Nama: Chatlea Cinta Putri Widyanto

Kelas : XI Laravel

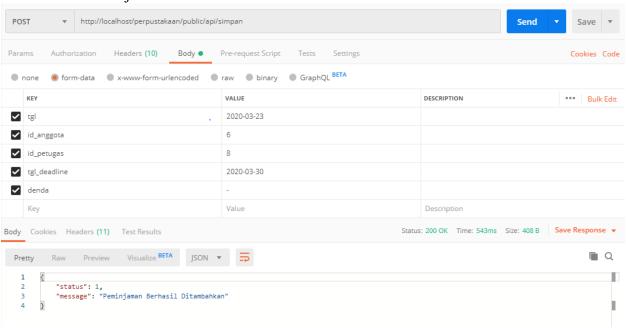
No. Absen: 09

TUGAS 6 TRANSAKSI PEMINJAMAN BUKU

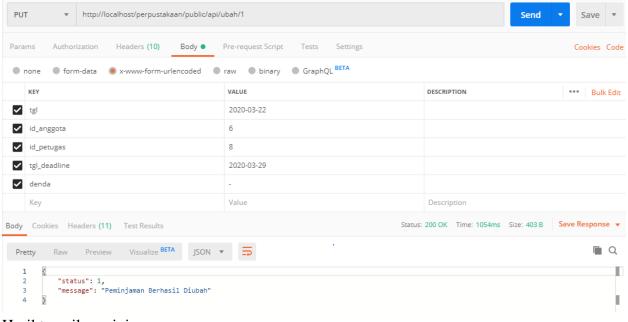
Hasil Peminjaman

Peminjaman hanya dapat diakses oleh petugas

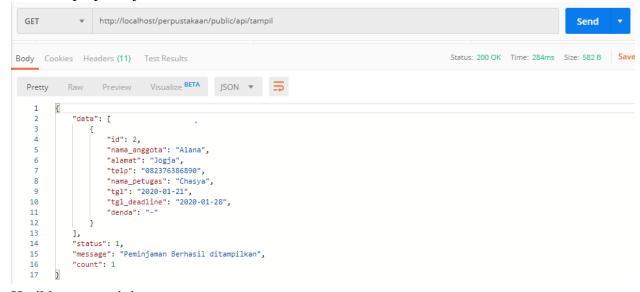
• Hasil Create Peminjaman



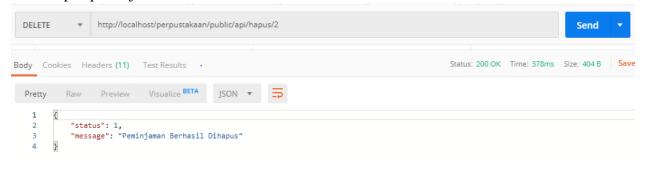
• Hasil update peminjaman



• Hasil tampil peminjaman



• Hasil hapus peminjaman



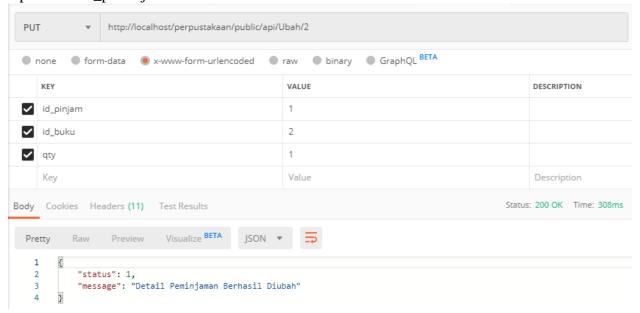
Hasil detail_peminjaman

Detail peminjaman hanya dapat diakses oleh admin

• Create detail_peminjaman



• Update detail_peminjaman

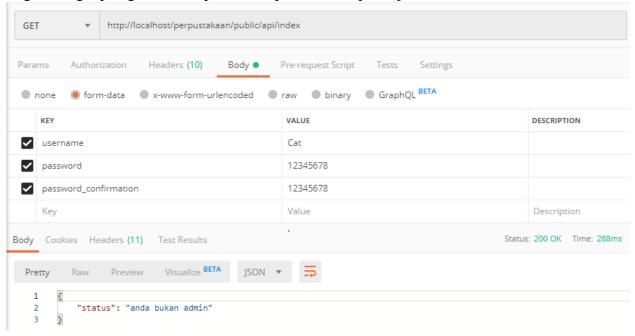


• Delete detail_peminjaman



Tampil detail peminjaman hanya dapat diakses oleh admin

• Login sebagai petugas tidak dapat menampilkan detail peminjaman



• Login sebagai admin

```
GET
                    http://localhost/perpustakaan/public/api/index
                                                                                                         Status: 200 OK Time: 266ms
Body Cookies Headers (11) Test Results
                                Visualize BETA
  Pretty
                    Preview
    2
             "data": [
    3
    4
                     "id": 1,
                      "id_anggota": 8,
    5
                     "nama_anggota": "Alana",
    6
                      "id_petugas": 8,
                      "tgl": "2020-03-22",
    8
                      "tgl_deadline": "2020-03-29",
    9
   10
                      "denda": "-",
                      "detail_buku": [
   11
   12
                             "id_pinjam": 1,
   13
                              "id_buku": 5,
   14
   15
                              "qty": 1
   16
                         }.
   17
   18
                             "id_pinjam": 1,
                              "id_buku": 2,
   19
                              "qty": 1
   20
   21
   22
   23
   24
         }
   25
```

SS Model

• Model peminjaman

```
m Peminjaman.php
 namespace App;
 use Illuminate\Database\Eloquent\Model;
 class Peminjaman extends Model
     protected $table = "peminjaman";
     protected $primaryKey = "id";
     protected $fillable = [
         'tgl',
         'id_anggota',
         'id_petugas',
         'tgl_deadline',
         'denda',
     ];
     public function Anggota_model(){
         return $this->belongsTo('App\Anggota_model', 'id_anggota');
     public function Petugas_model(){
         return $this->belongsTo('App\Petugas_model', 'id_petugas');
     public function detail(){
      return $this->hasOne('App\detail_peminjaman','id');
     public $timestamps = false;
```

