



PERTEMUAN 7

LOGIN FORM DAN CRUD

1. Membuat Login

a. Membuat model Member

Buat sebuah file dengan nama **Member.php** pada folder *Models* dan ketikkan kode berikut

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Member extends Model
{
    protected $table = 'member';

    public $timestamps = false;
}
```

b. Membuat model Login

Buat sebuah file dengan nama **Login.php** pada folder “*app/Models*” dan ketikkan kode berikut

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Login extends Model
{
    protected $table = 'member_token';

    protected $fillable = ['member_id', 'auth_key'];

    public $timestamps = false;
}
```

c. Membuat controller Login

Buat sebuah file dengan nama **LoginController** pada folder “app/Http/Controllers” dan ketikkan kode berikut

```
<?php

namespace App\Http\Controllers;

use App\Models>Login;
use App\Models\Member;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;

class LoginController extends Controller
{
    public function login(Request $request)
    {
        $email = $request->input('email');
        $password = $request->input('password');

        $member = Member::query()->firstWhere(['email' => $email]);
        if ($member == null) {
            return $this->responseHasil(400, false, 'Email tidak ditemukan');
        }
        if (!Hash::check($password, $member->password)) {
            return $this->responseHasil(400, false, 'Password tidak valid');
        }
    }
}
```

```
    return $this->responseHasil(400, false, 'Password tidak valid');
}
$login = Login::create([
    'member_id' => $member->id,
    'auth_key' => $this->RandomString(),
]);
if (!$login) {
    return $this->responseHasil(401, false, 'Unauthorized');
}
$data = [
    'token' => $login->auth_key,
    'user' => [
        'id' => $member->id,
        'email' => $member->email,
    ]
];
return $this->responseHasil(200, true, $data);
}
```



```
return $this->responseHasil(200, true, $data);  
}  
  
private function RandomString($length = 100)  
{  
    $karakkter =  
'012345678dssd9abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';  
    $panjang_karakter = strlen($karakkter);  
    $str = "";  
    for ($i = 0; $i < $length; $i++) {  
        $str .= $karakkter[rand(0, $panjang_karakter - 1)];  
    }  
    return $str;  
}  
}
```

d. Menambahkan route Login

Pada route tambahkan kode untuk mengakses login sehingga menjadi sebagai berikut

```
Srouter->post('/registrasi', ['uses' => 'RegistrasiController@registrasi']);  
Srouter->post('/login', ['uses' => 'LoginController@login']);
```

e. Mencoba Rest

Silahkan coba Rest API dengan memasukkan url <http://localhost/toko-api/public/login> dengan method POST

The screenshot shows a REST client interface with a POST request to `localhost/toko-api/public/login`. The request body is set to `form-data` and contains the following data:

KEY	VALUE
<input checked="" type="checkbox"/> email	admin@admin.com
<input checked="" type="checkbox"/> password	admin
Key	Value

The response is a 200 OK status with a response time of 689 ms and a size of 468 B. The response body is shown in JSON format:

```
{
  "code": 200,
  "status": true,
  "data": {
    "token": "hjFgkg9Qe4TRuUb1zQgc8yE0wGQ2SzzeD0NYo6C8R2fqvVtc45wcweKcCmYUzTXT54smeVVRfkoRXLDjuYKLfd2ZYddqYrJtfgq",
    "user": {
      "id": 6,
      "email": "admin@admin.com"
    }
  }
}
```




2. CRUD Produk

a. Membuat model Produk

Buat sebuah file dengan nama **Produk.php** pada folder “*app/Models*” dan ketikkan kode berikut

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Produk extends Model
{
    protected $table = 'produk';

    protected $fillable = ['kode_produk', 'nama_produk', 'harga'];

    public $timestamps = false;
}
```

b. Membuat controller produk

Buat sebuah file dengan nama **ProdukController** pada folder “app/Http/Controllers” dan ketikkan kode berikut

```
<?php

namespace App\Http\Controllers;

use App\Models\Produk;
use Illuminate\Http\Request;

class ProdukController extends Controller
{

}
```

Selanjutnya pada class **ProdukController** kita akan menambahkan fungsi (function) CRUD produk, yaitu:

1. Membuat fungsi create produk

```
public function create(Request $request)
{
    $kodeProduk = $request->input('kode_produk');
    $namaProduk = $request->input('nama_produk');
    $harga = $request->input('harga');

    $produk = Produk::create([
        'kode_produk' => $kodeProduk,
        'nama_produk' => $namaProduk,
        'harga' => $harga,
    ]);
    return $this->responseHasil(200, true, $produk);
}
```

2. Membuat fungsi list produk

