1. BukuModel.php

2. Buku create

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\BukuModel;
use Validator;
use JWTAuth;
use Auth;
class Bukucontroller extends Controller
    public function store(Request $req)
         if(Auth::user()->level=="admin") {
        $validator = Validator::make($req->all(),
                  'judul'=>'required',
                 'penerbit'=>'required',
                 'pengarang'=>'required',
'foto'=>'required',
        );
if($validator->fails()){
Pesnonse()->js
             return Response()->json($validator->errors());
             $simpan = BukuModel::create([
                'judul'=>$req->judul,
                 'penerbit'=>$req->penerbit,
                 'pengarang'=>$req->pengarang,
'foto'=>$req->foto
        if($simpan){
            $status="1";
             $message="Data berhasil ditambahkan!";
             $status="0";
             $message="Data tidak berhasil ditambahkan";
             return Response()->json(compact('status', 'message'));
      } else {
         echo "Maaf, anda bukan admin.";
```

## 3. Buku update

```
public function update($id, Request $req){
    if(Auth::user()->level=="admin") {
    $validator=Validator::make($req->all(),
             'penerbit'=>'required',
            'pengarang'=>'required',
'foto'=>'required'
    if($validator->fails()){
        return Response()->json($validator->errors());
    $ubah=BukuModel::where('id', $id)->update([
        'judul'=>$req->judul,
        'penerbit'=>$req->penerbit,
        'pengarang'=>$req->pengarang,
        'foto'=>$req->foto
    if($ubah){
        $status="1";
        $message="Data berhasil diubah!";
    } else {
       $status="0";
        $message="Data tidak berhasil diubah";
       return Response()->json(compact('status', 'message'));
   echo "Maaf, anda bukan admin.";
```

#### 4. Buku delete

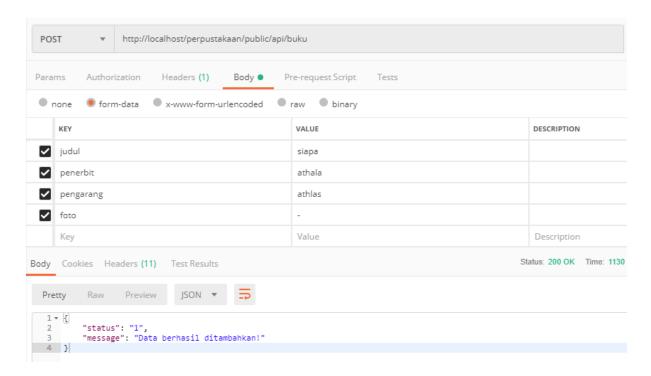
```
public function destroy($id){
    if(Auth::user()->level=="admin") {
        $hapus=BukuModel::where('id', $id)->delete();
        if($hapus){
            $status="1";
            $message="Data berhasil dihapus!";
        } else {
                $status="0";
                $message="Data tidak berhasil dihapus";
        }
        return Response()->json(compact('status','message'));
    } else {
        echo "Maaf, anda bukan admin.";
    }
}
```

#### 5. Buku Read

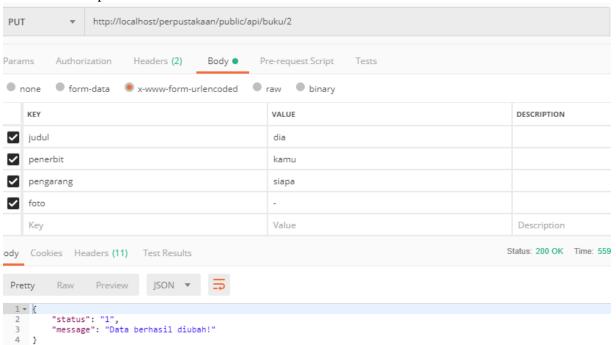
```
public function show(){
    if(Auth::user()->level=="admin") {
    $buku = BukuModel::all();
    $status="1";

    return Response()->json(compact('buku', 'status'));
} else {
    echo "Maaf, anda bukan admin.";
}
```

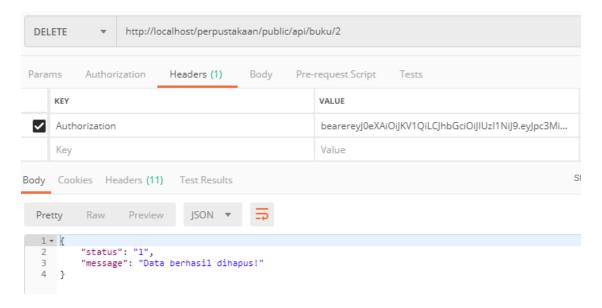
### 6. Postman buku create



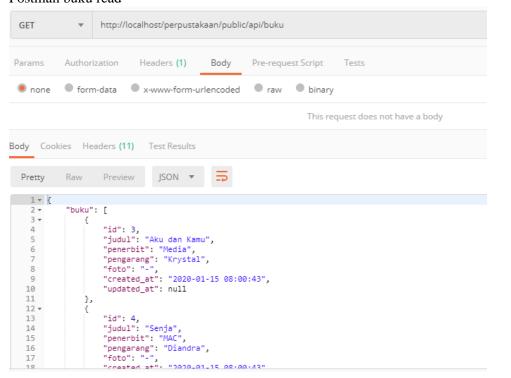
## 7. Postman buku update



#### 8. Postman buku delete



#### 9. Postman buku read



#### **ANGGOTA**

10. AnggotaModel.php

