

Nama: Daffa Annisa Hidayatul Ays Sujianto

Kelas/no.abs: XI RPL 2 /09

Buatlah barangmodel, customermodel dan transaksimodel di command prompt dengan ketik

```
php artisan make:model nama_model
```

Buka file model barangmodel di App/barangmodel.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class barangmodel extends Model
{
    protected $table='barang';
    protected $primaryKey='kode_barang';
    protected $foreignkey='id_customer';
    public $timestamps=false;

    protected $fillable = [
        'nama_barang', 'harga', 'id_customer'
    ];
}
```

```
.env  barangmodel.php  customermodel.php

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class customermodel extends Model
8  {
9      protected $table='customer';
10     protected $primaryKey='id_customer';
11     public $timestamps=false;
12
13     protected $fillable = [
14         'nama'
15     ];
16 }
<
```

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class transaksimodel extends Model
8  {
9      protected $table='transaksi';
10     protected $primaryKey='id_transaksi';
11     public $timestamps=false;
12
13     protected $fillable = [
14         'kode_barang','id_customer'
15     ];
16 }

```

- `protected $table = 'nama_tabel';` untuk mendefinisikan model tersebut mewakili tabel apa
- `protected $primaryKey = 'kolom_primary_key';` untuk mendefinisikan apa nama kolom primary key di tabel tersebut.
- `protected $fillable = ['kolom_a', 'kolom_b', 'kolom_c', ...];` untuk mendefinisikan kolom apa saja yang terdapat pada tabel tersebut.

Buatlah file controller barangcontroller, customercontroller, dan transaksicontroller di command prompt dengan ketik :

`php artisan make:controller nama_controller`

Buka file controller barangcontroller di `App\http\controller\barangcontroller.php`

```

4
5  use Illuminate\Http\Request;
6  use App\barangmodel;
7  use Illuminate\Support\Facades\Validator;
8
9  class barangcontroller extends Controller
10 {
11     public function store(Request $req)
12     {
13         $validator=Validator::make($req->all(),
14             [
15                 'nama_barang'=>'required',
16                 'harga'      =>'required',
17                 'id_customer'=>'required'
18             ]
19         );
20         if ($validator->fails()) {
21             return Response()->json($validator->errors());
22         }
23
24         $simpan=barangmodel::create([
25             'nama_barang'=>$req->nama_barang,
26             'harga'      =>$req->harga,
27             'id_customer'=>$req->id_customer
28         ]);
29         if ($simpan) {
30             return Response()->json(['status'=>1]);
31         }else {
32             return Response()->json(['status'=>0]);
33         }
34     }
35 }
36

```

```
.env    barangmodel.php    customermodel.php    transaksimodel.php

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\customermodel;
7  use Illuminate\Support\Facades\Validator;
8
9  class customercontroller extends Controller
10 {
11     public function store(Request $req)
12     {
13         $validator=Validator::make($req->all(),
14             [
15                 'nama'=>'required'|
16             ]
17         );
18         if ($validator->fails()) {
19             return Response()->json($validator->errors());
20         }
21
22         $simpan=customermodel::create([
23             'nama'=>$req->nama
24         ]);
25         if ($simpan) {
26             return Response()->json(['status'=>1]);
27         }else {
28             return Response()->json(['status'=>0]);
29         }
30     }
31 }
32
```

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\transaksimodel;
7  use Illuminate\Support\Facades\Validator;
8
9  class transaksicontroller extends Controller
10 {
11     public function store(Request $req)
12     {
13         $validator=Validator::make($req->all(),
14             [
15                 'kode_barang'=>'required',
16                 'id_customer'=>'required'
17             ]
18         );
19         if ($validator->fails()) {
20             return Response()->json($validator->errors());
21         }
22
23         $simpan=transaksimodel::create([
24             'kode_barang'=>$req->kode_barang,
25             'id_customer'=>$req->id_customer
26         ]);
27         if ($simpan) {
28             return Response()->json(['status'=>1]);
29         }else {
30             return Response()->json(['status'=>0]);
31         }
32     }
33 }
```

Use App\nama_model; -> untuk mengconect kan ke kelas model(App->karena berada pada folder App)

Validator ->validasi

Karena data melalui post maka menggunakan parameter untung menangkap nilai inputan.

Jika memang harus di input atau diisi maka ‘required’

Jika gagal maka error tampil berupa json, jika tidak error maka akan melakukan perintah selanjutnya yaitu simpan

Memanggil nama model, bisa menggunakan create atau menggunakan insert.

Menggunakan \$req -> karena semua inputan masuk di \$req.

Bagian kanan adalah name inputan dan sebelah kiri berdasarkan database.

Fungsi logika untuk menandai apakah proses tersebut sukses atau tidak.

Buka routes\api.php untuk mengatur url nya, lalu edit seperti dibawah ini

