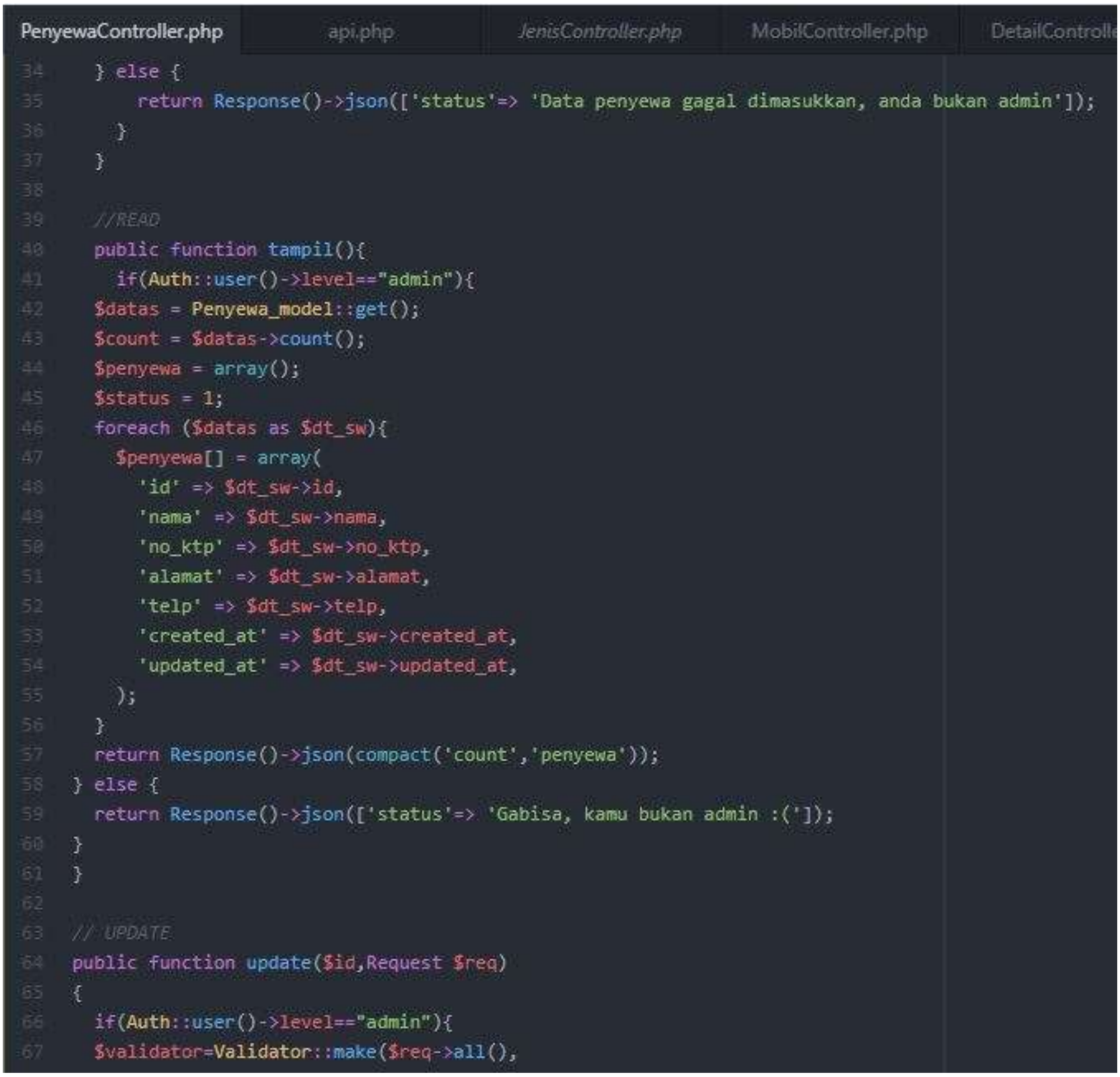
PetugasController.php

```php
      public function tampil(){
        if(Auth::user()->level=="admin"){
      $datas = Petugas_model::get();
      $count = $datas->count();
      $film = array();
      $status = 1;
      foreach ($datas as $dt_sw){
        $film[] = array(
          'id' => $dt_sw->id,
          'nama_petugas' => $dt_sw->nama_petugas,
          'telp' => $dt_sw->telp,
          'username' => $dt_sw->username,
          'password' => $dt_sw->password,
          'level' => $dt_sw->level,
          'created_at' => $dt_sw->created_at,
          'updated_at' => $dt_sw->updated_at,
        );
      }
      return Response()->json(compact('count','film'));
    } else {
      return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
    }
    }

    public function update($id,Request $req)
    {
      if(Auth::user()->level=="admin"){
      $validator=Validator::make($req->all(),
      [
        'nama_petugas' => 'required',
        'telp' => 'required',
```

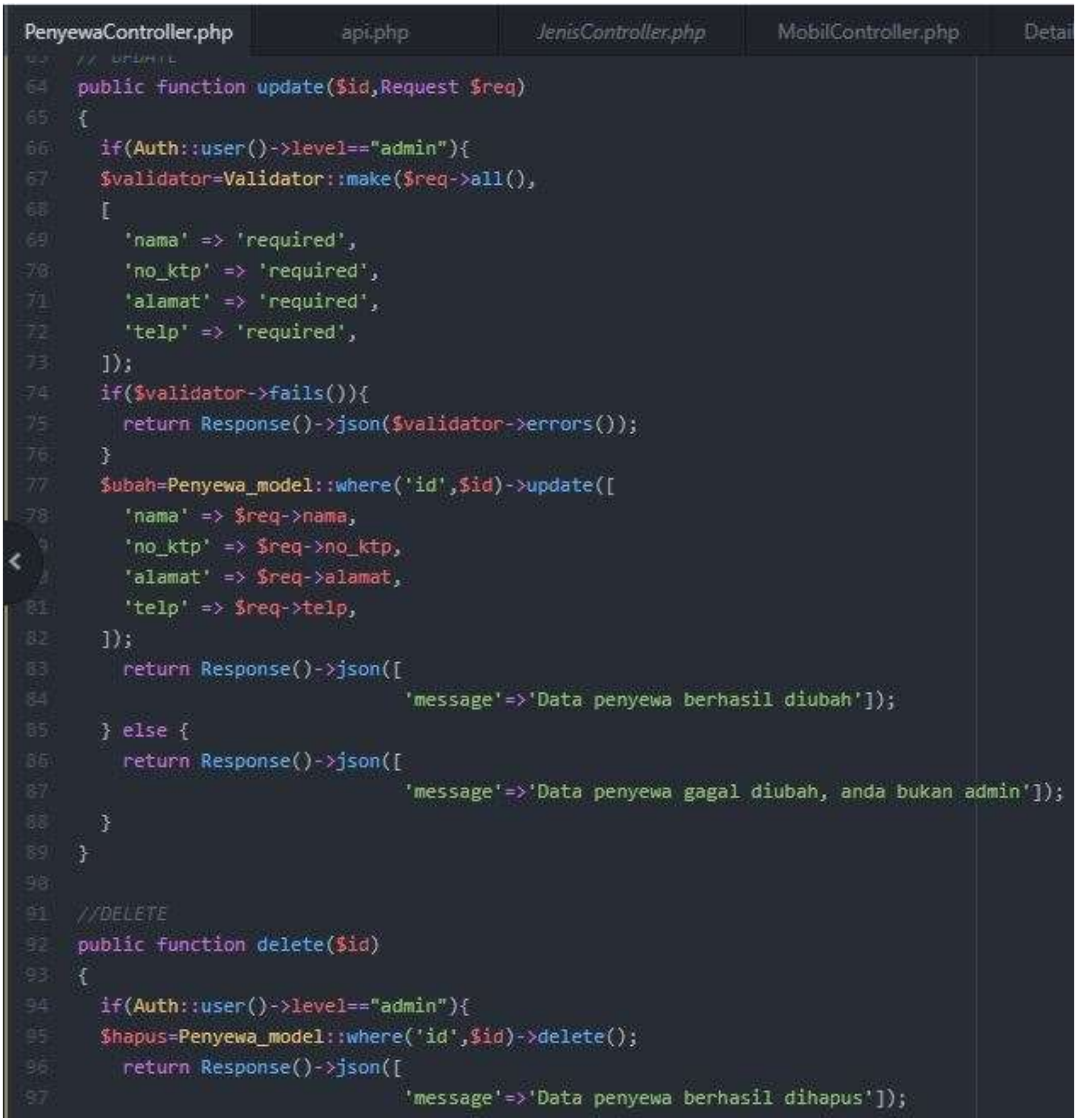
b. Penyewa



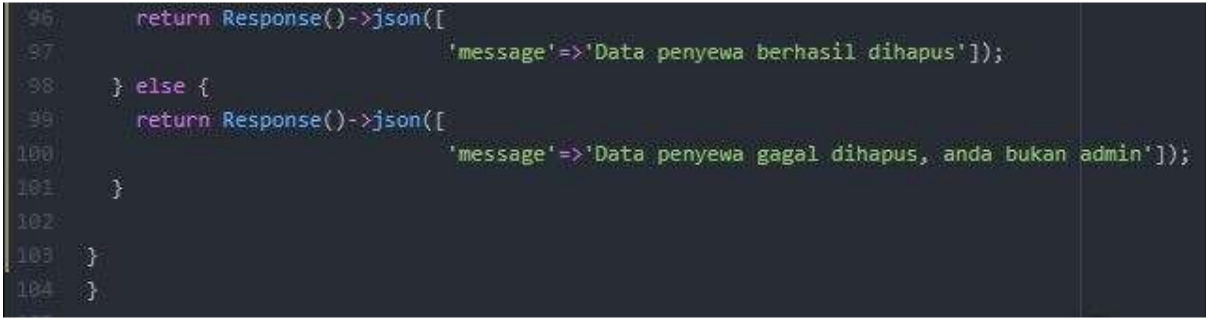
```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Penyewa_model;
use Illuminate\Support\Facades\Validator;
use Auth;

class PenyewaController extends Controller
{
  // CREATE
  public function store(Request $req)
  {
    if(Auth::user()->level=="admin"){
    $validator = Validator::make($req->all(),
    [
      'nama' => 'required',
      'no_ktp' => 'required',
      'alamat' => 'required',
      'telp' => 'required'
    ]);
    if($validator->fails()){
      return Response()->json($validator->errors());
    }
    $simpan = Penyewa_model::create([
      'nama' => $req->nama,
      'no_ktp' => $req->no_ktp,
      'alamat' => $req->alamat,
      'telp' => $req->telp,

    ]);
    return Response()->json(['status'=> 'Data penyewa berhasil dimasukkan']);
  } else {
```



```php
  } else {
      return Response()->json(['status'=> 'Data penyewa gagal dimasukkan, anda bukan admin']);
    }
  }

  //READ
  public function tampil(){
    if(Auth::user()->level=="admin"){
  $datas = Penyewa_model::get();
  $count = $datas->count();
  $penyewa = array();
  $status = 1;
  foreach ($datas as $dt_sw){
    $penyewa[] = array(
      'id' => $dt_sw->id,
      'nama' => $dt_sw->nama,
      'no_ktp' => $dt_sw->no_ktp,
      'alamat' => $dt_sw->alamat,
      'telp' => $dt_sw->telp,
      'created_at' => $dt_sw->created_at,
      'updated_at' => $dt_sw->updated_at,
    );
  }
  return Response()->json(compact('count','penyewa'));
} else {
  return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
}
}

// UPDATE
public function update($id,Request $req)
{
  if(Auth::user()->level=="admin"){
  $validator=Validator::make($req->all(),
```