```
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class AnggotaModel extends Model
8  {
9    protected $table = "anggota";
protected $primaryKey = "id";
public $timestamps = false;
protected $fillable = [
13
14  ];
15  }
16</pre>
```

11. Anggota Create

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\AnggotaModel;
use Validator;
use JWTAuth;
use Auth;
class Anggotacontroller extends Controller
     public function store(Request $req)
         if(Auth::user()->level=="admin") {
             $validator = Validator::make($req->all(),
                      'nama_anggota'=>'required',
                      'alamat'=>'required',
'telp'=>'required'
             if($validator->fails()){
                return Response()->json($validator->erors());
             } else {
                $simpan = AnggotaModel::create([
                     'nama_anggota'=>$req->nama_anggota,
'alamat'=>$req->alamat,
                     'telp'=>$req->telp
             if($simpan){
                $status="1";
                 $message="Data berhasil ditambahkan!";
                 $status="0";
                 $message="Data tidak berhasil ditambahkan";
                 return Response()->json(compact('status','message'));
         } else {
             echo "Maaf, data hanya dapat diakses oleh admin.";
```

12. Anggota Update

```
public function update($id, Request $req){
    if(Auth::user()->level=="admin"){
        $validator=Validator::make($req->all(),
                'nama_anggota'=>'required',
                'alamat'=>'required',
                'telp'=>'required'
        if($validator->fails()){
            return Response()->json($validator->errors());
        $ubah=AnggotaModel::where('id', $id)->update([
            'nama_anggota'=>$req->nama_anggota,
            'alamat'=>$req->alamat,
            'telp'=>$req->telp
        1);
        if($ubah){
            $status="1";
            $message="Data berhasil diubah!";
        } else {
            $status="0";
            $message="Data tidak berhasil diubah";
            return Response()->json(compact('status', 'message'));
    } else {
        echo "Maaf, data hanya dapat diakses oleh admin.";
```

13. Anggota Delete

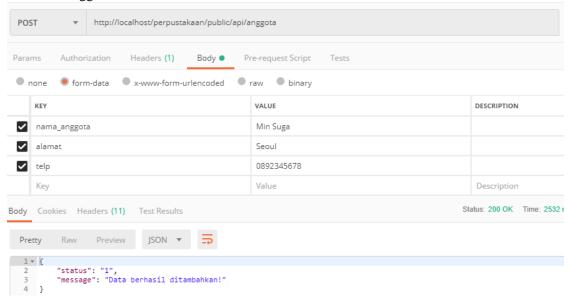
```
public function destroy($id){
    if(Auth::user()->level=="admin"){
        $hapus=AnggotaModel::where('id', $id)->delete();
        if($hapus){
            $status="1";
            $message="Data berhasil dihapus!";
        } else {
            $status="0";
            $message="Data tidak berhasil dihapus";
        }
        return Response()->json(compact('status','message'));
    }else {
        echo "Maaf, data hanya dapat diakses oleh admin.";
    }
}
```

14. Anggota Read

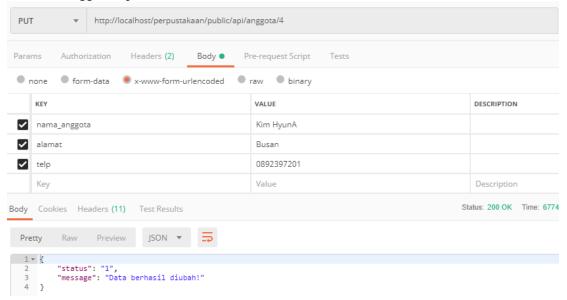
```
public function show(){
    if(Auth::user()->level=="admin"){
        $anggota = AnggotaModel::all();
        $status="1";

        return Response()->json(compact('anggota', 'status'));
    }else {
        echo "Maaf, data hanya dapat diakses oleh admin.";
    }
}
```

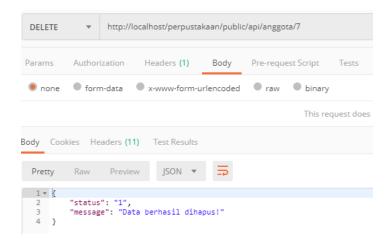
## 15. Postman Anggota Create



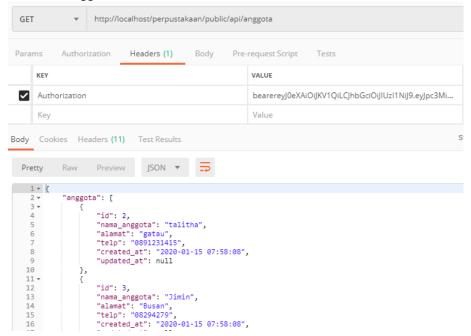
# 16. Postman Anggota Update



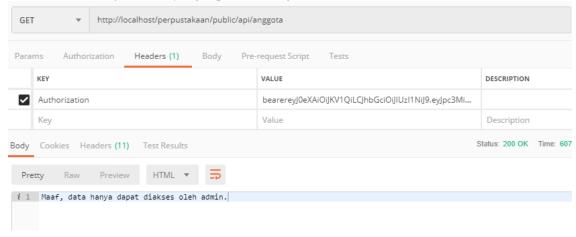
## 17. Postman Anggota Delete



## 18. Postman Anggota Read



19. Salah satu contoh jika level yang dipilih saat login bukan admin



NAMA : Talitha Sevrilla Duriga

No. : 38

Kelas : XI RPL Laravel