```
public function list()  
{  
    $produk = Produk::all();  
    return $this->responseHasil(200, true, $produk);  
}
```


3. Membuat fungsi tampil produk

```
public function show($id)
{
    $produk = Produk::findOrFail($id);
    return $this->responseHasil(200, true, $produk);
}
```

4. Membuat fungsi update produk

```
public function update(Request $request, $id)
{
    $kodeProduk = $request->input('kode_produk');
    $namaProduk = $request->input('nama_produk');
    $harga = $request->input('harga');

    $produk = Produk::findOrFail($id);
    $result = $produk->update([
        'kode_produk' => $kodeProduk,
        'nama_produk' => $namaProduk,
        'harga' => $harga,
    ]);
    return $this->responseHasil(200, true, $result);
}
```

5. Membuat fungsi delete produk

```
public function delete($id)
{
    $produk = Produk::findOrFail($id);
    $delete = $produk->delete();
    return $this->responseHasil(200, true, $delete);
}
```

Sehingga adapun keseluruhan kode **ProdukController.php** adalah sebagai berikut

```
<?php

namespace App\Http\Controllers;

use App\Models\Produk;
use Illuminate\Http\Request;

class ProdukController extends Controller
{
    public function create(Request $request)
    {
        $kodeProduk = $request->input('kode_produk');
        $namaProduk = $request->input('nama_produk');
        $harga = $request->input('harga');

        $produk = Produk::create([
            'kode_produk' => $kodeProduk,
            'nama_produk' => $namaProduk,
            'harga' => $harga,
        ]);
        return $this->responseHasil(200, true, $produk);
    }
}
```

```
public function list()
{
    $produk = Produk::all();
    return $this->responseHasil(200, true, $produk);
}

public function show($id)
{
    $produk = Produk::findOrFail($id);
    return $this->responseHasil(200, true, $produk);
}

public function update(Request $request, $id)
{
    $kodeProduk = $request->input('kode_produk');
    $namaProduk = $request->input('nama_produk');
    $harga = $request->input('harga');

    $produk = Produk::findOrFail($id);
    $result = $produk->update([
        'kode_produk' => $kodeProduk,
        'nama_produk' => $namaProduk,
        'harga' => $harga,
    ]);
    return $this->responseHasil(200, true, $result);
}
```



```
public function delete($id)
{
    $produk = Produk::findOrFail($id);
    $delete = $produk->delete();
    return $this->responseHasil(200, true, $delete);
}
```

6. Menambahkan route produk

Agar `ProdukController` dapat diakses, selanjutnya tambah route untuk mengakses produk pada file “`routes/web.php`”

```
$router->group(['prefix' => 'produk'], function ($router) {  
    $router->post('/', ['uses' => 'ProdukController@create']);  
    $router->get('/', ['uses' => 'ProdukController@list']);  
    $router->get('/{id}', ['uses' => 'ProdukController@show']);  
    $router->post('/{id}/update', ['uses' => 'ProdukController@update']);  
    $router->delete('/{id}', ['uses' => 'ProdukController@delete']);  
});
```

Adapun keseluruhan kode pada “routes/web.php” adalah sebagai berikut

```
<?php

/** @var \Laravel\Lumen\Routing\Router $router */

/*
|-----
| Application Routes
|-----
|
| Here is where you can register all of the routes for an application.
| It is a breeze. Simply tell Lumen the URIs it should respond to
| and give it the Closure to call when that URI is requested.
|
*/

$router->get('/', function () use ($router) {
    return $router->app->version();
});

$router->post('/registrasi', ['uses' => 'RegistrasiController@registrasi']);
$router->post('/login', ['uses' => 'LoginController@login']);
```

```
$router->post('/registrasi', ['uses' => 'RegistrasiController@registrasi']);  
$router->post('/login', ['uses' => 'LoginController@login']);  
  
$router->group(['prefix' => 'produk'], function ($router) {  
    $router->post('/', ['uses' => 'ProdukController@create']);  
    $router->get('/', ['uses' => 'ProdukController@list']);  
    $router->get('/{id}', ['uses' => 'ProdukController@show']);  
    $router->post('/{id}/update', ['uses' => 'ProdukController@update']);  
    $router->delete('/{id}', ['uses' => 'ProdukController@delete']);  
});
```


c. Mencoba Rest

Create Produk (localhost/toko-api/public/produk) dengan method POST

POST localhost/toko-api/public/produk

Body form-data

	KEY	VALUE	...	Bulk Edit
<input checked="" type="checkbox"/>	kode_produk	A001		
<input checked="" type="checkbox"/>	nama_produk	Kulkas Sharp		
<input checked="" type="checkbox"/>	harga	2500000		
	Key	Value		

Body Pretty Raw Preview Visualize JSON