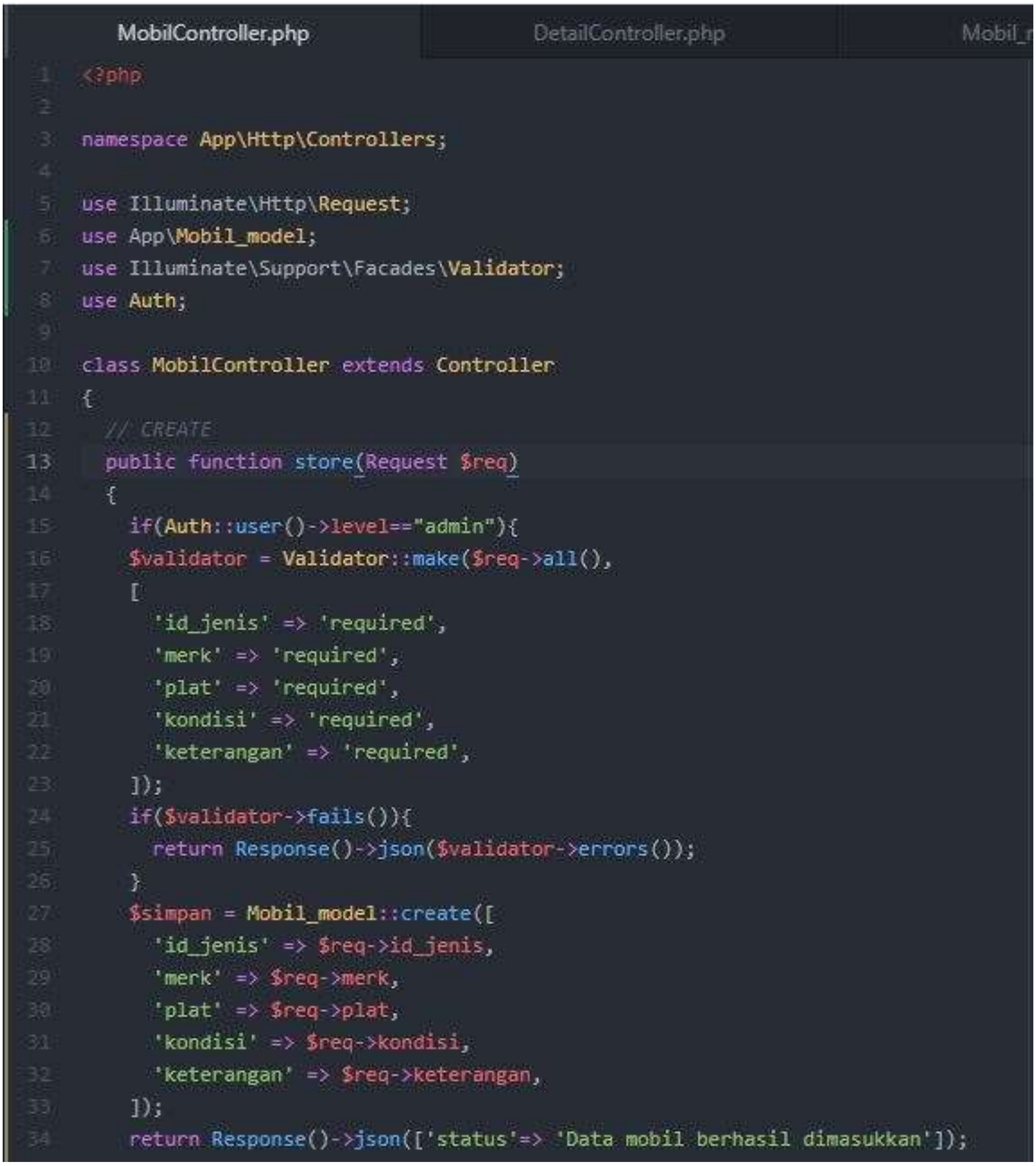


```
// UPDATE
public function update($id,Request $req)
{
  if(Auth::user()->level=="admin"){
  $validator=Validator::make($req->all(),
  [
    'nama' => 'required',
    'no_ktp' => 'required',
    'alamat' => 'required',
    'telp' => 'required',
  ]);
  if($validator->fails()){
    return Response()->json($validator->errors());
  }
  $ubah=Penyewa_model::where('id',$id)->update([
    'nama' => $req->nama,
    'no_ktp' => $req->no_ktp,
    'alamat' => $req->alamat,
    'telp' => $req->telp,
  ]);
    return Response()->json([
                             'message'=>'Data penyewa berhasil diubah']);
  } else {
    return Response()->json([
                             'message'=>'Data penyewa gagal diubah, anda bukan admin']);
  }
}

//DELETE
public function delete($id)
{
  if(Auth::user()->level=="admin"){
  $hapus=Penyewa_model::where('id',$id)->delete();
    return Response()->json([
                             'message'=>'Data penyewa berhasil dihapus']);
```

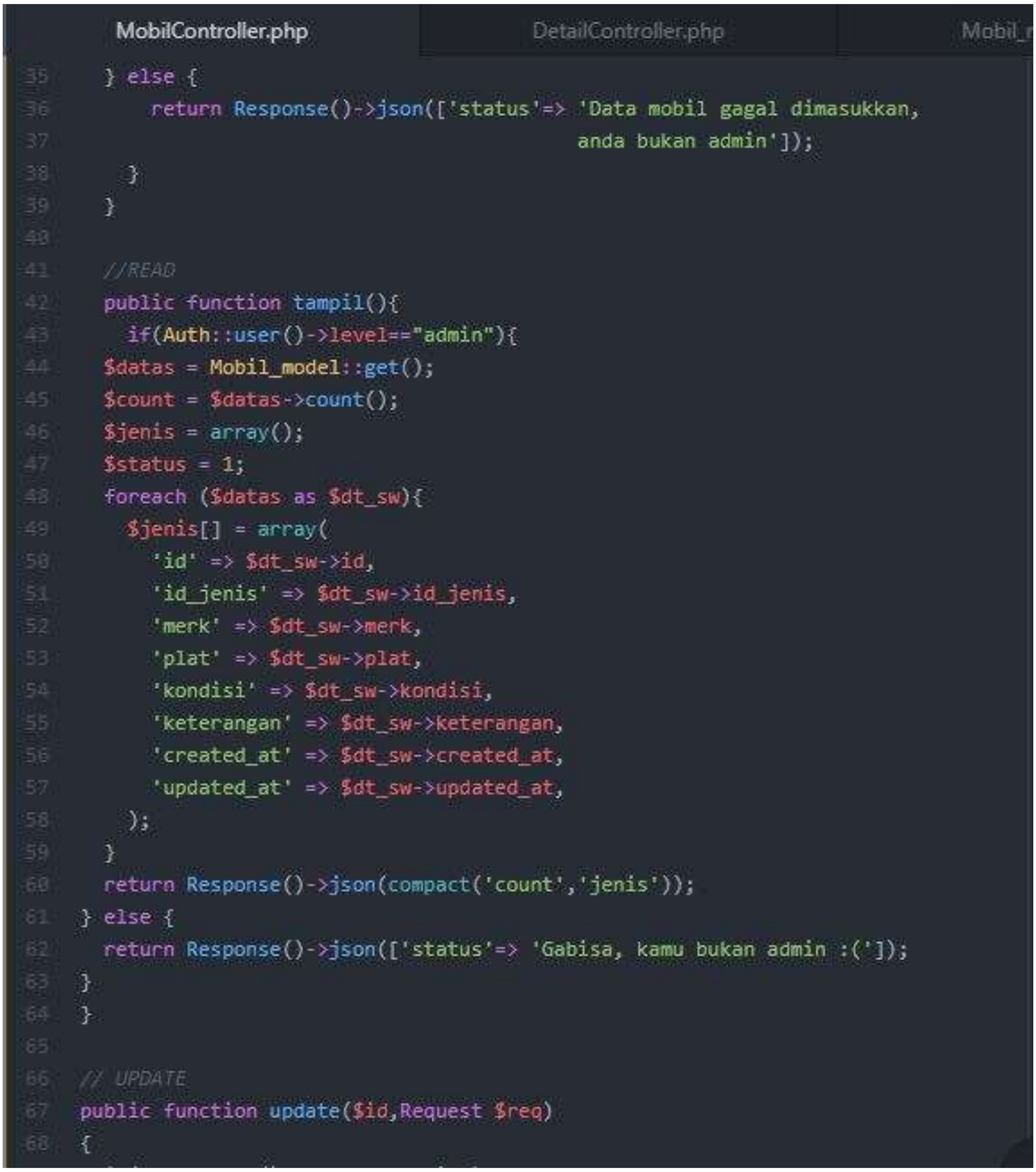


```
    return Response()->json([
                             'message'=>'Data penyewa berhasil dihapus']);
  } else {
    return Response()->json([
                             'message'=>'Data penyewa gagal dihapus, anda bukan admin']);
  }

}
}
```

