Nama : Muhammad Bagas Ramadhan

Absen : 19

API

```
*/

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

Route::post('register', 'Petugas@register');
Route::post('login', 'Petugas@login');

Route::get("/","Petugas@tampil")->middleware('jwt.verify');

Route::get('user', 'Petugas@getAuthenticatedUser')->middleware('jwt.verify');

Route::post('/simpan_penyewa', 'Penyewa@store')->middleware('jwt.verify');
Route::put('/ubah_penyewa/{id}', 'Penyewa@update')->middleware('jwt.verify');
Route::get('/tampil_penyewa', 'Penyewa@tampil')->middleware('jwt.verify');
Route::get('/index_penyewa/{id}', 'Penyewa@index')->middleware('jwt.verify');
Route::delete('/hapus_penyewa/{id}', 'Penyewa@destroy')->middleware('jwt.verify');

Route::post('/simpan_jenismobil', 'Jenismobil@store')->middleware('jwt.verify');
Route::put('/ubah_jenismobil/{id}', 'Jenismobil@update')->middleware('jwt.verify');
Route::get('/tampil_jenismobil', 'Jenismobil@tampil')->middleware('jwt.verify');
Route::get('/index_jenismobil/{id}', 'Jenismobil@index')->middleware('jwt.verify');
Route::delete('/hapus_jenismobil/{id}', 'Jenismobil@destroy')->middleware('jwt.verify');

Route::post('/simpan_mobil', 'Mobil@store')->middleware('jwt.verify');
Route::put('/ubah_mobil/{id}', 'Mobil@update')->middleware('jwt.verify');
Route::get('/tampil_mobil', 'Mobil@tampil')->middleware('jwt.verify');
Route::get('/index_mobil/{id}', 'Mobil@index')->middleware('jwt.verify');
Route::delete('/hapus_mobil/{id}', 'Mobil@destroy')->middleware('jwt.verify');

Route::post('/simpan_transaksi', 'Transaksi@store')->middleware('jwt.verify');
Route::put('/ubah_transaksi/{id}', 'Transaksi@update')->middleware('jwt.verify');
Route::post('/tampil_transaksi', 'Transaksi@show')->middleware('jwt.verify');
Route::get('/index_transaksi/{id}', 'Transaksi@index')->middleware('jwt.verify');
Route::delete('/hapus_transaksi/{id}', 'Transaksi@destroy')->middleware('jwt.verify');

Route::post('/simpan_detail', 'detailtrans@store')->middleware('jwt.verify');
Route::delete('/hapus_detail/{id}', 'detailtrans@destroy')->middleware('jwt.verify');
Route::get('/tampil_detail', 'detailtrans@tampil')->middleware('jwt.verify');
Route::put('/ubah_detail/{id}', 'detailtrans@update')->middleware('jwt.verify');
```

Controller Petugas

```php
<?php

namespace App\Http\Controllers;

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

class Petugas 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'));
    }

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

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

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

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

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

Lanjutan controller petugas

```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'));
    }
    public function tampil()
    {
        $data=['status'=>Auth::user()->nama_petugas];
        return response()->json($data);
    }
```

Controller penyewa

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\PenyewaModel;
use Auth;
use DB;
use Illuminate\Support\Facades\Validator;
class Penyewa extends Controller
{
    public function index($id)
    {
        if(Auth::user()->level=="admin"){
            $penyewa=DB::table('penyewa')
            ->where('penyewa.id',$id)
            ->get();
            return response()->json($penyewa);
        }else{
            return response()->json(['status'=>'anda bukan admin']);
        }
    }
    public function store(Request $req)
    {
        if(Auth::user()->level=="admin"){
        $validator=Validator::make($req->all(),
        [
            'nama_penyewa'=>'required',
            'telp'=>'required',
            'no_ktp'=>'required',
            'alamat'=>'required'
        ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $simpan=PenyewaModel::create([
            'nama_penyewa'=>$req->nama_penyewa,
            'telp'=>$req->telp,
            'no_ktp'=>$req->no_ktp,
            'alamat'=>$req->alamat
        ]);
        $status=1;
        $message="Penyewa Berhasil Ditambahkan";
        if($simpan){
          return Response()->json(compact('status','message'));
        }else {
          return Response()->json(['status'=>0]);
        }
    }
    else {
        return response()->json(['status'=>'anda bukan admin']);
```