• Model detail_peminjaman

SS Controller

```
Http > Controllers > 💝 PeminjamanController.php
 <?php
 namespace App\Http\Controllers;
 use Illuminate\Http\Request;
 use App\Peminjaman;
 use App\Buku;
 use App\detail_peminjaman;
 use JWTAuth;
 use DB;
 use Auth;
 use Illuminate\Support\Facades\Validator;
 use Tymon\JWTAuth\Exceptions\JWTException;
 class PeminjamanController extends Controller
   public function index(){
     if(Auth::user()->level=="admin"){
       $peminjaman=DB::table('peminjaman')
       ->join('anggota','anggota.id','peminjaman.id_anggota')
       ->join('petugas','petugas.id','peminjaman.id_petugas')
       ->select('peminjaman.id','peminjaman.id_anggota','anggota.nama_anggota','peminjaman.id_petugas'
                  ,'peminjaman.tgl','peminjaman.tgl_deadline','peminjaman.denda')
       ->get();
       $data=array();
       foreach ($peminjaman as $dt_pj){
         $ok=detail_peminjaman::where('id_pinjam',$dt_pj->id)->get();
         $arr_detail=array();
         foreach ($ok as $ok) {
           $arr_detail[]=array(
            'id_pinjam' =>$ok->id_pinjam,
            'id_buku' => $ok->id_buku,
            'qty' => $ok->qty
          $data[]=array(
```

```
$data[]=array(
        'id' => $dt_pj->id,
        'id_anggota' => $dt_pj->id_anggota,
        'nama_anggota' => $dt_pj->nama_anggota,
        'id_petugas' => $dt_pj->id_petugas,
       'tgl' => $dt_pj->tgl,
        'tgl_deadline' => $dt_pj->tgl_deadline,
        'denda' => $dt_pj->denda,
        'detail_buku' => $arr_detail
      );
   return response()->json(compact('data'));
   return response()->json(['status'=>'anda bukan admin']);
}
 public function store(Request $request){
   $validator=Validator::make($request->all(),
        'id_anggota'=>'required',
       'id_petugas'=>'required',
        'tgl'=>'required',
        'tgl_deadline'=>'required',
        'denda'=>'required'
   if($validator->fails()){
     return Response()->json($validator->errors());
   $simpan=Peminjaman::create([
      'id_anggota'=>$request->id_anggota,
      'id_petugas'=>$request->id_petugas,
      'tgl'=>$request->tgl,
      'tgl_deadline'=>$request->tgl_deadline,
      'denda'=>$request->denda
   1);
```

```
1);
  $status=1;
  $message="Peminjaman Berhasil Ditambahkan";
  if($simpan){
    return Response()->json(compact('status', 'message'));
  }else {
    return Response()->json(['status'=>0]);
public function update($id,Request $request){
  $validator=Validator::make($request->all(),
      'id_anggota'=>'required',
      'id_petugas'=>'required',
      'tgl'=>'required',
      'tgl_deadline'=>'required',
      'denda'=>'required'
);
if($validator->fails()){
 return Response()->json($validator->errors());
$ubah=Peminjaman::where('id',$id)->update([
  'id_anggota'=>$request->id_anggota,
  'id petugas'=>$request->id petugas,
  'tgl'=>$request->tgl,
  'tgl_deadline'=>$request->tgl_deadline,
  'denda'=>$request->denda
1);
$status=1;