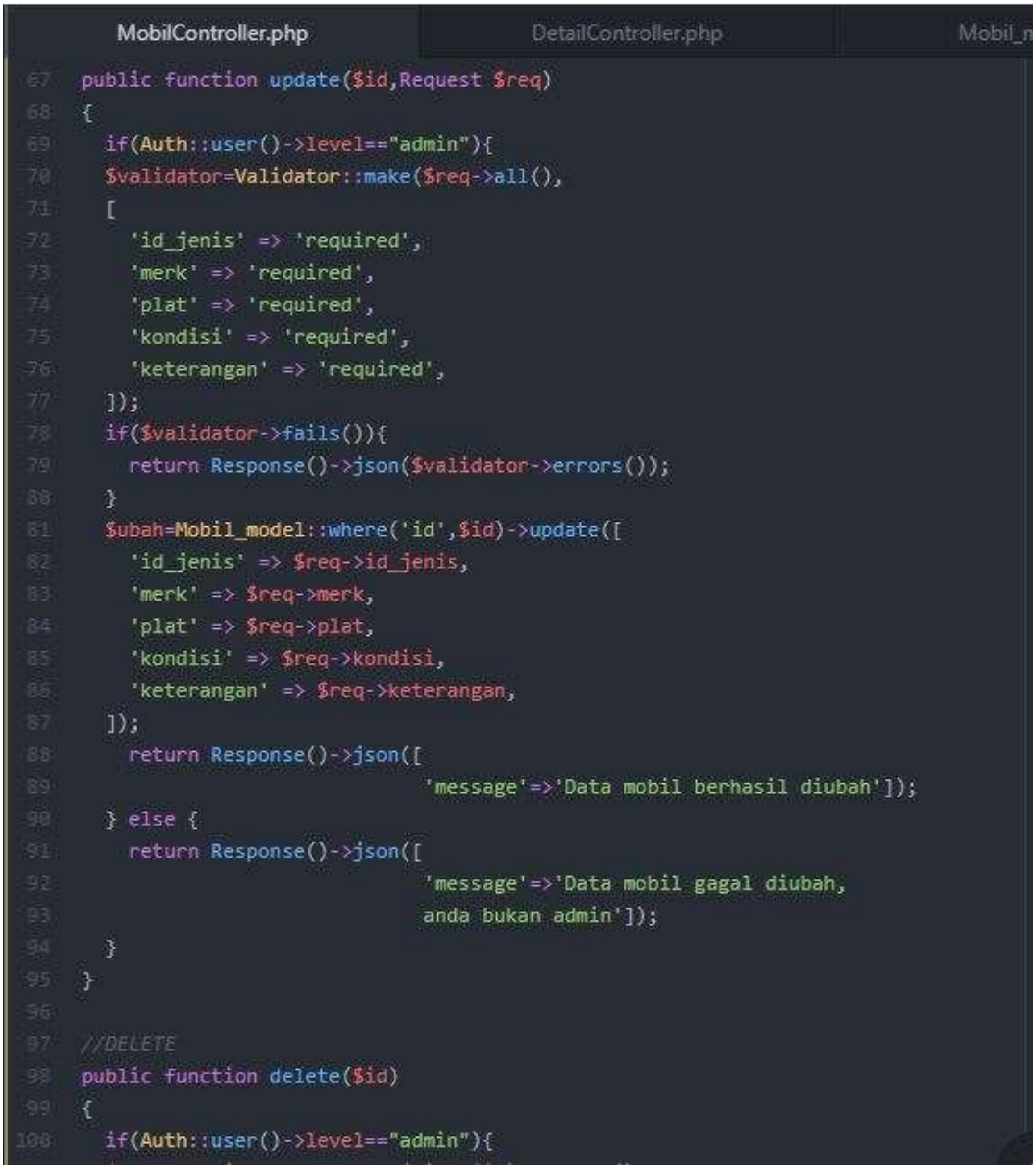
c. Mobil



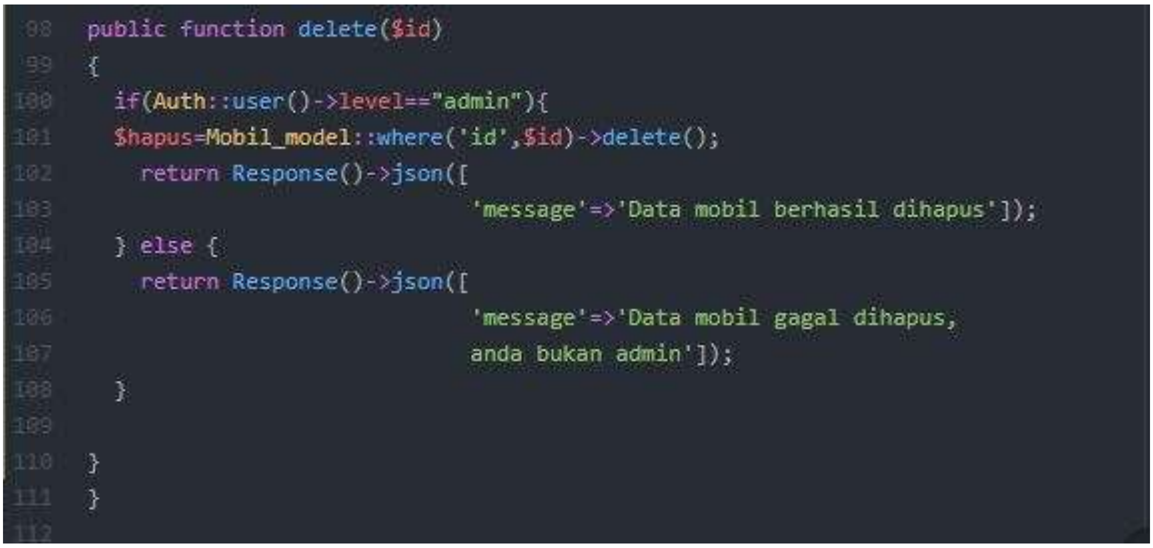
```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Mobil_model;
use Illuminate\Support\Facades\Validator;
use Auth;

class MobilController extends Controller
{
  // CREATE
  public function store(Request $req)
  {
    if(Auth::user()->level=="admin"){
    $validator = Validator::make($req->all(),
    [
      'id_jenis' => 'required',
      'merk' => 'required',
      'plat' => 'required',
      'kondisi' => 'required',
      'keterangan' => 'required',
    ]);
    if($validator->fails()){
      return Response()->json($validator->errors());
    }
    $simpan = Mobil_model::create([
      'id_jenis' => $req->id_jenis,
      'merk' => $req->merk,
      'plat' => $req->plat,
      'kondisi' => $req->kondisi,
      'keterangan' => $req->keterangan,
    ]);
    return Response()->json(['status'=> 'Data mobil berhasil dimasukkan']);
```



```php
  } else {
      return Response()->json(['status'=> 'Data mobil gagal dimasukkan,
                                          anda bukan admin']);
    }
  }

  //READ
  public function tampil(){
    if(Auth::user()->level=="admin"){
  $datas = Mobil_model::get();
  $count = $datas->count();
  $jenis = array();
  $status = 1;
  foreach ($datas as $dt_sw){
    $jenis[] = array(
      'id' => $dt_sw->id,
      'id_jenis' => $dt_sw->id_jenis,
      'merk' => $dt_sw->merk,
      'plat' => $dt_sw->plat,
      'kondisi' => $dt_sw->kondisi,
      'keterangan' => $dt_sw->keterangan,
      'created_at' => $dt_sw->created_at,
      'updated_at' => $dt_sw->updated_at,
    );
  }
  return Response()->json(compact('count','jenis'));
} else {
  return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
}
}

// UPDATE
public function update($id,Request $req)
{
```

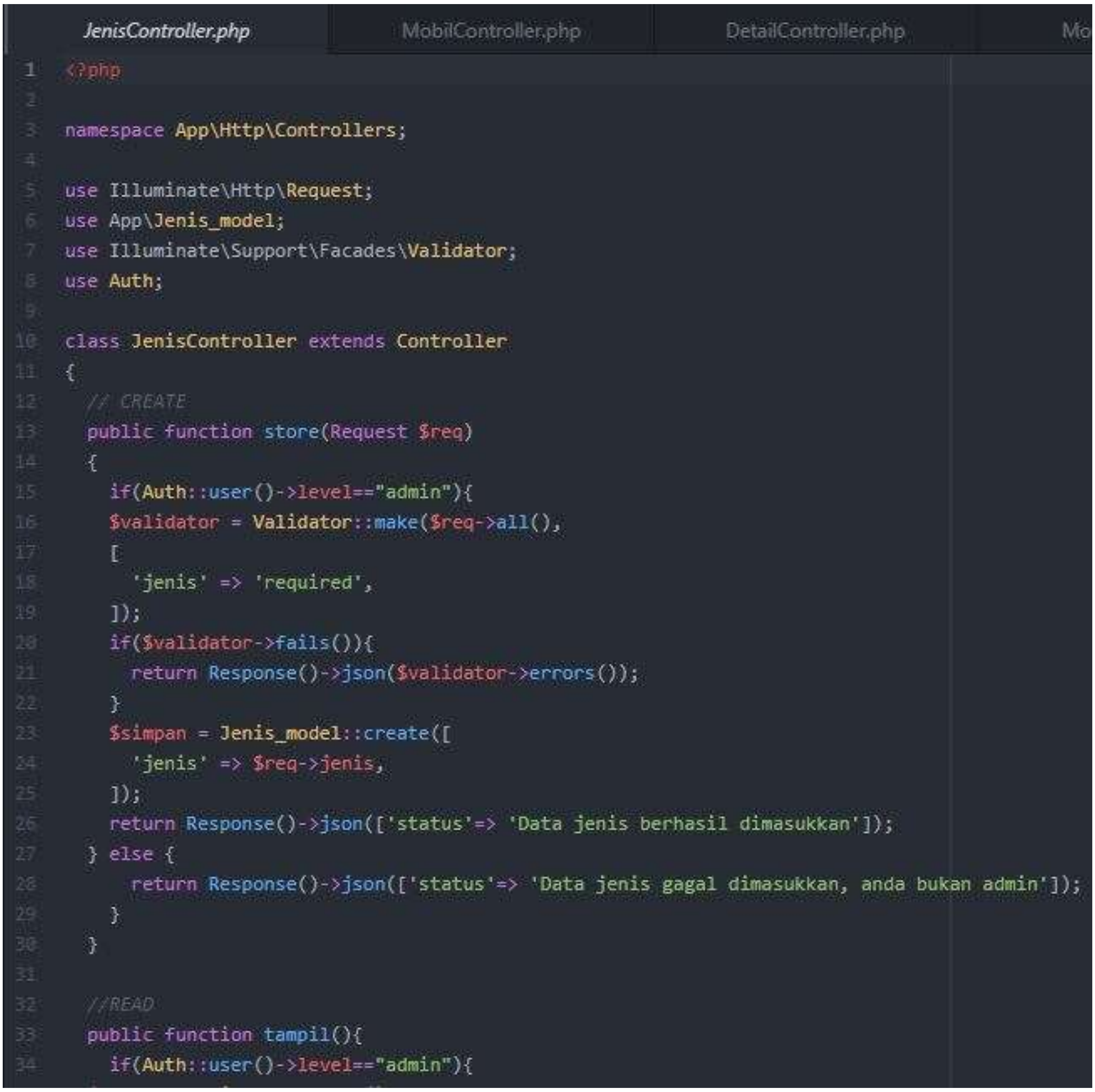


```php
public function update($id,Request $req)
{
  if(Auth::user()->level=="admin"){
  $validator=Validator::make($req->all(),
  [
    'id_jenis' => 'required',
    'merk' => 'required',
    'plat' => 'required',
    'kondisi' => 'required',
    'keterangan' => 'required',
  ]);
  if($validator->fails()){
    return Response()->json($validator->errors());
  }
  $ubah=Mobil_model::where('id',$id)->update([
    'id_jenis' => $req->id_jenis,
    'merk' => $req->merk,
    'plat' => $req->plat,
    'kondisi' => $req->kondisi,
    'keterangan' => $req->keterangan,
  ]);
    return Response()->json([
                             'message'=>'Data mobil berhasil diubah']);
  } else {
    return Response()->json([
                             'message'=>'Data mobil gagal diubah,
                             anda bukan admin']);
  }
}

//DELETE
public function delete($id)
{
  if(Auth::user()->level=="admin"){
```

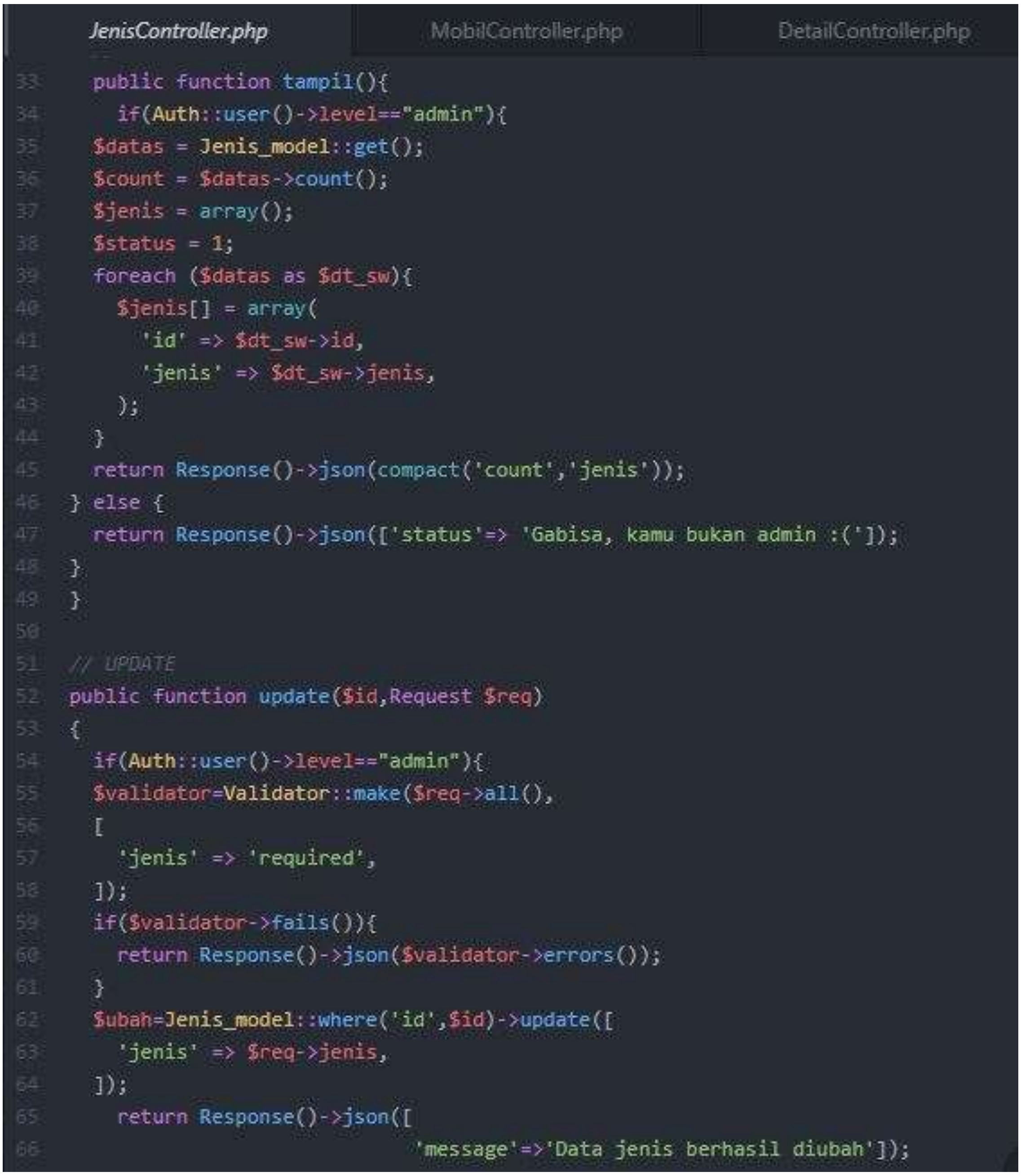


```php
public function delete($id)
{
  if(Auth::user()->level=="admin"){
  $hapus=Mobil_model::where('id',$id)->delete();
    return Response()->json([
                             'message'=>'Data mobil berhasil dihapus']);
  } else {
    return Response()->json([
                             'message'=>'Data mobil gagal dihapus,
                             anda bukan admin']);
  }

}
}
```