Lanjutan controller penyewa

```php
    }
    public function update($id,Request $request){
        if(Auth::user()->level=="admin"){
        $validator=Validator::make($request->all(),
            [
                'nama_penyewa'=>'required',
                'telp'=>'required',
                'no_ktp'=>'required',
                'alamat'=>'required'
            ]
        );

        if($validator->fails()){
        return Response()->json($validator->errors());
        }

        $ubah=PenyewaModel::where('id',$id)->update([
            'nama_penyewa'=>$request->nama_penyewa,
            'telp'=>$request->telp,
            'no_ktp'=>$request->no_ktp,
            'alamat'=>$request->alamat
        ]);
        $status=1;
        $message="Penyewa Berhasil Diubah";
        if($ubah){
        return Response()->json(compact('status','message'));
        }else {
        return Response()->json(['status'=>0]);
        }
        }
        else {
        return response()->json(['status'=>'anda bukan admin']);
        }
    }

    public function destroy($id){
        if(Auth::user()->level=="admin"){
        $hapus=PenyewaModel::where('id',$id)->delete();
        $status=1;
        $message="Penyewa Berhasil Dihapus";
        if($hapus){
        return Response()->json(compact('status','message'));
        }else {
        return Response()->json(['status'=>0]);
        }
    }
    else {
        return response()->json(['status'=>'anda bukan admin']);
        }
    }
}
```

```php
    public function tampil(){
        if(Auth::user()->level=="admin"){
            $datas = PenyewaModel::get();
            $count = $datas->count();
            $penyewa = array();
            $status = 1;
            foreach ($datas as $dt_pl){
                $penyewa[] = array(
                    'id' => $dt_pl->id,
                    'nama_penyewa' => $dt_pl->nama_penyewa,
                    'alamat' => $dt_pl->alamat,
                    'telp' => $dt_pl->telp,
                    'no_ktp' => $dt_pl->no_ktp
                );
            }
            return Response()->json(compact('count','penyewa'));
        } else {
            return Response()->json(['status'=> 'Tidak bisa, anda bukan admin']);
        }
    }
}
```

Controller Mobil

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\MobilModel;
use Auth;
use DB;
use Illuminate\Support\Facades\Validator;
class Mobil extends Controller
{
    public function index($id)
    {
        if(Auth::user()->level=="admin"){
            $mobil=DB::table('mobil')
            ->where('mobil.id',$id)
            ->get();
            return response()->json($mobil);
        }else{
            return response()->json(['status'=>'anda bukan admin']);
        }
    }
    public function store(Request $req)
    {
        if(Auth::user()->level=="admin"){
        $validator=Validator::make($req->all(),
        [
            'nama_mobil'=>'required',
            'id_jenis_mobil'=>'required',
            'plat_nomer'=>'required',
            'kondisi'=>'required',
        ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $simpan=MobilModel::create([
            'nama_mobil'=>$req->nama_mobil,
            'id_jenis_mobil'=>$req->id_jenis_mobil,
            'plat_nomer'=>$req->plat_nomer,
            'kondisi'=>$req->kondisi
        ]);
        $status=1;
        $message="Mobil Berhasil Ditambahkan";
        if($simpan){
            return Response()->json(compact('status','message'));
        }else {
            return Response()->json(['status'=>0]);
        }
        }
        else {
            return response()->json(['status'=>'anda bukan admin']);
```

Lanjutan Controller Mobil