```
19
20 Route::post('/simpan_barang','barangcontroller@store');
21
22 Route::post('/simpan_customer','customercontroller@store');
23
24 Route::post('/simpan_transaksi','transaksicontroller@store');
25
```

Masukkan nama endpoint bisa apa saja. Selanjutnya memanggil controller.

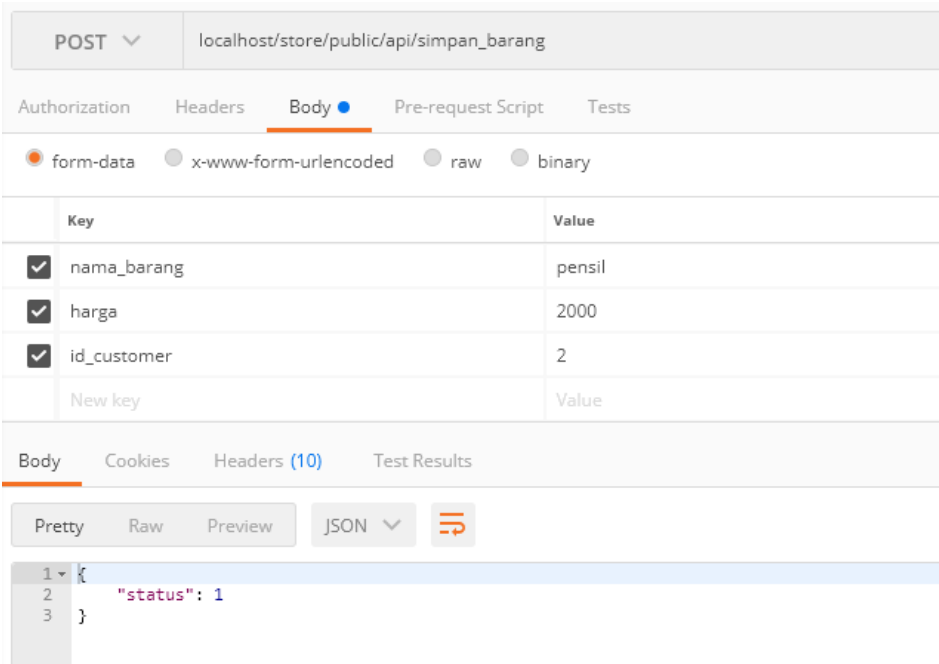
Buka postman dan ketik url:

localhost/nama_folder/public/api/nama_endpoint

Ganti method nya dengan POST

Isikan nama kolom sesuai kolom yang ada di database pada table barang,customer dan transaksi dan datanya. .

Jangan lupa centang tab body lalu pilih form-data



POST

localhost/store/public/api/simpan_customer

Authorization

Headers

Body

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

	Key	Value
<input checked="" type="checkbox"/>	nama	Annisa
	New key	Value

Body

Cookies

Headers (10)

Test Results

Pretty

Raw

Preview

JSON

1

2

3

{

"status": 1

}

POST

localhost/store/public/api/simpan_transaksi

Authorization

Headers

Body

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

	Key	Value	Description
<input checked="" type="checkbox"/>	kode_barang	2	
<input checked="" type="checkbox"/>	id_customer	1	
	New key	Value	Description

Body

Cookies

Headers (10)

Test Results

Pretty

Raw

Preview

JSON

1

2

3

{

"status": 1

}