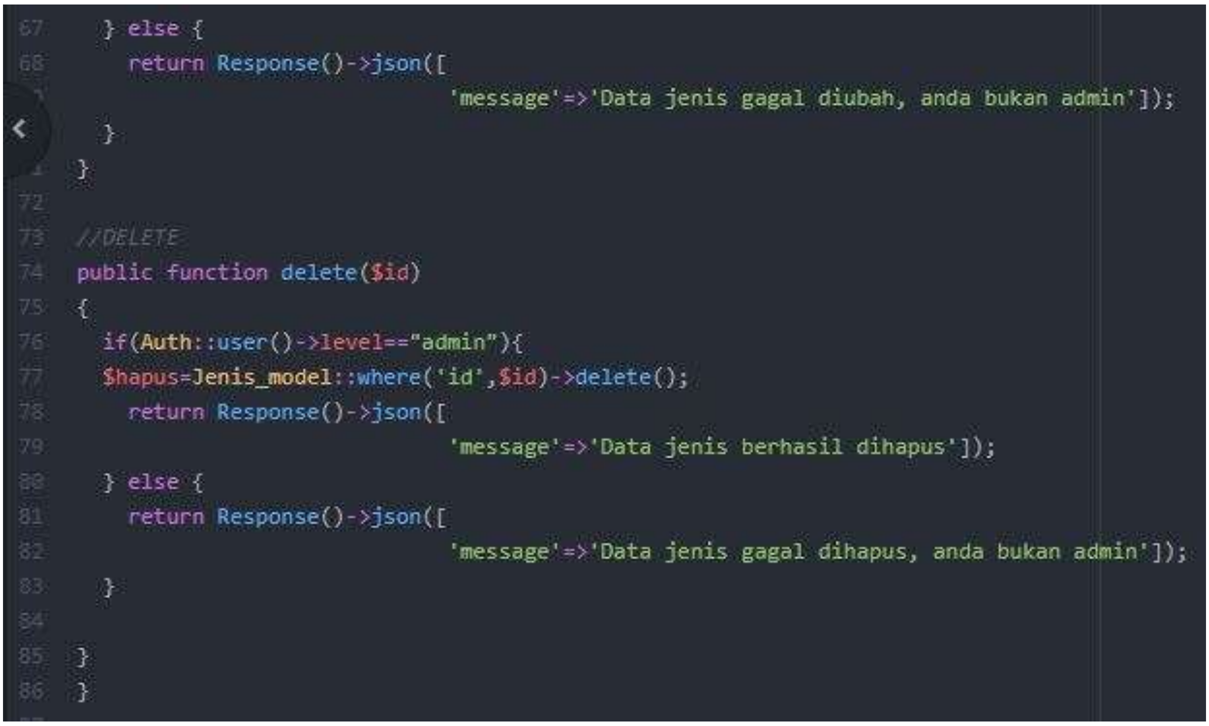
d. Jenis mobil



```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Jenis_model;
use Illuminate\Support\Facades\Validator;
use Auth;

class JenisController extends Controller
{
  // CREATE
  public function store(Request $req)
  {
    if(Auth::user()->level=="admin"){
    $validator = Validator::make($req->all(),
    [
      'jenis' => 'required',
    ]);
    if($validator->fails()){
      return Response()->json($validator->errors());
    }
    $simpan = Jenis_model::create([
      'jenis' => $req->jenis,
    ]);
    return Response()->json(['status'=> 'Data jenis berhasil dimasukkan']);
  } else {
      return Response()->json(['status'=> 'Data jenis gagal dimasukkan, anda bukan admin']);
    }
  }

  //READ
  public function tampil(){
    if(Auth::user()->level=="admin"){
```

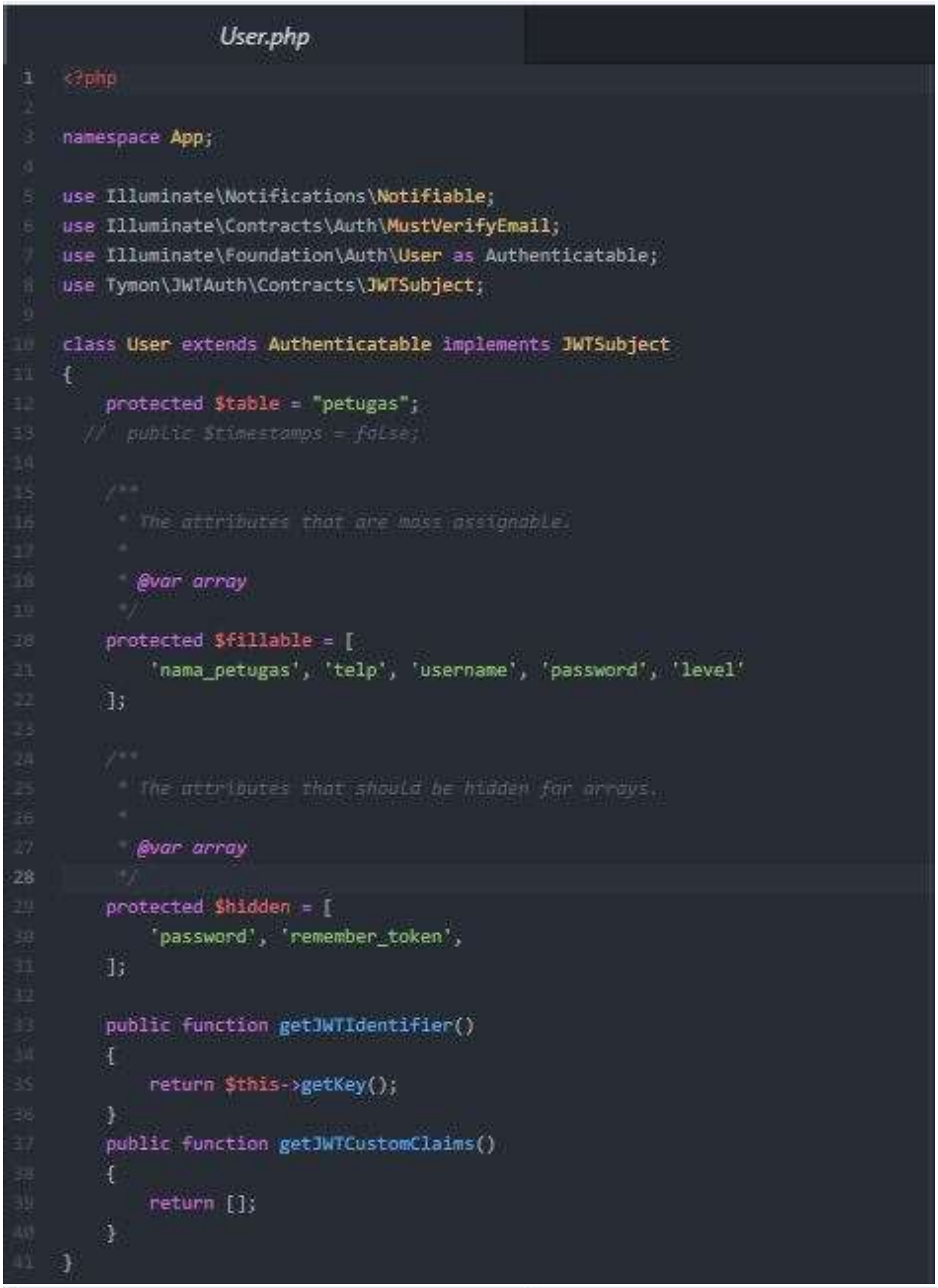


```php
  public function tampil(){
    if(Auth::user()->level=="admin"){
  $datas = Jenis_model::get();
  $count = $datas->count();
  $jenis = array();
  $status = 1;
  foreach ($datas as $dt_sw){
    $jenis[] = array(
      'id' => $dt_sw->id,
      'jenis' => $dt_sw->jenis,
    );
  }
  return Response()->json(compact('count','jenis'));
} else {
  return Response()->json(['status'=> 'Gabisa, kamu bukan admin :(']);
}
}

// UPDATE
public function update($id,Request $req)
{
  if(Auth::user()->level=="admin"){
  $validator=Validator::make($req->all(),
  [
    'jenis' => 'required',
  ]);
  if($validator->fails()){
    return Response()->json($validator->errors());
  }
  $ubah=Jenis_model::where('id',$id)->update([
    'jenis' => $req->jenis,
  ]);
    return Response()->json([
                             'message'=>'Data jenis berhasil diubah']);
```

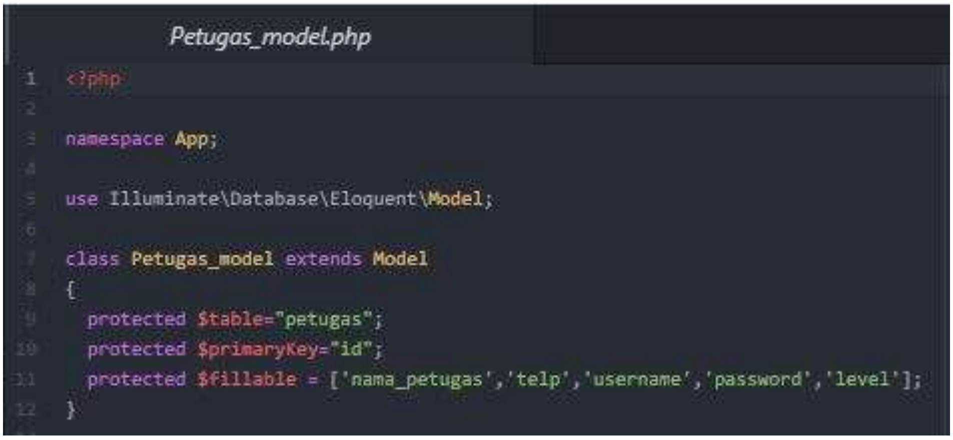


```
    } else {
      return Response()->json([
                                   'message'=>'Data jenis gagal diubah, anda bukan admin']);
    }
  }

  //DELETE
  public function delete($id)
  {
    if(Auth::user()->level=="admin"){
    $hapus=Jenis_model::where('id',$id)->delete();
      return Response()->json([
                                   'message'=>'Data jenis berhasil dihapus']);
    } else {
      return Response()->json([
                                   'message'=>'Data jenis gagal dihapus, anda bukan admin']);
    }

  }
}
```

2. Model
   a. Petugas

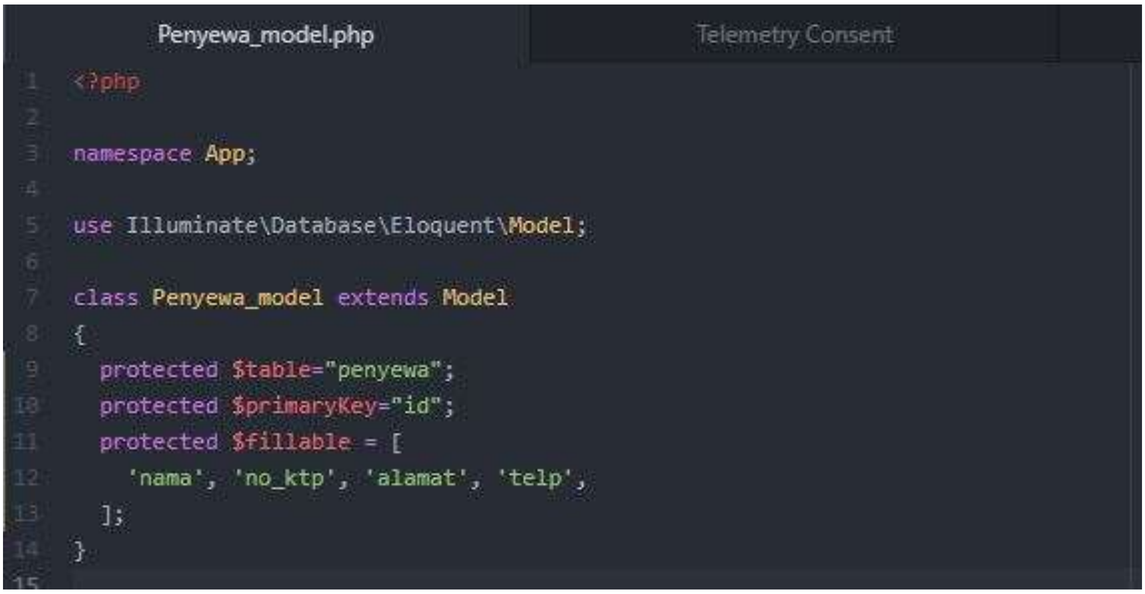


```php
<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Tymon\JWTAuth\Contracts\JWTSubject;

class User extends Authenticatable implements JWTSubject
{
    protected $table = "petugas";
  //  public $timestamps = false;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'nama_petugas', 'telp', 'username', 'password', 'level'
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    public function getJWTIdentifier()
    {
        return $this->getKey();
    }
    public function getJWTCustomClaims()
    {
        return [];
    }
}
```