```php
public function update($id, Request $request){
    if(Auth::user()->level=="admin"){
    $validator=Validator::make($request->all(),
        [
            'nama_mobil'=>'required',
            'id_jenis_mobil'=>'required',
            'plat_nomer'=>'required',
            'kondisi'=>'required',
        ]
    );

    if($validator->fails()){
    return Response()->json($validator->errors());
    }

    $ubah=MobilModel::where('id',$id)->update([
        'nama_mobil'=>$request->nama_mobil,
        'id_jenis_mobil'=>$request->id_jenis_mobil,
        'plat_nomer'=>$request->plat_nomer,
        'kondisi'=>$request->kondisi
    ]);
    $status=1;
    $message="Mobil Berhasil Diubah";
    if($ubah){
    return Response()->json(compact('status','message'));
    }else {
    return Response()->json(['status'=>0]);
    }
    }
    else {
    return response()->json(['status'=>'anda bukan admin']);
    }
}

public function destroy($id){
    if(Auth::user()->level=="admin"){
    $hapus=MobilModel::where('id',$id)->delete();
    $status=1;
    $message="Mobil Berhasil Dihapus";
    if($hapus){
    return Response()->json(compact('status','message'));
    }else {
    return Response()->json(['status'=>0]);
    }
    }
    else {
        return response()->json(['status'=>'anda bukan admin']);
    }
}
```

```php
public function tampil(){
    if(Auth::user()->level=="admin"){
        $datas = MobilModel::get();
        $count = $datas->count();
        $mobil = array();
        $status = 1;
        foreach ($datas as $dt_pl){
            $mobil[] = array(
                'id' => $dt_pl->id,
                'nama_mobil' => $dt_pl->nama_mobil,
                'id_jenis_mobil' => $dt_pl->id_jenis_mobil,
                'plat_nomer' => $dt_pl->plat_nomer,
                'kondisi' => $dt_pl->kondisi
            );
        }
        return Response()->json(compact('count','mobil'));
    } else {
        return Response()->json(['status'=> 'Tidak bisa, anda bukan admin']);
    }
}
```

Controller jenis mobil

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\JenismobilModel;
use Auth;
use DB;
use Illuminate\Support\Facades\Validator;
class Jenismobil extends Controller
{
    public function index($id)
    {
        if(Auth::user()->level=="admin"){
            $jeniscuci=DB::table('jenis_cuci')
            ->where('jenis_mobil.id',$id)
            ->get();
            return response()->json($jeniscuci);
        }else{
            return response()->json(['status'=>'anda bukan admin']);
        }
    }
    public function store(Request $req)
    {
        if(Auth::user()->level=="admin"){
        $validator=Validator::make($req->all(),
        [
            'jenis_mobil'=>'required',
        ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $simpan=JenismobilModel::create([
            'jenis_mobil'=>$req->jenis_mobil
        ]);
        $status=1;
        $message="Jenis Mobil Berhasil Ditambahkan";
        if($simpan){
          return Response()->json(compact('status','message'));
        }else {
          return Response()->json(['status'=>0]);
        }
        }
        else {
            return response()->json(['status'=>'anda bukan admin']);
        }
```

