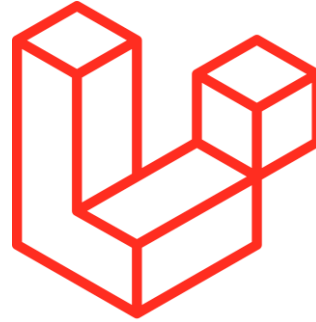


Pemrograman Backend



Pesantren PeTIK II YBM PLN

Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,
Kec. Jombang, Kabupaten Jombang, Jawa Timur



14. Membangun Web Service REST API (Bagian 2)





Membuat Operasi CRUD Web Service REST API

- CRUD adalah singkatan dari create, read, update, and delete. Keempat poin ini merupakan fungsi-fungsi utama yang diimplementasikan dalam aplikasi yang terkoneksi ke sebuah database.
- Akronim CRUD mengidentifikasi semua fungsi utama yang melekat pada database relasional dan aplikasi yang digunakan untuk mengelolanya, termasuk Oracle Database, Microsoft SQL Server, MySQL, PostgreSQL dan database lainnya.
- Seperti halnya Laravel, pada Lumen kita juga dapat membuat operasi CRUD. Perbedaannya di Laravel untuk operasi CRUD-nya, kita menggunakan web browser. Pada Lumen kita gunakan software Postman.
- Postman adalah sebuah aplikasi yang berfungsi sebagai REST CLIENT untuk uji coba REST API. Postman biasa digunakan oleh developer pembuat API sebagai tools untuk menguji API yang telah mereka buat.





Membuat Model

Pada Lumen tidak ada perintah/command untuk membuat model, kita membuatnya secara manual. Buatlah file baru secara manual dengan nama Produk.php di dalam folder app/Models dan silahkan masukkan kode berikut ini :

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Produk extends Model
```

```
{
```

```
    //mapping ke tabel produk
```

```
    protected $table = 'produk';
```

```
    //mapping ke field/kolom pada tabel produk
```

```
    protected $fillable = [
```

```
        'namaProduk', 'deskripsiProduk', 'hargaProduk', 'kategoriProduk',
```

```
    ];
```



```
}
```





Membuat Controller

Setelah berhasil membuat model baru, sekarang kita lanjutkan membuat sebuah controller baru, buatlah file baru dengan nama `ProdukController.php` di dalam folder `app/Http/Controllers`. Buat fungsi `index` untuk menampilkan data seperti kode berikut ini :

```
app > Http > Controllers >  ProdukController.php >  ProdukController
1  <?php
2  namespace App\Http\Controllers;
3  use App\Models\Produk;
4  use Illuminate\Http\Request;
5  use Illuminate\Support\Facades\Validator;
6  class ProdukController extends Controller
7  {
8      public function index(){
9          $produk = Produk::all();
10         return response()->json([
11             'success' => true,
12             'message' => 'List Semua Produk',
13             'data'    => $produk], 200);
14     }
15 }
```

Menambahkan Routing Baru

Selanjutnya tambahkanlah routing baru dengan method **get** untuk menjalankan **fungsi index** yang berada di Produkcontroller, buka file **routes/web.php** lalu tambahkan kode di bawah ini:

```
$router->get('/produk', 'ProdukController@index');
```

Pada Lumen routingnya sedikit berbeda dengan Laravel dan pada Lumen tidak bisa menggunakan **Route::resource** seperti halnya Laravel.



Menjalankan Server

Setelah kita menambahkan route baru, sekarang kita bisa menguji aplikasi kita untuk menampilkan data dari database. Jalankan server project kita melalui terminal/CMD, silahkan masuk ke direktori project kita, lalu jalankan perintah di bawah ini :