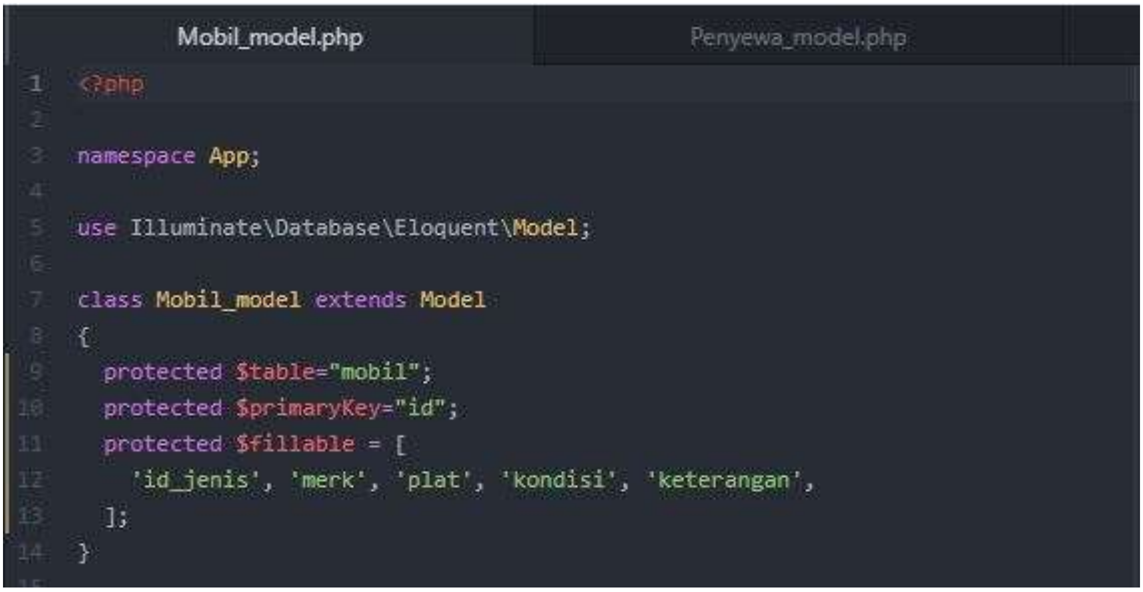
```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Petugas_model extends Model
{
  protected $table="petugas";
  protected $primaryKey="id";
  protected $fillable = ['nama_petugas','telp','username','password','level'];
}
```

b. Penyewa



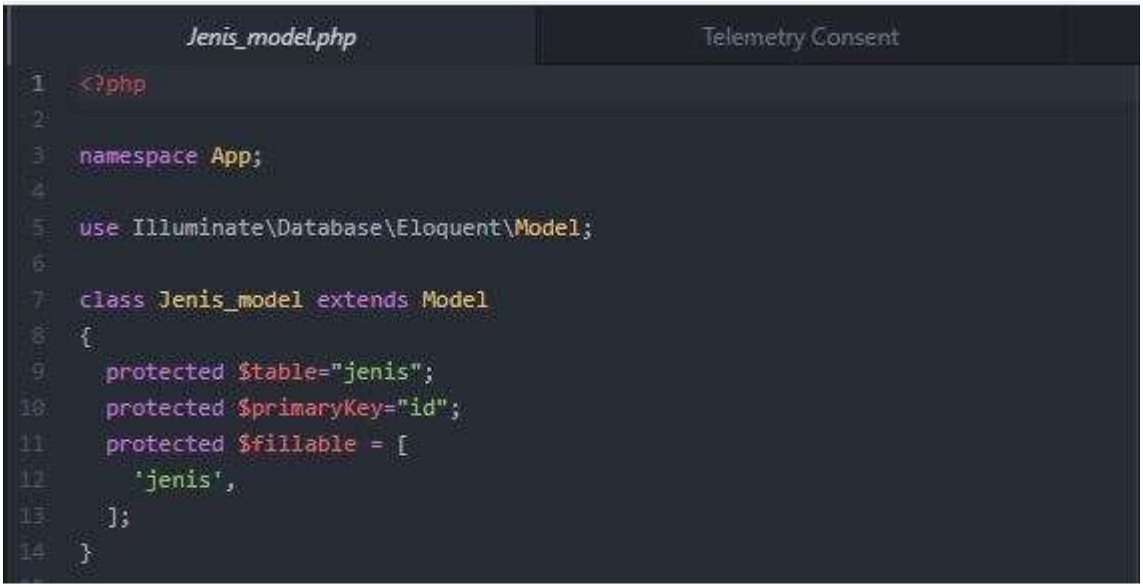
```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Penyewa_model extends Model
{
  protected $table="penyewa";
  protected $primaryKey="id";
  protected $fillable = [
    'nama', 'no_ktp', 'alamat', 'telp',
  ];
}
```

c. Mobil



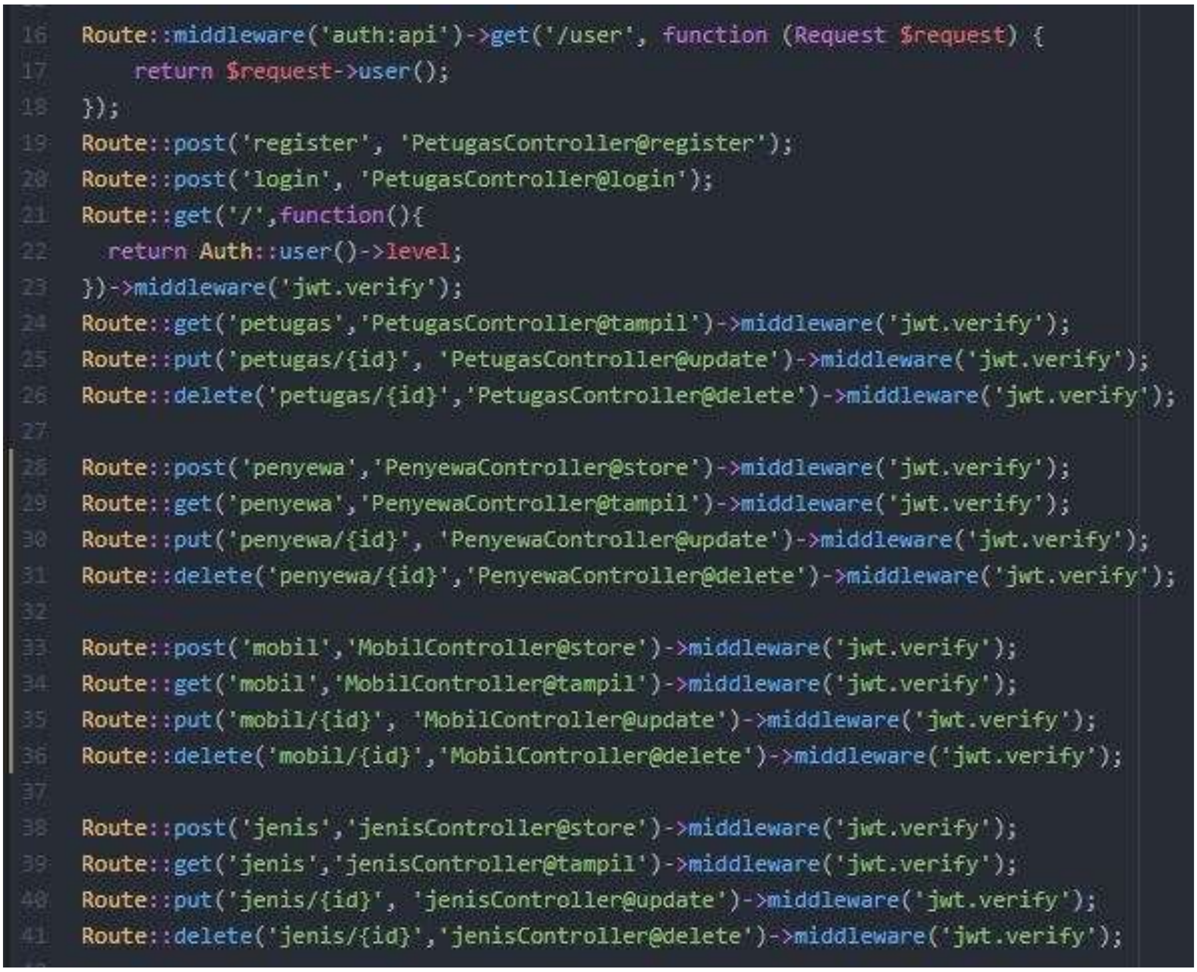
```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Mobil_model extends Model
{
  protected $table="mobil";
  protected $primaryKey="id";
  protected $fillable = [
    'id_jenis', 'merk', 'plat', 'kondisi', 'keterangan',
  ];
}
```

d. Jenis mobil



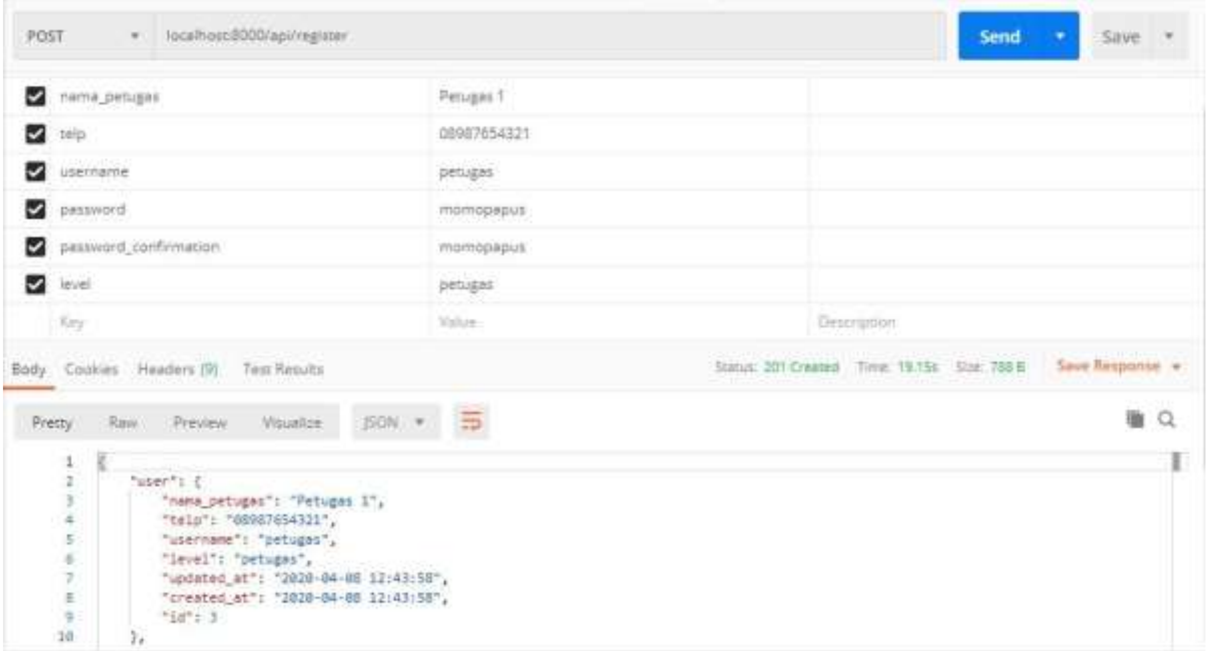
```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Jenis_model extends Model
{
  protected $table="jenis";
  protected $primaryKey="id";
  protected $fillable = [
    'jenis',
  ];
}
```