Lanjutan controller jenis mobil

```php
public function update($id,Request $request){
    if(Auth::user()->level=="admin"){
    $validator=Validator::make($request->all(),
        [
            'jenis_mobil'=>'required',
        ]
    );

    if($validator->fails()){
    return Response()->json($validator->errors());
    }

    $ubah=JenismobilModel::where('id',$id)->update([
        'jenis_mobil'=>$request->jenis_mobil,
    ]);
    $status=1;
    $message="Jenis Mobil Berhasil Diubah";
    if($ubah){
    return Response()->json(compact('status','message'));
    }else {
    return Response()->json(['status'=>0]);

    }
    }
    else {
    return response()->json(['status'=>'anda bukan admin']);
    }
}
public function destroy($id){
    if(Auth::user()->level=="admin"){
    $hapus=JenismobilModel::where('id',$id)->delete();
    $status=1;
    $message="Jenis Mobil Berhasil Dihapus";
    if($hapus){
    return Response()->json(compact('status','message'));
    }else {
    return Response()->json(['status'=>0]);
    }
    }
    else {
    return response()->json(['status'=>'anda bukan admin']);
    }
}
```

```php
public function tampil(){
    if(Auth::user()->level=="admin"){
        $datas = JenismobilModel::get();
        $count = $datas->count();
        $jenismobil = array();
        $status = 1;
        foreach ($datas as $dt_jc){
            $jenismobil[] = array(
                'id' => $dt_jc->id,
                'jenis_mobil' => $dt_jc->jenis_mobil,
            );
        }
        return Response()->json(compact('count','jenismobil'));
    } else {
        return Response()->json(['status'=> 'Tidak bisa, anda bukan admin']);
    }
}
}
```

Controller detail transaksi

```php
<?php

namespace App\Http\Controllers;

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

class detailtrans extends Controller
{
    public function store(Request $req){
        if (Auth::user()->level=='petugas') {
            $validator=Validator::make($req->all(),[
                'id_transaksi' => 'required',
                'id_jenis_mobil' => 'required',
                'qty' => 'required',
            ]);
            if($validator->fails()){
                return Response()->json($validator->errors());
            }
            $harga = JenismobilModel::where('id_jenis_mobil',$req->id_jenis_mobil)->first();
            $subtotal = $harga->harga_perhari * $req->qty;

            $simpan=detailModel::create([
                'id_transaksi' => $req->id_transaksi,
                'id_jenis_mobil' => $req->id_jenis_mobil,
                'qty' => $req->qty,
                'subtotal' => $subtotal,
            ]);
            if($simpan){
                $data['status']="Berhasil";
                $data['message']="Data berhasil disimpan!";
                return Response()->json($data);
            }else{
                $data['status']="Gagal";
                $data['message']="Data gagal disimpan!";
                return Response()->json($data);
            }
        } else {
            $data['status']="Gagal";
            $data['Message']="Anda bukan Petugas!";
            return Response()->json($data);
        }
    }
    public function destroy($id){
```

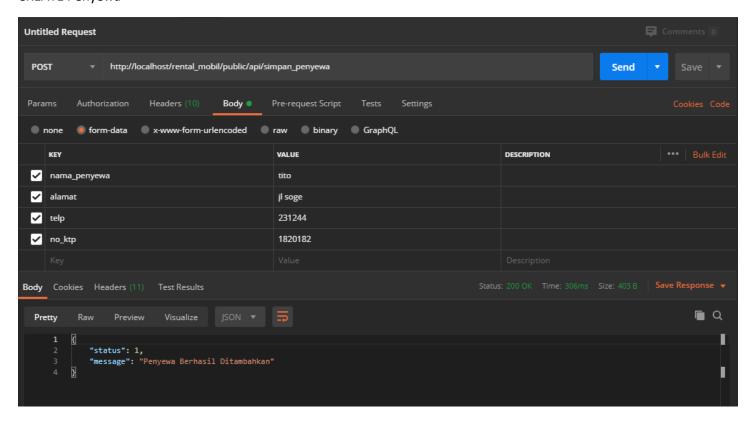Lanjutan controller detail transaksi

```php
public function destroy($id){
    if(Auth::user()->level=="petugas"){
    $hapus=detailModel::where('id',$id)->delete();
    $status=1;
    $message="Detail Transaksi Berhasil Dihapus";
    if($hapus){
    return Response()->json(compact('status','message'));
    }else {
    return Response()->json(['status'=>0]);
    }
    }
    else {
        return response()->json(['status'=>'anda bukan petugas']);
    }
}
public function tampil(){
    if(Auth::user()->level=="petugas"){
        $datas = detailModel::get();
        $count = $datas->count();
        $detail = array();
        $status = 1;
        foreach ($datas as $dt_pl){
            $detail[] = array(
                'id_transaksi' => $dt_pl->id_transaksi,
                'id_jenismobil' => $dt_pl->id_jenis_mobil,
                'qty' => $dt_pl->qty,
                'subtotal' => $dt_pl->subtotal
            );
        }
        return Response()->json(compact('count','detail'));
    } else {
        return Response()->json(['status'=> 'Tidak bisa, anda bukan petugas']);
    }
}
public function update($id,Request $request){
    if(Auth::user()->level=="petugas"){
    $validator=Validator::make($request->all(),
        [
            'id_transaksi' => 'required',
            'id_jenis_mobil' => 'required',
            'qty' => 'required',

        ]
    );

    if($validator->fails()){
    return Response()->json($validator->errors());
    }

    $harga = JenismobilModel::where('id_jenis_mobil',$request->id_jenis_mobil)->first();
    $subtotal = $harga->harga_perhari * $request->qty;
```

```php
        $ubah=detailModel::where('id',$id)->update([
                        'id_transaksi' => $request->id_transaksi,
                        'id_jenis_mobil' => $request->id_jenis_mobil,
                        'qty' => $request->qty,
                        'subtotal' => $subtotal
        ]);
        $status=1;
        $message="Detail Berhasil Diubah";
        if($ubah){
        return Response()->json(compact('status','message'));
        }else {
        return Response()->json(['status'=>0]);
        }
        }
    else {
    return response()->json(['status'=>'anda bukan petugas']);
    }
    }
}
```