```
php -S localhost:8000 -t public
```

Pada Lumen kita tidak bisa menjalankan perintah:

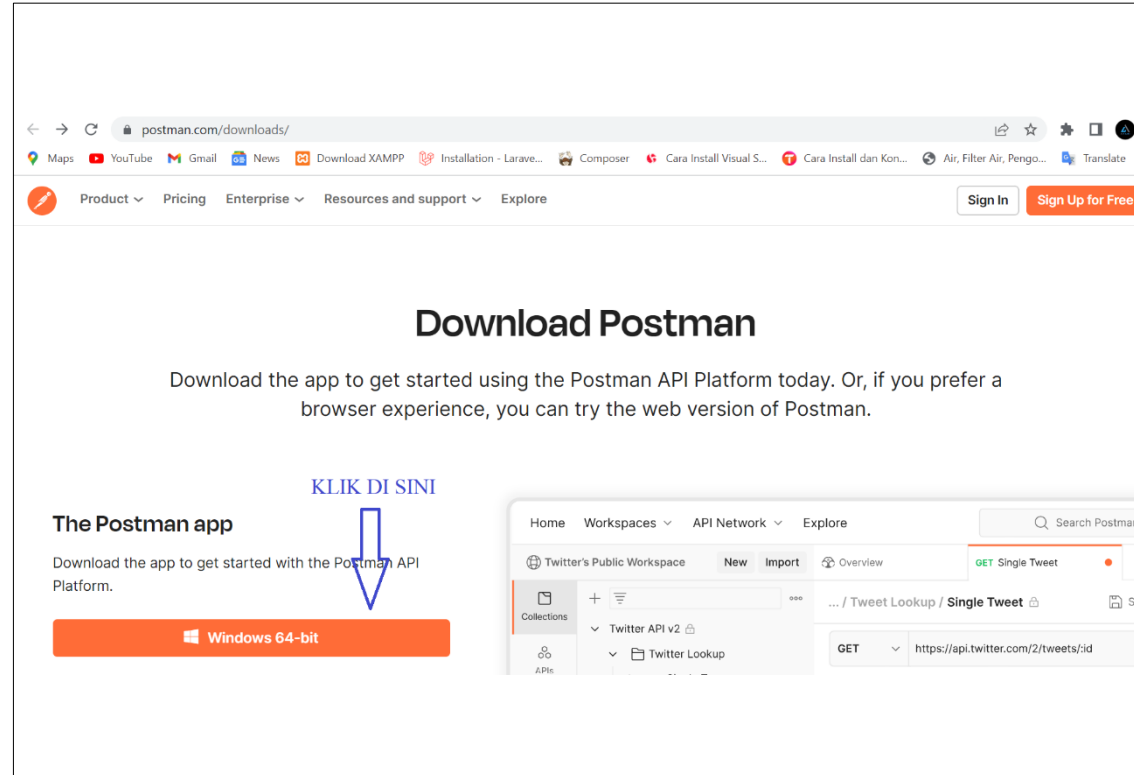
```
php artisan serve
```





INSTALL SOFTWARE POSTMAN

Software bisa di Download di link berikut : <https://www.postman.com/downloads/>



postman.com/downloads/

Product Pricing Enterprise Resources and support Explore

Sign In Sign Up for Free

Download Postman

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

KLIK DI SINI

The Postman app

Download the app to get started with the Postman API Platform.

Windows 64-bit

Home Workspaces API Network Explore

Twitter's Public Workspace New Import Overview

GET Single Tweet

Search Postman

... / Tweet Lookup / Single Tweet

GET https://api.twitter.com/2/tweets/id



INSTALL SOFTWARE POSTMAN



Cara pemakaian Postman, perhatikan gambar dibawah ini :



Create an account or sign in

Create Free Account

Sign in

KLIK DI SINI



Create your account or sign in later? [Skip and go to the app](#)

