

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

The screenshot shows a REST client interface with a POST request to `http://localhost/perpustakaan/public/api/simpan`. The request body is form-data with the following fields:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> tgl	2020-03-23	
<input checked="" type="checkbox"/> id_anggota	6	
<input checked="" type="checkbox"/> id_petugas	8	
<input checked="" type="checkbox"/> tgl_deadline	2020-03-30	
<input checked="" type="checkbox"/> denda	-	
Key	Value	Description

The response status is 200 OK, Time: 543ms, Size: 408 B. The response body is JSON:

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

- Hasil update peminjaman

The screenshot shows a REST client interface with a PUT request to `http://localhost/perpustakaan/public/api/ubah/1`. The request body is `x-www-form-urlencoded` and contains the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> tgl	2020-03-22	
<input checked="" type="checkbox"/> id_anggota	6	
<input checked="" type="checkbox"/> id_petugas	8	
<input checked="" type="checkbox"/> tgl_deadline	2020-03-29	
<input checked="" type="checkbox"/> denda	-	

The response status is 200 OK, with a time of 1054ms and size of 403 B. The response body is JSON:

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

- Hasil tampil peminjaman

The screenshot shows a REST client interface with a GET request to `http://localhost/perpustakaan/public/api/tampil`. The response status is 200 OK, with a time of 284ms and size of 582 B. The response body is JSON:

```
1 {
2   "data": [
3     {
4       "id": 2,
5       "nama_anggota": "Alana",
6       "alamat": "Jogja",
7       "telp": "082376386890",
8       "nama_petugas": "Chasya",
9       "tgl": "2020-01-21",
10      "tgl_deadline": "2020-01-28",
11      "denda": "-"
12    }
13  ],
14  "status": 1,
15  "message": "Peminjaman Berhasil ditampilkan",
16  "count": 1
17 }
```

- Hasil hapus peminjaman

The screenshot shows a REST client interface with a DELETE request to `http://localhost/perpustakaan/public/api/hapus/2`. The response status is 200 OK, with a time of 378ms and size of 404 B. The response body is JSON:

```
1 {
2   "status": 1,
3   "message": "Peminjaman Berhasil Dihapus"
4 }
```

## Hasil detail\_peminjaman

Detail peminjaman hanya dapat diakses oleh admin

- Create detail\_peminjaman

POST ▼ http://localhost/perpustakaan/public/api/Simpan Send ▼

<input checked="" type="checkbox"/>	id_buku	5	
<input checked="" type="checkbox"/>	id_pinjam	1	
<input checked="" type="checkbox"/>	qty	1	
	Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 3.52s Size: 415 B Save

Pretty Raw Preview Visualize BETA JSON ▼ ≡

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

- Update detail\_peminjaman

PUT ▼ http://localhost/perpustakaan/public/api/Ubah/2

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

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	id_pinjam	1	
<input checked="" type="checkbox"/>	id_buku	2	
<input checked="" type="checkbox"/>	qty	1	
	Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 308ms

Pretty Raw Preview Visualize BETA JSON ▼ ≡

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

- Delete detail\_peminjaman

DELETE ▼ http://localhost/perpustakaan/public/api/Hapus/3

Body Cookies Headers (11) Test Results Status: 200 OK Time: 381ms

Pretty Raw Preview Visualize BETA JSON ▼ ≡

```
1 {
2   "status": 1,
3   "message": "Detail Peminjaman Berhasil Dihapus"
4 }
```

Tampil detail peminjaman hanya dapat diakses oleh admin

- Login sebagai petugas tidak dapat menampilkan detail peminjaman

GET http://localhost/perpustakaan/public/api/index

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

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> username	Cat	
<input checked="" type="checkbox"/> password	12345678	
<input checked="" type="checkbox"/> password_confirmation	12345678	
Key	Value	Description

Body Cookies Headers (11) Test Results Status: 200 OK Time: 288ms

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": "anda bukan admin"
3 }
```

- Login sebagai admin

GET http://localhost/perpustakaan/public/api/index

Body Cookies Headers (11) Test Results Status: 200 OK Time: 266ms

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "data": [
3     {
4       "id": 1,
5       "id_anggota": 8,
6       "nama_anggota": "Alana",
7       "id_petugas": 8,
8       "tgl": "2020-03-22",
9       "tgl_deadline": "2020-03-29",
10      "denda": "-",
11      "detail_buku": [
12        {
13          "id_pinjam": 1,
14          "id_buku": 5,
15          "qty": 1
16        },
17        {
18          "id_pinjam": 1,
19          "id_buku": 2,
20          "qty": 1
21        }
22      ]
23    }
24  ]
25 }
```

## SS Model

- Model peminjaman

```
Peminjaman.php
<?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

```
detail_peminjaman.php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class detail_peminjaman extends Model
{
    protected $table="detail_peminjaman";
    protected $primaryKey="id";
    protected $fillable = [
        'id_pinjam',
        'id_buku',
        'qty'
    ];

    public function buku() {
        return $this->belongsTo('App/Buku', 'id_buku');
    }
    public function peminjaman() {
        return $this->belongsTo('App/Peminjaman', 'id_pinjam');
    }
}
```

## 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'));
} else{
    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
    ]);
    $status=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
    ]);
    $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'));
    }else {
        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
]);
$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
    ]);
    $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');

```