Controller Transaksi

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\transaksiModel;
use App\PenyewaModel;
use App\JenismobilModel;
use App\detailModel;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use JWTAuth;
use Auth;
use DB;
use Tymon\JWTAuth\Exceptions\JWTException;

class Transaksi extends Controller
{
    public function store(Request $req){
        if (Auth::user()->level=='petugas') {
            $validator=Validator::make($req->all(),[
                'id_mobil' => 'required',
                'id_petugas' => 'required',
                'id_penyewa' => 'required',
                'tgl_transaksi' => 'required',
                'tgl_selesai' => 'required',
            ]);
            if($validator->fails()){
                return Response()->json($validator->errors());
            }

            $simpan=transaksiModel::create([
                'id_mobil' => $req->id_mobil,
                'id_petugas' => $req->id_petugas,
                'id_penyewa' => $req->id_penyewa,
                'tgl_transaksi' => $req->tgl_transaksi,
                'tgl_selesai' => $req->tgl_selesai
            ]);
            if($simpan){
                $data['status']="Berhasil";
                $data['message']="Data berhasil disimpan!";
                return Response()->json($data);
            }else{
                $data['status']="Gagal";
                $data['message']="Data gagal disimpan!";
                return Response()->json($data);
            }
        } else {
            $data['status']="Gagal";
            $data['Message']="Anda bukan Petugas!";
            return Response()->json($data);
        }
```

Lanjutan controller transaksi
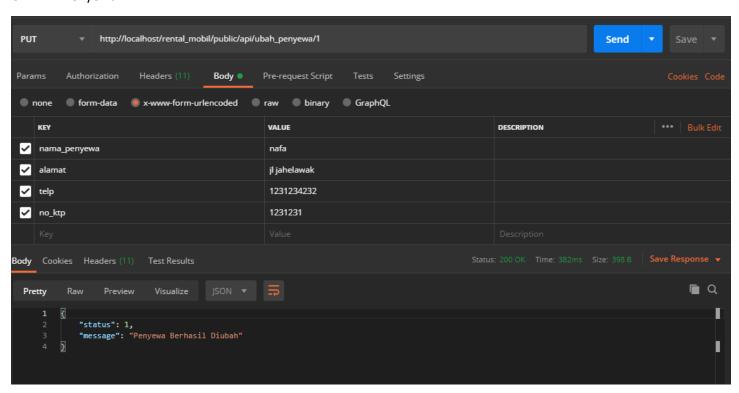
```php
public function destroy($id){
    if(Auth::user()->level=="petugas"){
    $hapus=transaksiModel::where('id',$id)->delete();
    $status=1;
    $message="Transaksi Berhasil Dihapus";
    if($hapus){
    return Response()->json(compact('status','message'));
    }else {
    return Response()->json(['status'=>0]);
    }
}
else {
    return response()->json(['status'=>'anda bukan petugas']);
    }
}