3. API



```
Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
Route::post('register', 'PetugasController@register');
Route::post('login', 'PetugasController@login');
Route::get('/',function(){
  return Auth::user()->level;
})->middleware('jwt.verify');
Route::get('petugas','PetugasController@tampil')->middleware('jwt.verify');
Route::put('petugas/{id}', 'PetugasController@update')->middleware('jwt.verify');
Route::delete('petugas/{id}','PetugasController@delete')->middleware('jwt.verify');

Route::post('penyewa','PenyewaController@store')->middleware('jwt.verify');
Route::get('penyewa','PenyewaController@tampil')->middleware('jwt.verify');
Route::put('penyewa/{id}', 'PenyewaController@update')->middleware('jwt.verify');
Route::delete('penyewa/{id}','PenyewaController@delete')->middleware('jwt.verify');

Route::post('mobil','MobilController@store')->middleware('jwt.verify');
Route::get('mobil','MobilController@tampil')->middleware('jwt.verify');
Route::put('mobil/{id}', 'MobilController@update')->middleware('jwt.verify');
Route::delete('mobil/{id}','MobilController@delete')->middleware('jwt.verify');

Route::post('jenis','jenisController@store')->middleware('jwt.verify');
Route::get('jenis','jenisController@tampil')->middleware('jwt.verify');
Route::put('jenis/{id}', 'jenisController@update')->middleware('jwt.verify');
Route::delete('jenis/{id}','jenisController@delete')->middleware('jwt.verify');
```

4. Hasil
   a. Petugas
      1. Register



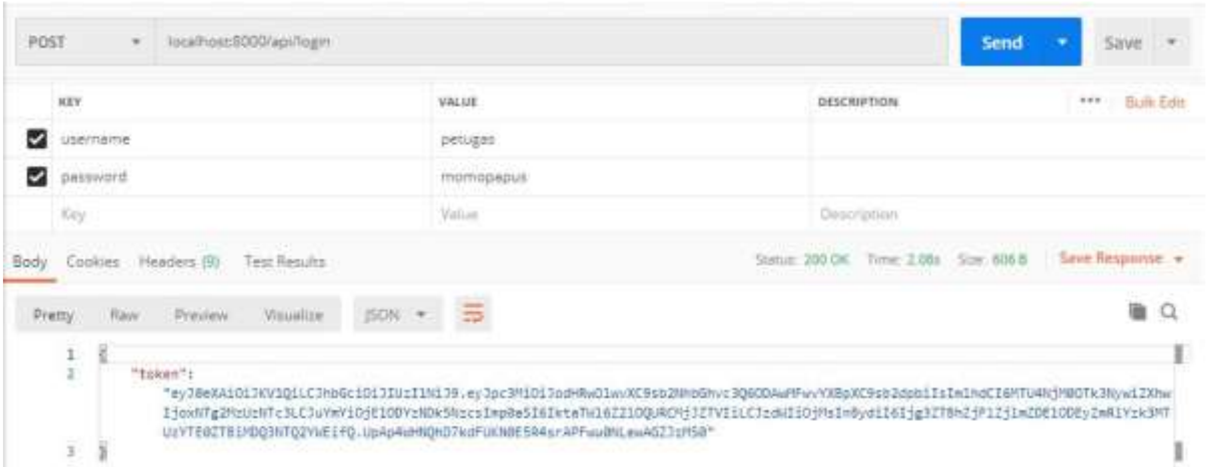
localhost:8000/api/register

☑ nama_petugas	Petugas 1	
☑ telp	08987654321	
☑ username	petugas	
☑ password	momopapus	
☑ password_confirmation	momopapus	
☑ level	petugas	
Key	Value	Description

Status: 201 Created Time: 19.15s Size: 788 B

```
{
    "user": {
        "nama_petugas": "Petugas 1",
        "telp": "08987654321",
        "username": "petugas",
        "level": "petugas",
        "updated_at": "2020-04-08 12:43:58",
        "created_at": "2020-04-08 12:43:58",
        "id": 3
    },
```

      2. Login



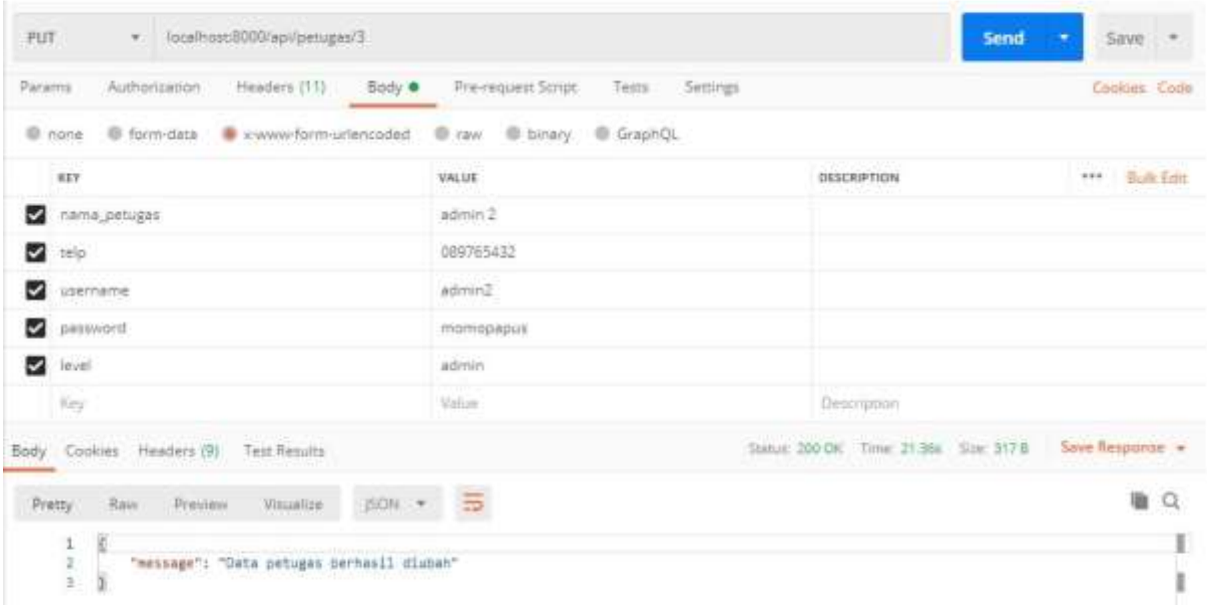
POST localhost:8000/api/login

KEY	VALUE	DESCRIPTION
☑ username	petugas	
☑ password	momopapus	
Key	Value	Description

Status: 200 OK Time: 2.08s Size: 606 B

```
{
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3Q6ODAwMFwvYXBpXC9sb2dpbiIsImlhdCI6MTU4NjM0OTk3NywiZXhw
        IjoxNTg2MzUzNTc3LCJuYmYiOjE1ODYzNDk5NzcsImp0aSI6IkteTWl6ZzlOQURCMjJZTVIiLCJzdWIiOjMsInBydiI6Ijg3ZTBhZjFlZjlmZDE1ODEyZmRlYzk3MT
        UzYTE0ZTBiMDQ3NTQ2YWEifQ.UpAp4wHNQhD7kdFUKN8E5R4srAPFwuBNLewAGZJsM50"
}
```

3. Update



PUT localhost:8000/api/petugas/3

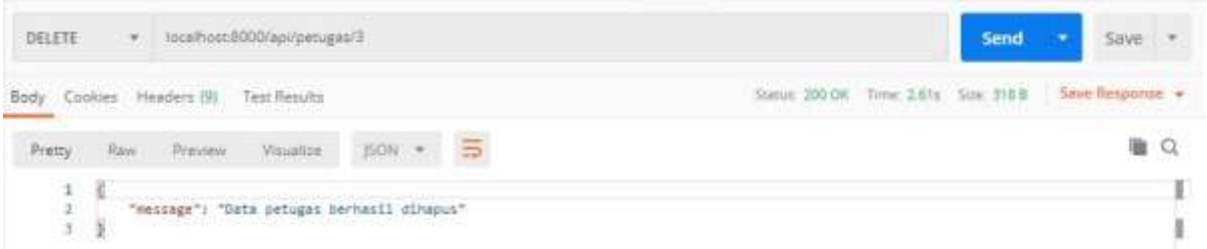
Body: ● x-www-form-urlencoded

KEY	VALUE	DESCRIPTION
☑ nama_petugas	admin 2	
☑ telp	089765432	
☑ username	admin2	
☑ password	momopapus	
☑ level	admin	
Key	Value	Description

Status: 200 OK Time: 21.36s Size: 317 B

```
{
    "message": "Data petugas berhasil diubah"
}
```

4. Delete



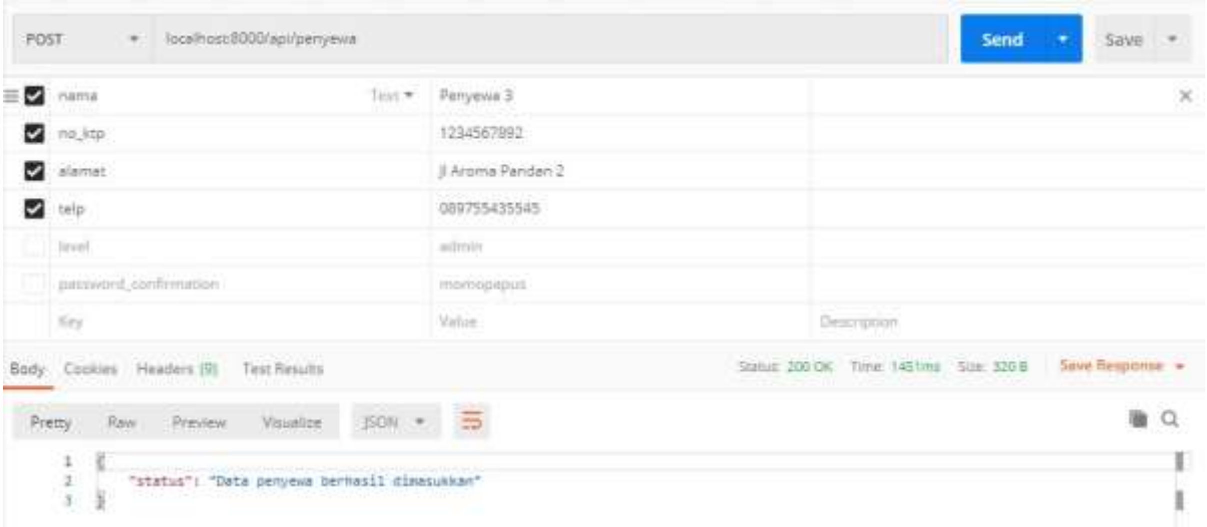
DELETE localhost:8000/api/petugas/3

Status: 200 OK Time: 2.61s Size: 318 B

```
{
    "message": "Data petugas berhasil dihapus"
}
```

b. Penyewa

1. Create



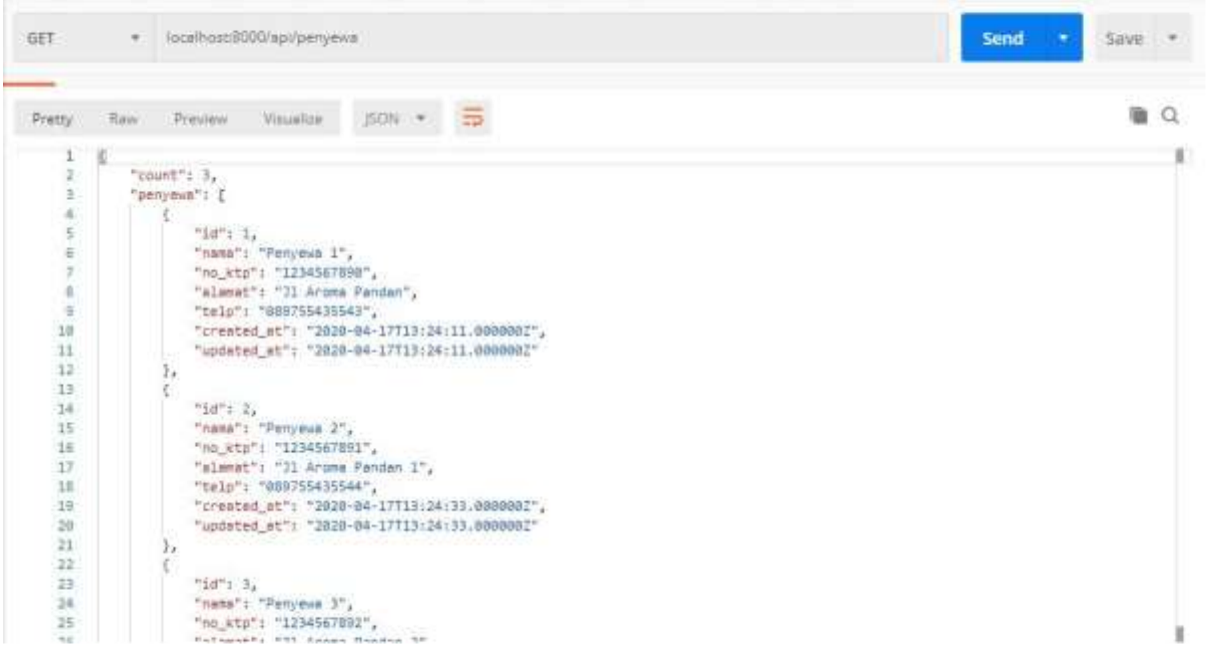
POST localhost:8000/api/penyewa

☑	nama	Text	Penyewa 3
☑	no_ktp		1234567892
☑	alamat		Jl Aroma Pandan 2
☑	telp		089755435545
□	level		admin
□	password_confirmation		momopepus
	Key		Value