$message="Peminjaman Berhasil Diubah";
if($ubah){
 return Response()->json(compact('status', 'message'));
}else {
 return Response()->json(['status'=>0]);
}
```

```
public function tampil_pinjam(){
 $data=DB::table('peminjaman')
  ->join('anggota','anggota.id','=','peminjaman.id_anggota')
  ->join('petugas','petugas.id','=','peminjaman.id_petugas')
  ->select('peminjaman.id','anggota.nama_anggota','anggota.alamat','anggota.telp',
         | 'petugas.nama_petugas', 'peminjaman.tgl', 'peminjaman.tgl_deadline', 'peminjaman.denda
  ->get();
 $count=$data->count();
 $status=1;
 $message="Peminjaman Berhasil ditampilkan";
  return response()->json(compact('data','status','message','count'));
public function destroy($id){
 $hapus=Peminjaman::where('id',$id)->delete();
 $status=1;
 $message="Peminjaman Berhasil Dihapus";
  if($hapus){
   return Response()->json(compact('status','message'));
    return Response()->json(['status'=>0]);
public function simpan(Request $request){
 $validator=Validator::make($request->all(),
      'id_pinjam'=>'required',
      'id_buku'=>'required',
      'qty'=>'required'
  if($validator->fails()){
  return Response()->json($validator->errors());
 $simpan=detail_peminjaman::create([
```

```
$simpan=detail_peminjaman::create([
    'id_pinjam'=>$request->id_pinjam,
    'id_buku'=>$request->id_buku,
    'qty'=>$request->qty
  1);
  $status=1;
  $message="Detail Peminjaman Berhasil Ditambahkan";
  if($simpan){
    return Response()->json(compact('status', 'message'));
  }else {
  return Response()->json(['status'=>0]);
}
public function ubah($id,Request $request){
  $validator=Validator::make($request->all(),
      'id_pinjam'=>'required',
      'id_buku'=>'required',
      'qty'=>'required'
);
if($validator->fails()){
  return Response()->json($validator->errors());
$ubah=detail_peminjaman::where('id',$id)->update([
  'id_pinjam'=>$request->id_pinjam,
  'id_buku'=>$request->id_buku,
  'qty'=>$request->qty
1);
$status=1;
$message="Detail Peminjaman Berhasil Diubah";
if($ubah){
  return Response()->json(compact('status', 'message'));
}else {
  return Response()->json(['status'=>0]);
```

```
return Response()->json(compact('status', 'message'));
}else {
  return Response()->json(['status'=>0]);
}

public function hapus($id){
  $hapus=detail_peminjaman::where('id',$id)->delete();
  $status=1;
  $message="Detail Peminjaman Berhasil Dihapus";
  if($hapus){
    return Response()->json(compact('status', 'message'));
  }else {
    return Response()->json(['status'=>0]);
  }
}
```

SS api.php

```
Route::get('index','PeminjamanController@index')->middleware('jwt.verify');
Route::post('simpan','PeminjamanController@store')->middleware('jwt.verify');
Route::put('/ubah/{id}','PeminjamanController@update')->middleware('jwt.verify');
Route::delete('hapus/{id}','PeminjamanController@destroy')->middleware('jwt.verify');
Route::get("tampil","PeminjamanController@tampil_pinjam")->middleware('jwt.verify');
```