public function show(Request $req){
    if(Auth::user()->level == "petugas"){
        $transaksi = DB::table('transaksi')->join('penyewa','penyewa.id','=','transaksi.id_penyewa')
        ->where('transaksi.tgl_transaksi','>=',$req->tgl_transaksi)
        ->where('transaksi.tgl_transaksi','<=',$req->tgl_selesai)
        ->select('nama_penyewa','telp','alamat','no_ktp','transaksi.id','tgl_transaksi','tgl_selesai')
        ->get();

        if($transaksi->count() > 0){

        $data_transaksi = array();
        foreach ($transaksi as $t){

            $grand = DB::table('detail_transaksi')->where('id','=',$t->id)
            ->groupBy('id')
            ->select(DB::raw('sum(subtotal) as grandtotal'))
            ->first();

            $detail = DB::table('detail_transaksi')->join('jenis_mobil','detail_transaksi.id_jenis_mobil','=','jenis_mobil.id_jenis_mobil')
            ->where('id','=',$t->id)
            ->get();

            $data_transaksi[] = array(
                'Tanggal Ambil' => $t->tgl_transaksi,
                'nama pelanggan' => $t->nama_penyewa,
                'alamat' => $t->alamat,
                'telp' => $t->telp,
                'no_ktp' => $t->no_ktp,
                'Tanggal Kembali' => $t->tgl_selesai,
                'Grand Total' => $grand,
                'Detail' => $detail,
            );