Status: 200 OK Time: 1451ms Size: 320 B

```
{
    "status": "Data penyewa berhasil dimasukkan"
}
```

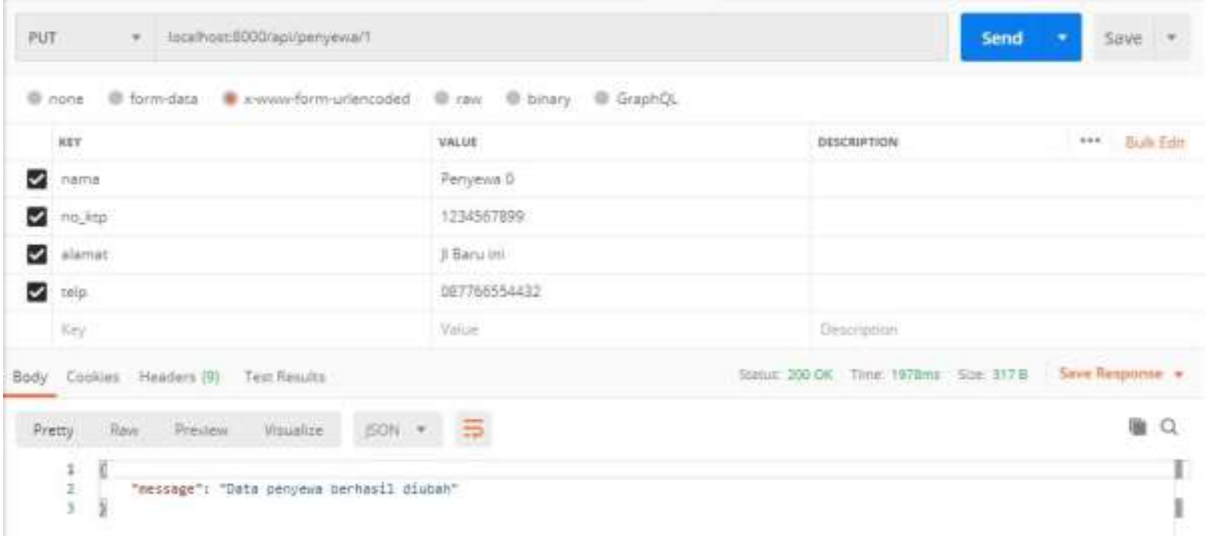
2. Read



GET localhost:8000/api/penyewa

```
{
    "count": 3,
    "penyewa": [
        {
            "id": 1,
            "nama": "Penyewa 1",
            "no_ktp": "1234567890",
            "alamat": "Jl Aroma Pandan",
            "telp": "089755435543",
            "created_at": "2020-04-17T13:24:11.000000Z",
            "updated_at": "2020-04-17T13:24:11.000000Z"
        },
        {
            "id": 2,
            "nama": "Penyewa 2",
            "no_ktp": "1234567891",
            "alamat": "Jl Aroma Pandan 1",
            "telp": "089755435544",
            "created_at": "2020-04-17T13:24:33.000000Z",
            "updated_at": "2020-04-17T13:24:33.000000Z"
        },
        {
            "id": 3,
            "nama": "Penyewa 3",
            "no_ktp": "1234567892",
```

3. Update



PUT localhost:8000/api/penyewa/1

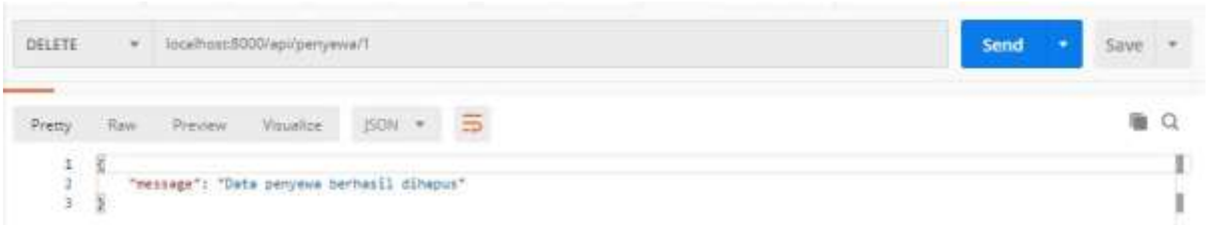
○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	nama	Penyewa 0	
☑	no_ktp	1234567899	
☑	alamat	Jl Baru ini	
☑	telp	087766554432	
	Key	Value	Description

Status: 200 OK Time: 1978ms Size: 317 B

```
{
    "message": "Data penyewa berhasil diubah"
}
```

4. Delete

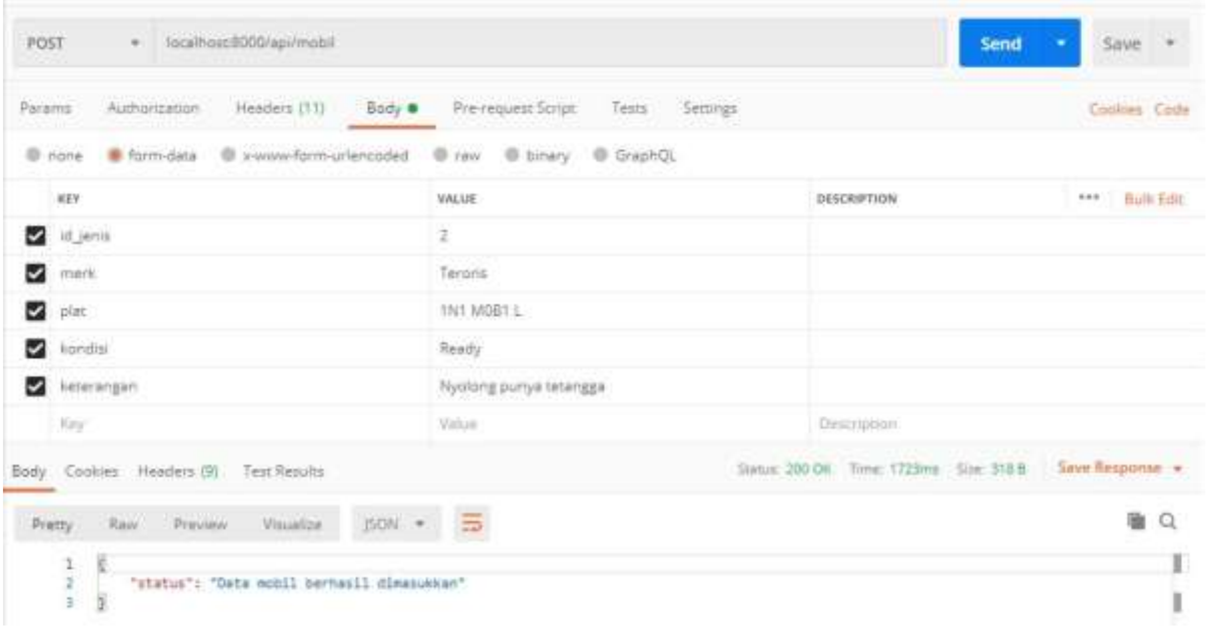


DELETE localhost:8000/api/penyewa/1

```
{
    "message": "Data penyewa berhasil dihapus"
}
```

c. Mobil

1. Create



POST localhost:8000/api/mobil

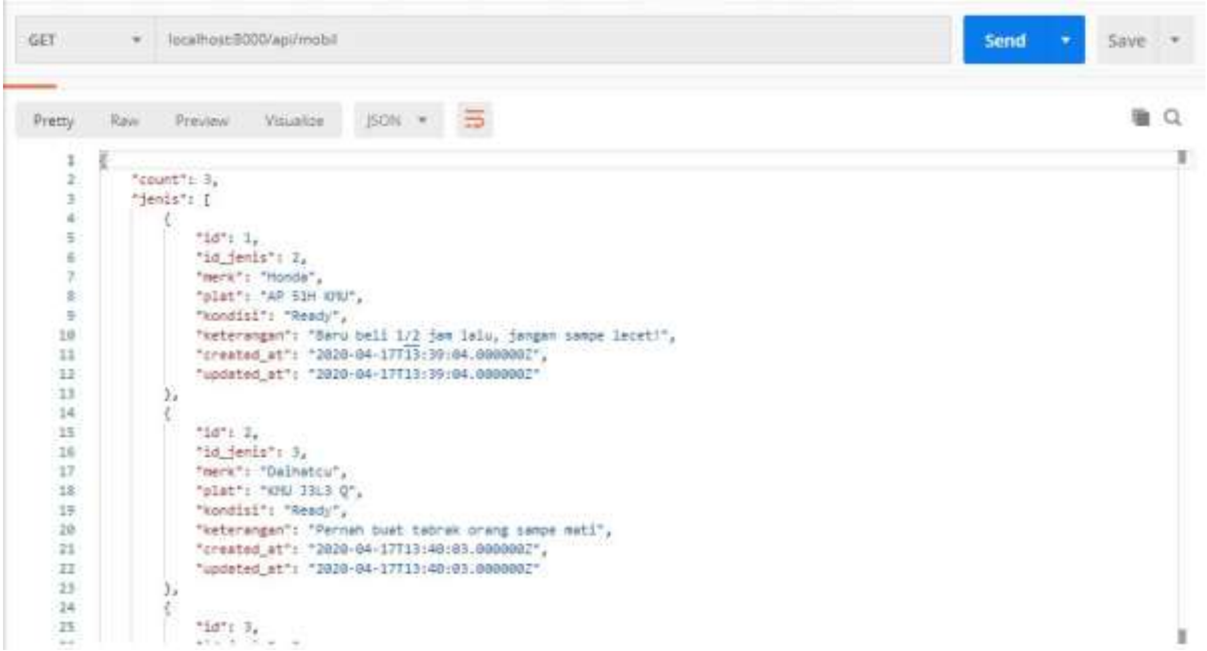
○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	id_jenis	2	
☑	merk	Teroris	
☑	plat	1N1 M0B1 L	
☑	kondisi	Ready	
☑	keterangan	Nyolong punya tetangga	
	Key	Value	Description

Status: 200 OK Time: 1723ms Size: 318 B

```
{
    "status": "Data mobil berhasil dimasukkan"
}
```