```
1 {  
2   "code": 200,  
3   "status": true,  
4   "data": {  
5     "kode_produk": "A001",  
6     "nama_produk": "Kulkas Sharp",  
7     "harga": "2500000",  
8     "id": 1  
9   }  
10 }
```


List Produk (localhost/toko-api/public/produk) dengan method GET

list

GET localhost/toko-api/public/produk

Params

Query Params

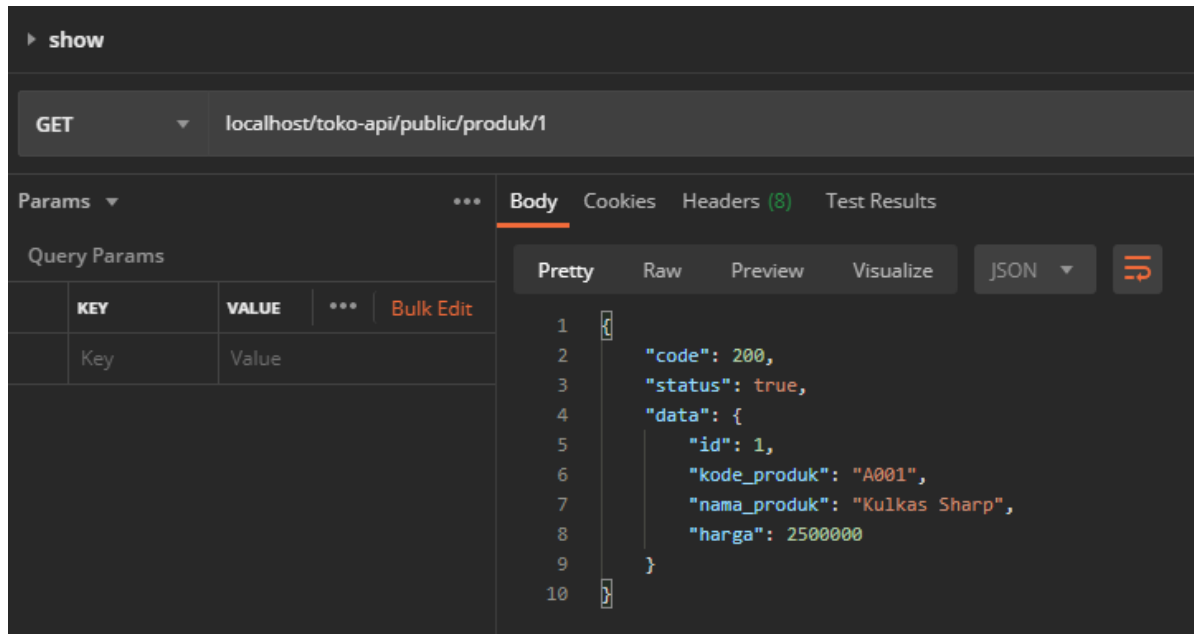
KEY	VALUE
Key	Value

Body

Pretty Raw Preview Visualize JSON

```
1 {
2   "code": 200,
3   "status": true,
4   "data": [
5     {
6       "id": 1,
7       "kode_produk": "A001",
8       "nama_produk": "Kulkas Sharp",
9       "harga": 2500000
10    },
11    {
12      "id": 2,
13      "kode_produk": "A002",
14      "nama_produk": "AC Panasonic",
15      "harga": 3000000
16    }
17  ]
18 }
```

Show Produk (localhost/toko-api/public/produk/{id}) dengan method GET



show

GET localhost/toko-api/public/produk/1

Params ▾ ... Body Cookies Headers (8) Test Results

Query Params

KEY	VALUE	...	Bulk Edit
Key	Value		

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "code": 200,
3   "status": true,
4   "data": {
5     "id": 1,
6     "kode_produk": "A001",
7     "nama_produk": "Kulkas Sharp",
8     "harga": 2500000
9   }
10 }
```

Update Produk (localhost/toko-api/public/produk/{id}/update) dengan method POST

► update

POST localhost/toko-api/public/produk/1/update

Body ▾ ...

form-data ▾

	KEY	VALUE	...	Bulk Edit
<input checked="" type="checkbox"/>	kode_produk	A001		
<input checked="" type="checkbox"/>	nama_produk	Kulkas Polytron		
<input checked="" type="checkbox"/>	harga	2000000		
	Key	Value		

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize

```
1 {  
2   "code": 200,  
3   "status": true,  
4   "data": true  
5 }
```

Delete Produk (localhost/toko-api/public/produk) dengan method DELETE