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

Lanjutan controller

```php
            return response()->json(compact('data_transaksi'));

    }else{
            $status = 'tidak ada transaksi antara tanggal '.$req->tgl_transaksi.' sampai dengan tanggal '.$req->tgl_selesai;
            return response()->json(compact('status'));
    }
        }else{
            return Response()->json('Anda Bukan Petugas');
        }
}
public function update($id,Request $request){
    if(Auth::user()->level=="petugas"){
    $validator=Validator::make($request->all(),
        [
                'id_mobil' => 'required',
                'id_petugas' => 'required',
                'id_penyewa' => 'required',
                'tgl_transaksi' => 'required',
                'tgl_selesai' => 'required',
        ]
    );

    if($validator->fails()){
    return Response()->json($validator->errors());
    }

    $ubah=transaksiModel::where('id',$id)->update([
        'id_mobil' => $request->id_mobil,
        'id_petugas' => $request->id_petugas,
        'id_penyewa' => $request->id_penyewa,
        'tgl_transaksi' => $request->tgl_transaksi,
        'tgl_selesai' => $request->tgl_selesai
    ]);
    $status=1;
    $message="Transaksi Berhasil Diubah";
    if($ubah){
    return Response()->json(compact('status','message'));
    }else {
    return Response()->json(['status'=>0]);
    }
    }
else {
return response()->json(['status'=>'anda bukan petugas']);
}
}
}
```
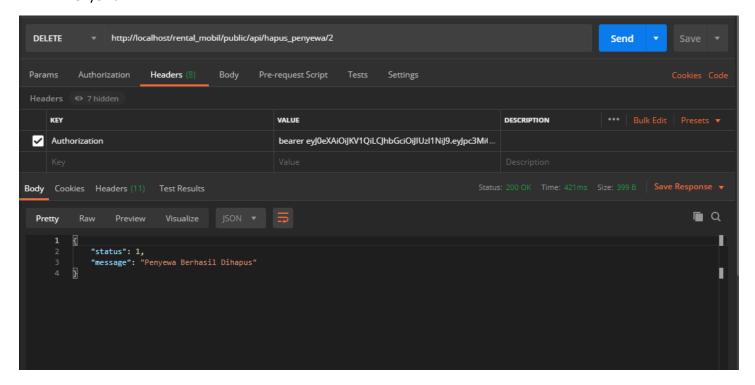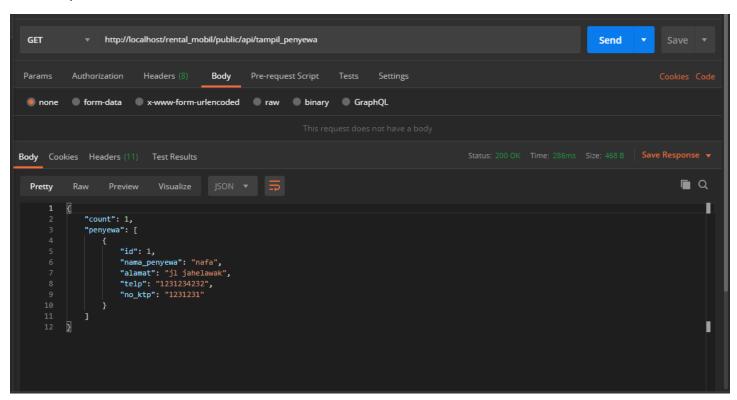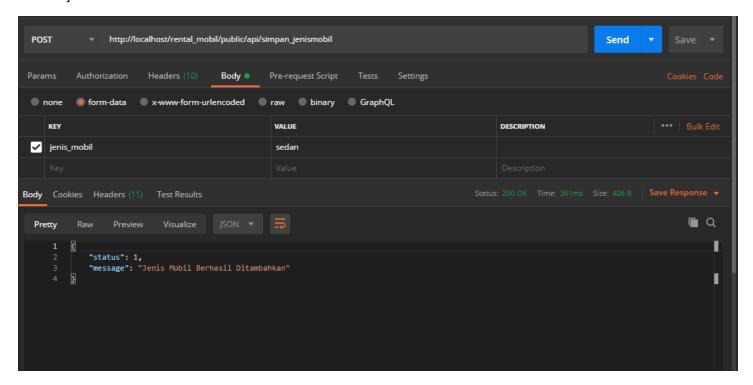
CREATE Penyewa

Untitled Request

POST http://localhost/rental_mobil/public/api/simpan_penyewa **Send** Save

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

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

KEY	VALUE	DESCRIPTION	··· Bulk Edit
☑ nama_penyewa	tito		
☑ alamat	jl soge		
☑ telp	231244		
☑ no_ktp	1820182		
Key	Value	Description	

Body Cookies Headers (11) Test Results   Status: 200 OK Time: 306ms Size: 403 B   Save Response ▾

Pretty Raw Preview Visualize JSON ▾

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

UPDATE Penyewa

PUT http://localhost/rental_mobil/public/api/ubah_penyewa/1 **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	··· Bulk Edit
☑ nama_penyewa	nafa		
☑ alamat	jl jahelawak		
☑ telp	1231234232		
☑ no_ktp	1231231		
Key	Value	Description	

Body Cookies Headers (11) Test Results   Status: 200 OK Time: 382ms Size: 398 B   Save Response ▾

Pretty Raw Preview Visualize JSON ▾

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

DELETE Penyewa

READ Penyewa

CREATE jenis mobil

POST http://localhost/rental_mobil/public/api/simpan_jenismobil

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

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

KEY	VALUE	DESCRIPTION
jenis_mobil	sedan	
Key	Value	Description

Body   Cookies   Headers (11)   Test Results   Status: 200 OK   Time: 361ms   Size: 406 B