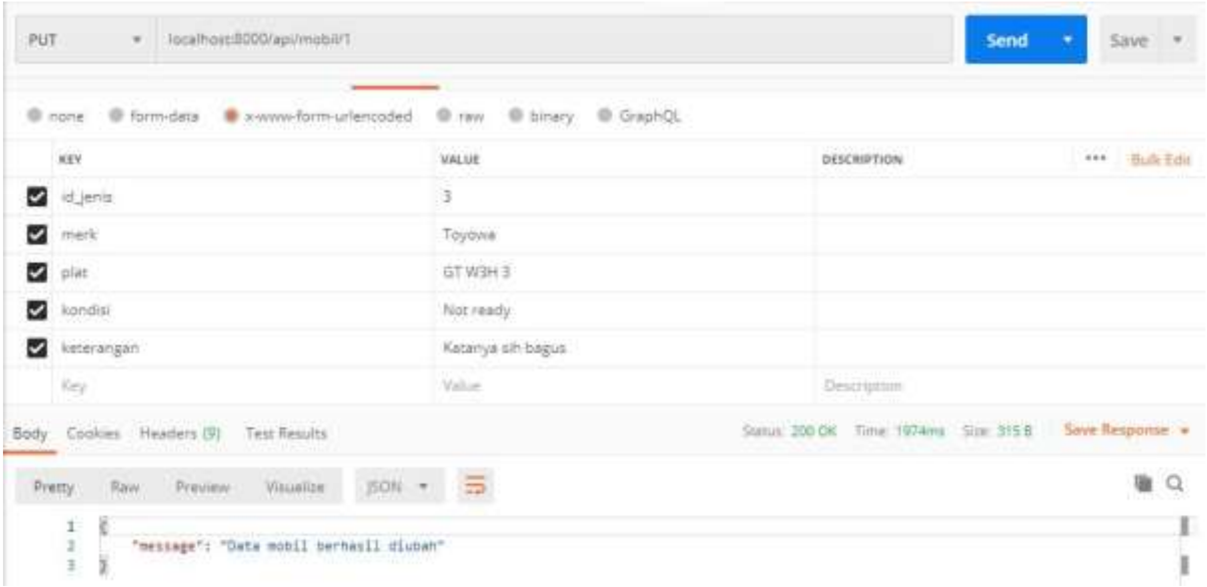
2. Read



GET localhost:8000/api/mobil

```
{
    "count": 3,
    "jenis": [
        {
            "id": 1,
            "id_jenis": 2,
            "merk": "Honda",
            "plat": "AP 51H KMU",
            "kondisi": "Ready",
            "keterangan": "Baru beli 1/2 jam lalu, jangan sampe lecet!",
            "created_at": "2020-04-17T13:39:04.000000Z",
            "updated_at": "2020-04-17T13:39:04.000000Z"
        },
        {
            "id": 2,
            "id_jenis": 3,
            "merk": "Daihatcu",
            "plat": "KMU J3L3 Q",
            "kondisi": "Ready",
            "keterangan": "Pernah buat tabrak orang sampe mati",
            "created_at": "2020-04-17T13:40:03.000000Z",
            "updated_at": "2020-04-17T13:40:03.000000Z"
        },
        {
            "id": 3,
```

3. Update



PUT localhost:8000/api/mobil/1

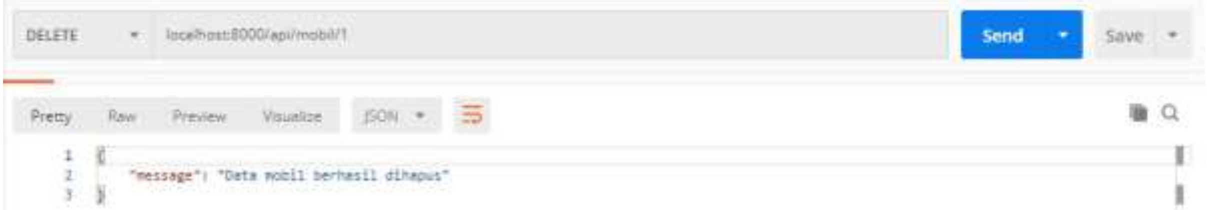
○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	id_jenis	3	
☑	merk	Toyowa	
☑	plat	GT W3H 3	
☑	kondisi	Not ready	
☑	keterangan	Katanya sih bagus	
	Key	Value	Description

Status: 200 OK Time: 1974ms Size: 315 B

```
{
    "message": "Data mobil berhasil diubah"
}
```

4. Delete

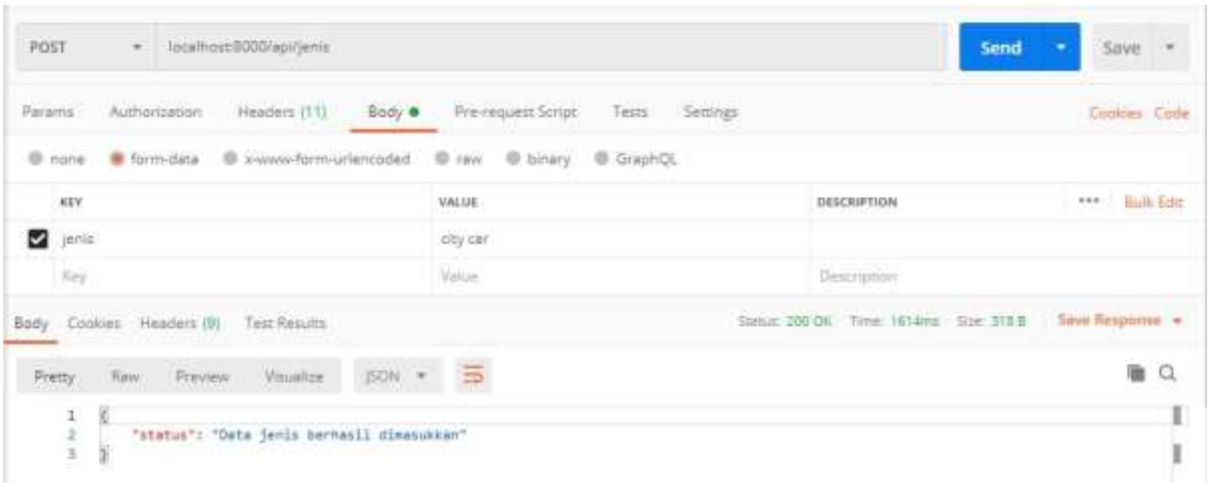


DELETE localhost:8000/api/mobil/1

```
{
    "message": "Data mobil berhasil dihapus"
}
```

d. Jenis

1. Create



POST localhost:8000/api/jenis

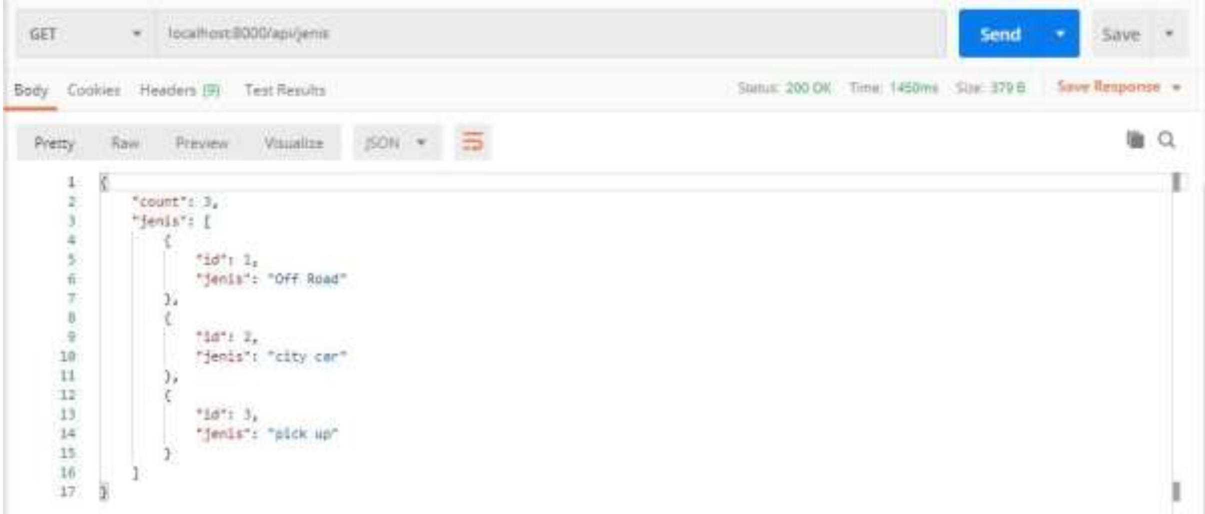
○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	jenis	city car	
	Key	Value	Description

Status: 200 OK Time: 1614ms Size: 318 B

```
{
    "status": "Data jenis berhasil dimasukkan"
}
```

2. Read

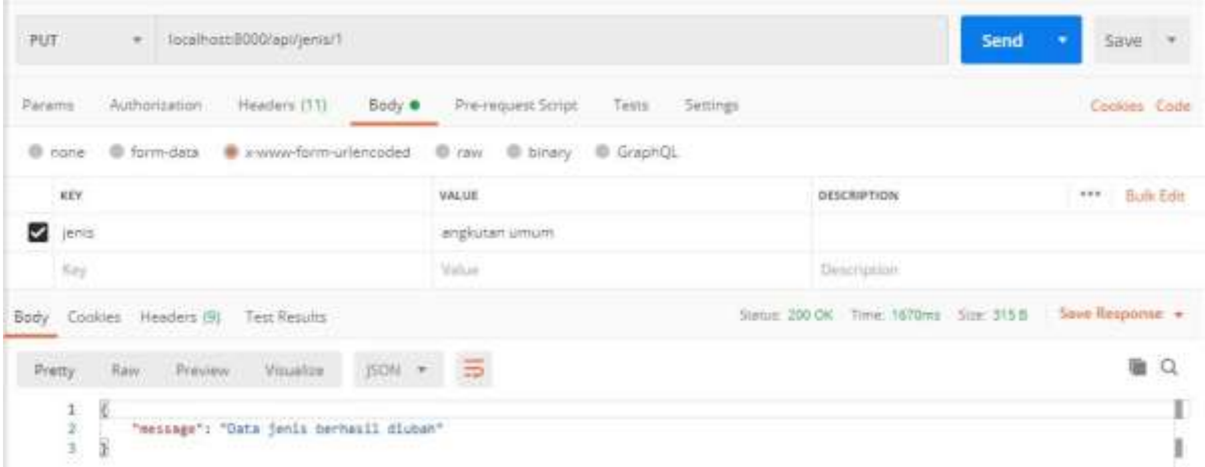


GET localhost:8000/api/jenis

Status: 200 OK Time: 1450ms Size: 379 B

```
{
    "count": 3,
    "jenis": [
        {
            "id": 1,
            "jenis": "Off Road"
        },
        {
            "id": 2,
            "jenis": "city car"
        },
        {
            "id": 3,
            "jenis": "pick up"
        }
    ]
}
```

3. Update



PUT localhost:8000/api/jenis/1

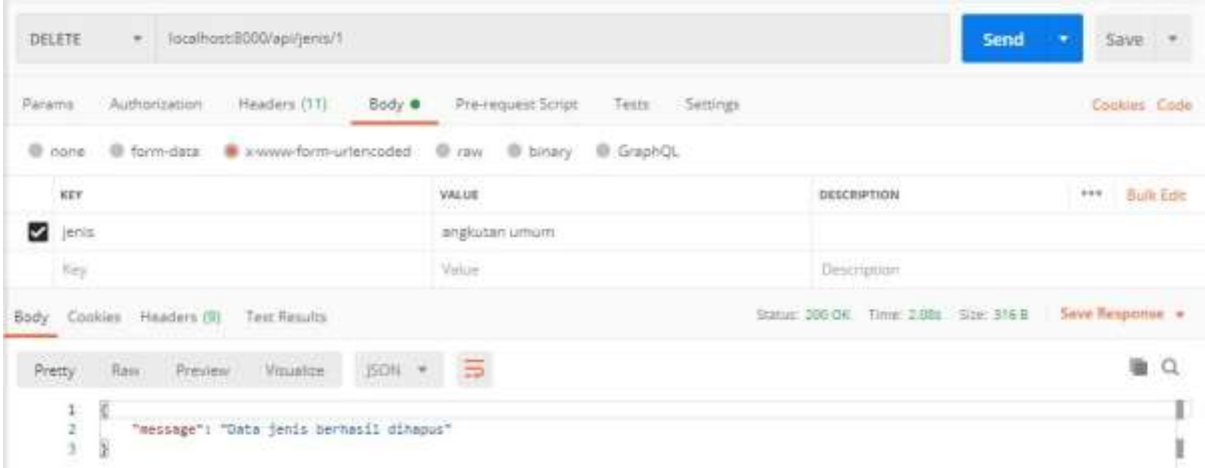
○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	jenis	angkutan umum	
	Key	Value	Description

Status: 200 OK Time: 1670ms Size: 315 B

```
{
    "message": "Data jenis berhasil diubah"
}
```

4. Delete



DELETE localhost:8000/api/jenis/1

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

	KEY	VALUE	DESCRIPTION
☑	jenis	angkutan umum	
	Key	Value	Description

Status: 200 OK Time: 2.08s Size: 316 B

```
{
    "message": "Data jenis berhasil dihapus"
}
```