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

UPDATE jenis mobil

PUT http://localhost/rental_mobil/public/api/ubah_jenismobil/1

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

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

KEY	VALUE	DESCRIPTION
jenis_mobil	mini bus	
Key	Value	Description

Body   Cookies   Headers (11)   Test Results   Status: 200 OK   Time: 340ms   Size: 402 B

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

DELETE jenis mobil

READ jenis mobil

CREATE Mobil

UPDATE

DELETE mobil

READ mobil
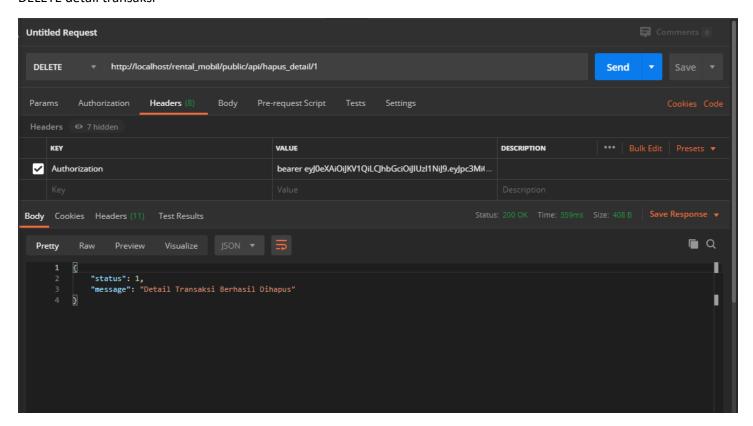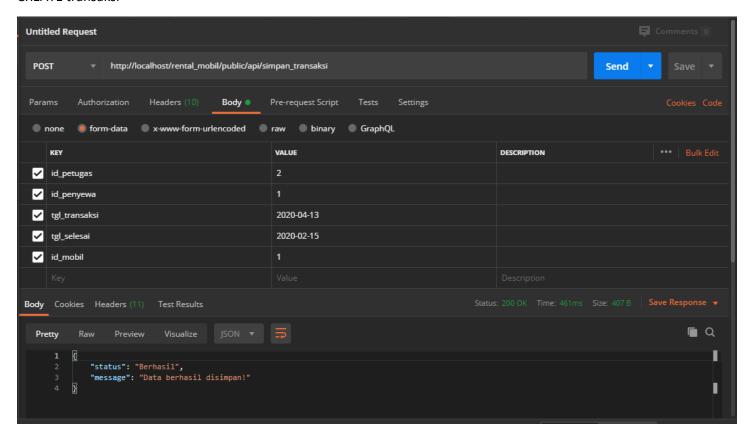
CREATE detail transaksi

UPDATE Detail transaksi

READ detail transaksi

DELETE detail transaksi

CREATE transaksi

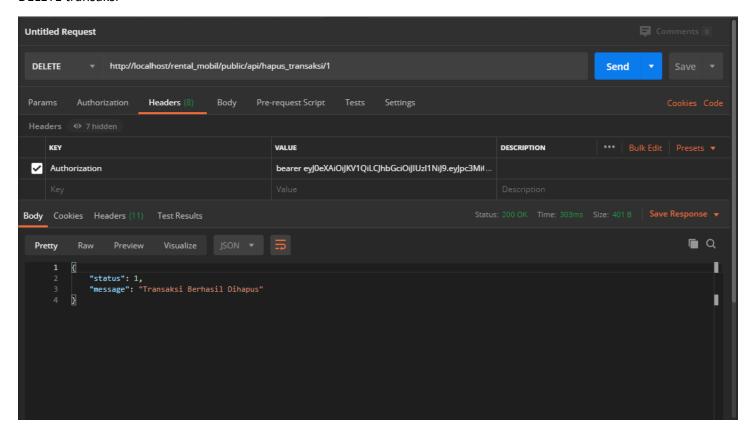UPDATE transaksi

DELETE transaksi

Tampil transaksi