A free Postman account lets you

- ✓ Organize all your API development in workspaces
- ✓ Create public workspaces to collaborate with over 10 million developers
- ✓ Back up your work on Postman's cloud
- ✓ Experience the best API development platform for free!





INSTALL SOFTWARE POSTMAN



Cara pemakaian Postman, perhatikan gambar dibawah ini :



Create an account or sign in

Create Free Account

Sign in

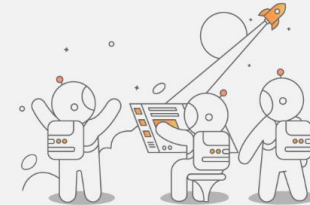
KLIK DI SINI



Create your account or sign in later? [Skip and go to the app](#)

A free Postman account lets you

- ✓ Organize all your API development in workspaces
- ✓ Create public workspaces to collaborate with over 10 million developers
- ✓ Back up your work on Postman's cloud
- ✓ Experience the best API development platform for free!





INSTALL SOFTWARE POSTMAN



Klik new pada postman, untuk membuat workspace baru

Home Workspaces Explore Search Postman Sign In Create Account

Scratch Pad New Import

Collections + New Collection This collection is empty Add a request to start working.

APIs Environments Mock Servers Monitors History

Overview POST http://localhost:8000/ No Environment

http://localhost:8000/produk Save

POST Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Response

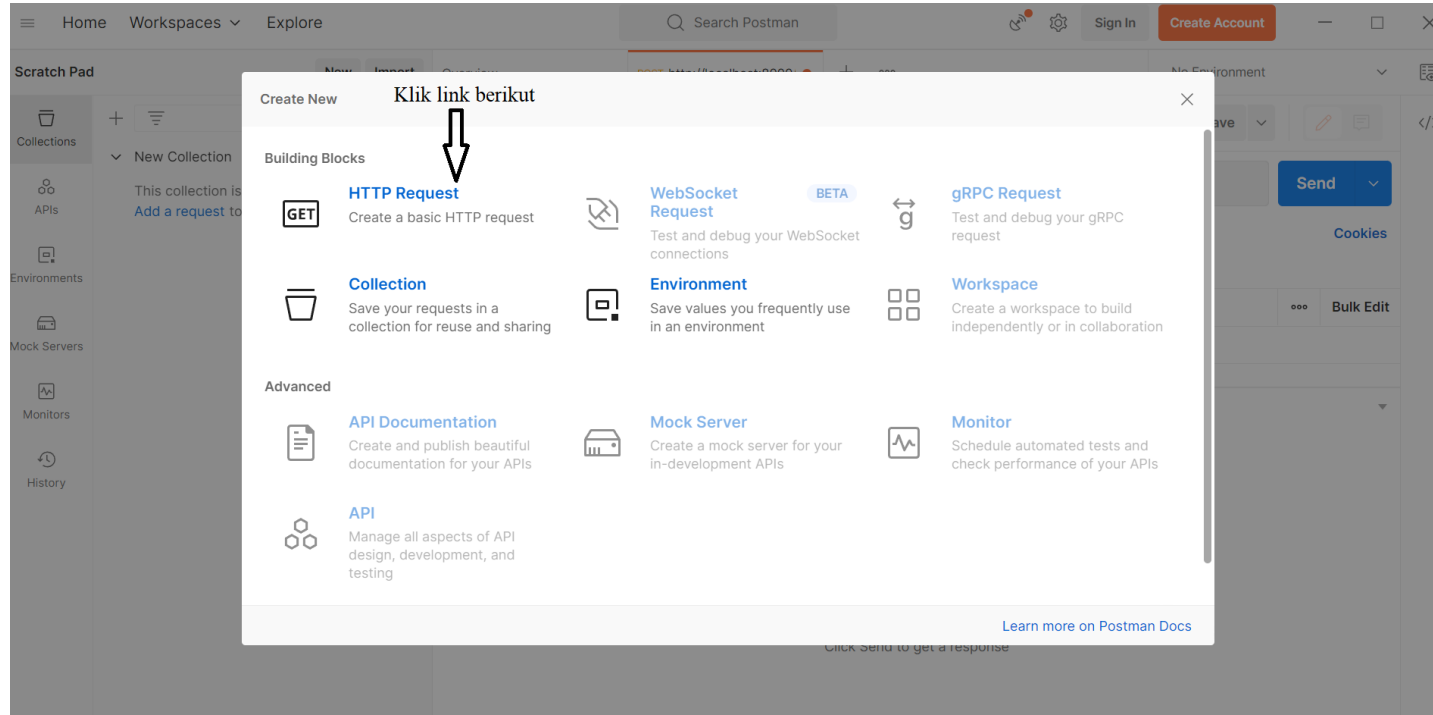
Click Send to get a response



INSTALL SOFTWARE POSTMAN



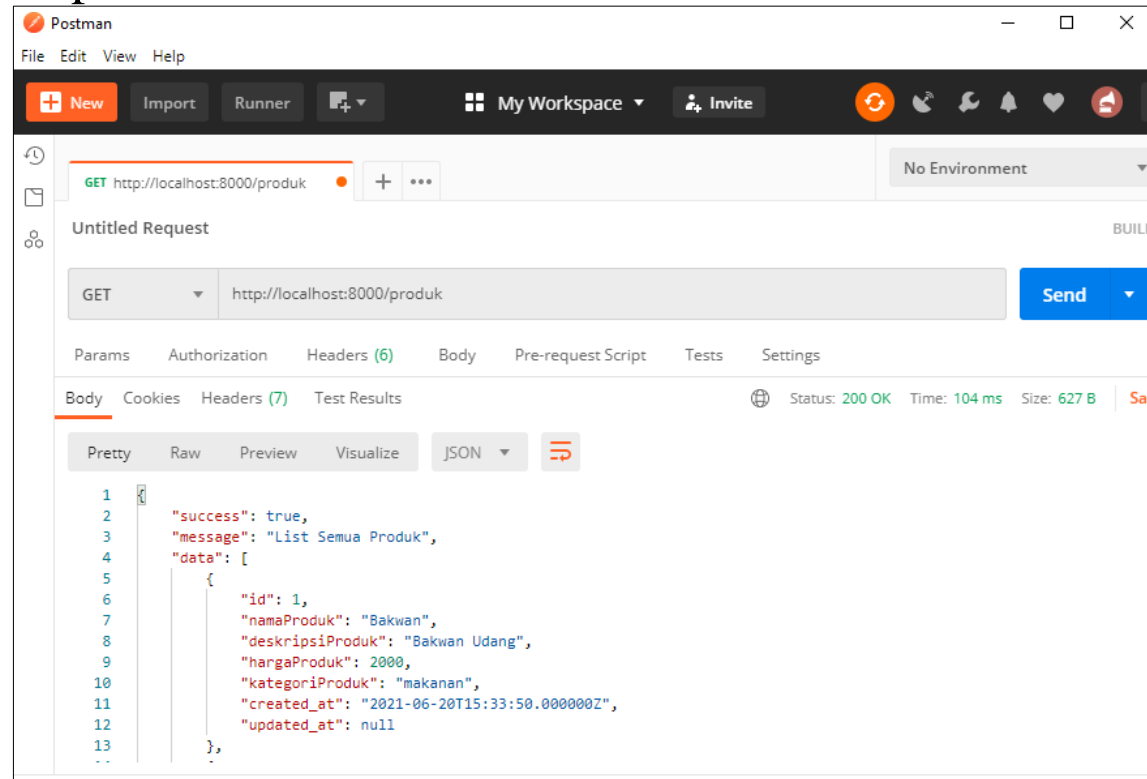
Pilih HTTP Request





Tampilan Data REST API

Sekarang kita bisa mencoba aplikasinya menggunakan Postman, silahkan masukkan `http://localhost:8000/produk`. Gunakan method **GET** setelah itu tekan tombol **SEND**.








Input Data REST API(1)

Selanjutnya kita akan input data dengan REST API di Lumen. Silahkan buka file PegawaiController.php di dalam folder app/Http/Controllers dan silahkan tambahkan fungsi store tepat di bawah fungsi index seperti kode program di bawah ini:

```
app > Http > Controllers >  ProdukController.php >  ProdukController
17
18 | public function store(Request $request)
19 | {
20 |     $validator = Validator::make($request->all(), [
21 |         'namaProduk' => 'required',
22 |         'deskripsiProduk' => 'required',
23 |         'hargaProduk' => 'required',
24 |         'kategoriProduk' => 'required',
25 |     ]);
26 |     if ($validator->fails()) {
27 |         return response()->json([
28 |             'success' => false,
29 |             'message' => 'Semua Kolom Wajib Diisi!',
30 |             'data' => $validator->errors()
31 |         ],401);
32 |     }
```



Input Data REST API(2)

```
app > Http > Controllers >  ProdukController.php >  ProdukController >  store
33 //-----lanjutan slide sebelumnya-----
34 else {
35     $produk = Produk::create([
36         'namaProduk' => $request->input('namaProduk'),
37         'deskripsiProduk' => $request->input('deskripsiProduk'),
38         'hargaProduk' => $request->input('hargaProduk'),
39         'kategoriProduk' => $request->input('kategoriProduk'),
40     ]);
41     if ($produk) {
42         return response()->json([
43             'success' => true,
44             'message' => 'Produk Berhasil Disimpan!',
45             'data' => $produk
46         ], 201);
47     } else {
48         return response()->json([
49             'success' => false,
50             'message' => 'Produk Gagal Disimpan!',
51         ], 400);
52     }
}
```





Menambahkan Routing Baru dan Menjalankan Server

Selanjutnya tambahkanlah routing baru dengan method **post** untuk menjalankan **fungsi store** yang berada di Produkcontroller, buka file **routes/web.php** lalu tambahkan kode di bawah ini:

```
$router->post('/input_produk', 'ProdukController@store');
```

Setelah kita menambahkan route baru, sekarang kita bisa menguji aplikasi kita. Jalankan server project kita melalui terminal/CMD, silahkan masuk ke direktori project kita, lalu jalankan perintah di bawah ini :

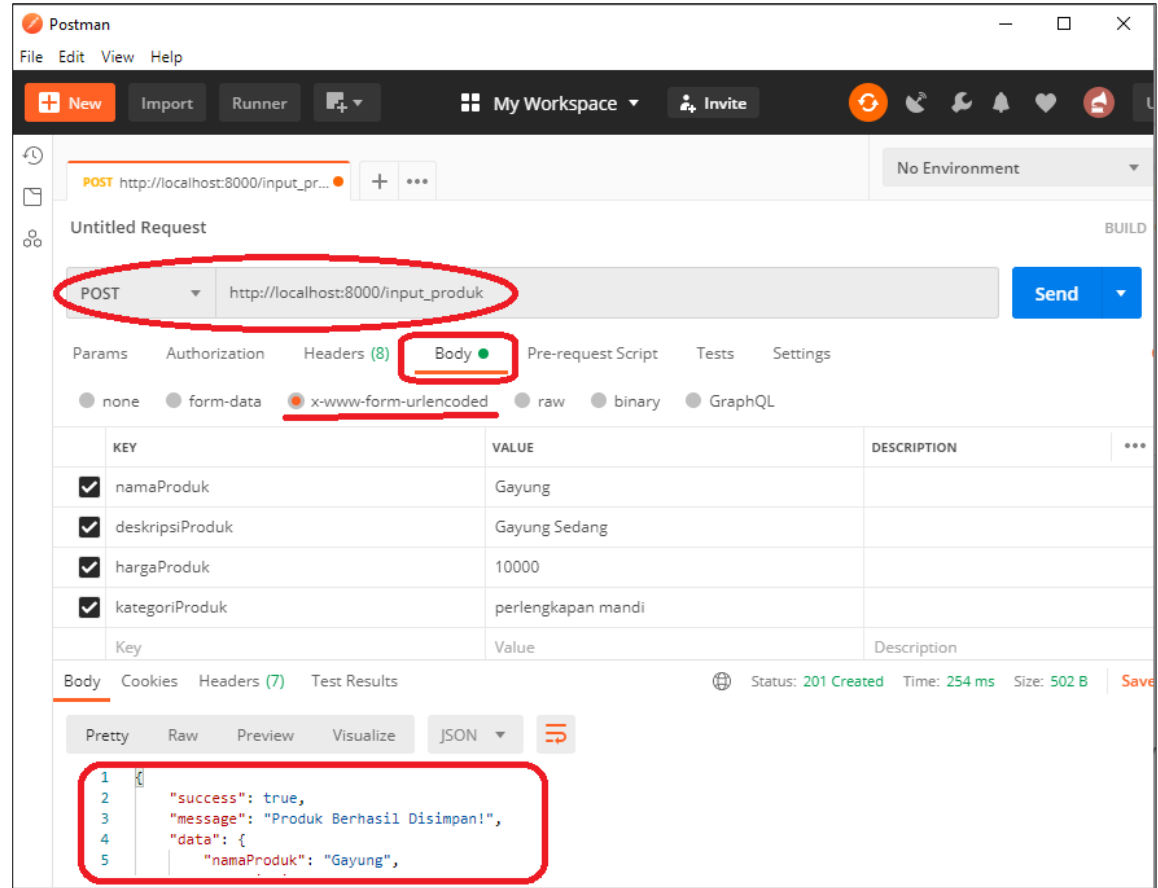
```
php -S localhost:8000 -t public
```





Tampilan Input Data REST API

- Sekarang kita bisa coba input data dengan REST API menggunakan Postman.
- Silahkan buka Postman, lalu ketikkan: `http://localhost:8000/input_produk` dan gunakan method **POST**.
- Lalu input data sesuai kolom/field di tabel produk masuk ke tab **Body** dan **x-www-form-urlencoded**, lalu masukkan berdasarkan nama field di bagian **KEY** dan input data di bagian **VALUE**.
- Setelah itu tekan tombol **SEND**.
- Setelah berhasil, di bagian paling bawah pada Postman ada keterangan Produk Berhasil Disimpan dan keterangan kolom-kolom dan isi datanya.





Menampilkan Data Berdasarkan ID

Selanjutnya kita akan tampilkan data berdasarkan ID dengan REST API di Lumen. Silahkan buka file PegawaiController.php di dalam folder app/Http/Controllers dan silahkan tambahkan fungsi show(\$id) tepat di bawah fungsi store() seperti kode program di bawah ini:

```
app > Http > Controllers >  ProdukController.php >  ProdukController

56 | public function show($id)
57 | {
58 |     $produk = Produk::find($id);
59 |     if ($produk) {
60 |         return response()->json([
61 |             'success' => true,
62 |             'message' => 'Detail Produk!',
63 |             'data'    => $produk
64 |         ], 200);
65 |     } else {
66 |         return response()->json([
67 |             'success' => false,
68 |             'message' => 'Produk Tidak Ditemukan!',
69 |         ], 404);
70 |     }
71 | }
```



Menambahkan Routing Baru dan Menjalankan Server

Selanjutnya tambahkanlah routing baru dengan method **get** untuk menjalankan **fungsi show** yang berada di **Produkcontroller**, buka file **routes/web.php** lalu tambahkan kode di bawah ini:

```
$router->get('/produk/{id}', 'ProdukController@show');
```

Setelah kita menambahkan route baru, sekarang kita bisa menguji aplikasi kita. Jalankan server project kita melalui terminal/CMD, silahkan masuk ke direktori project kita, lalu jalankan perintah di bawah ini :

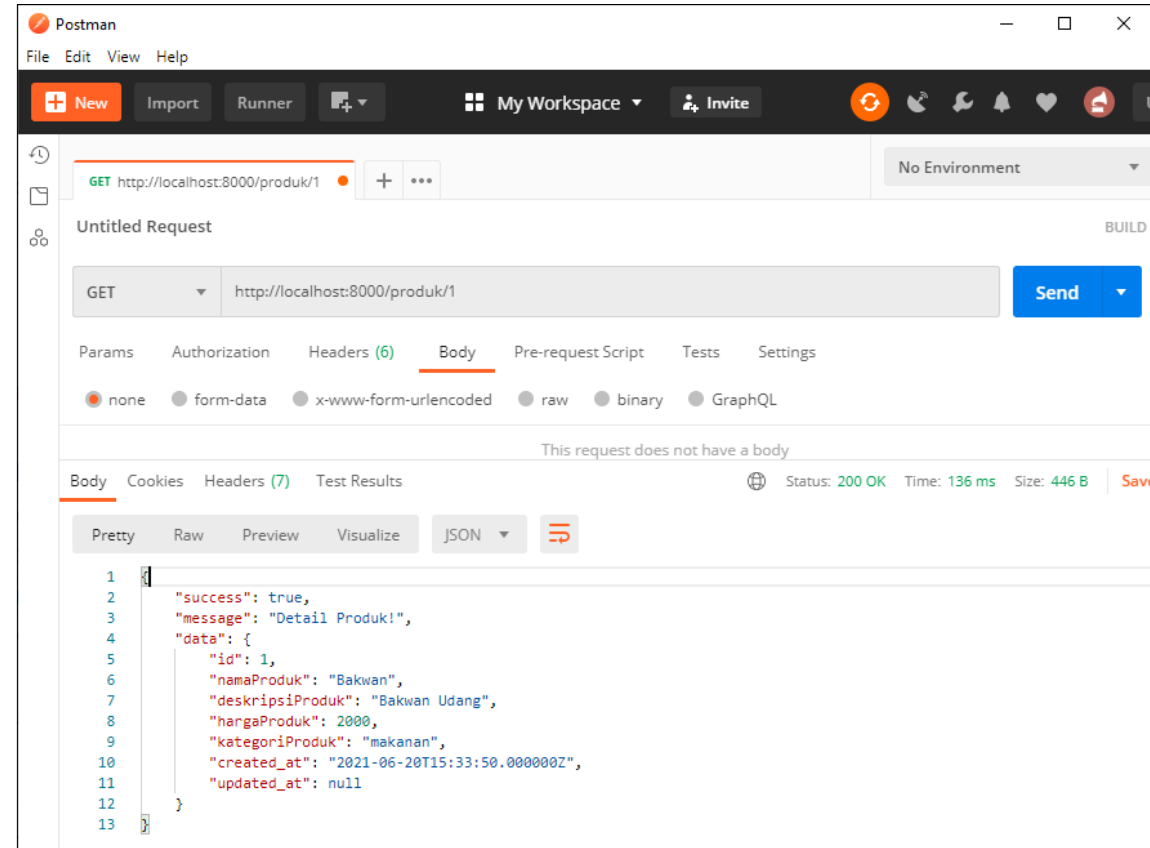
```
php -S localhost:8000 -t public
```





Tampilan Data Berdasarkan ID REST API

- Sekarang kita akan tampilkan data berdasarkan ID dengan REST API menggunakan Postman.
- Silahkan buka Postman, lalu ketikkan:
`http://localhost:8000/produk/1` dan gunakan method **GET**.
- Setelah itu tekan tombol **SEND**.
- Setelah berhasil, di bagian paling bawah pada Postman ada keterangan Detail Produk dan data produk berdasarkan ID yang dipilih.





Update Data REST API(1)






Selanjutnya kita akan ubah data dengan REST API di Lumen. Silahkan buka file PegawaiController.php di dalam folder app/Http/Controllers dan silahkan tambahkan fungsi update tepat di bawah fungsi show seperti kode program di bawah ini:

```
app > Http > Controllers > ProdukController.php > ProdukController > update
73 public function update(Request $request, $id)
74 {
75     $validator = Validator::make($request->all(), [
76         'namaProduk' => 'required',
77         'deskripsiProduk' => 'required',
78         'hargaProduk' => 'required',
79         'kategoriProduk' => 'required',
80     ]);
81     if ($validator->fails()) {
82         return response()->json([
83             'success' => false,
84             'message' => 'Semua Kolom Wajib Diisi!',
85             'data' => $validator->errors()
86         ],401);
87     }
```



Update Data REST API(2)

```
app > Http > Controllers >  ProdukController.php >  ProdukController >  update

88      //lanjutan slide sebelumnya
89      else {
90          $produk = Produk::whereId($id)->update([
91              'namaProduk' => $request->input('namaProduk'),
92              'deskripsiProduk' => $request->input('deskripsiProduk'),
93              'hargaProduk' => $request->input('hargaProduk'),
94              'kategoriProduk' => $request->input('kategoriProduk'),
95          ]);
96          if ($produk) {
97              return response()->json([
98                  'success' => true,
99                  'message' => 'Produk Berhasil Diupdate!',
100                 'data' => $produk
101             ], 201);
102          } else {
103              return response()->json([
104                  'success' => false,
105                  'message' => 'Produk Gagal Diupdate!',
106             ], 400);
107          }
108      }
109  }
```



Menambahkan Routing Baru dan Menjalankan Server

Selanjutnya tambahkanlah routing baru dengan method **put** untuk menjalankan **fungsi update** yang berada di Produkcontroller. Buka file **routes/web.php** lalu tambahkan kode di bawah ini:

```
$router->put('/produk/{id}', 'ProdukController@update');
```

Setelah kita menambahkan route baru, sekarang kita bisa menguji aplikasi kita. Jalankan server project kita melalui terminal/CMD, silahkan masuk ke direktori project kita, lalu jalankan perintah di bawah ini :

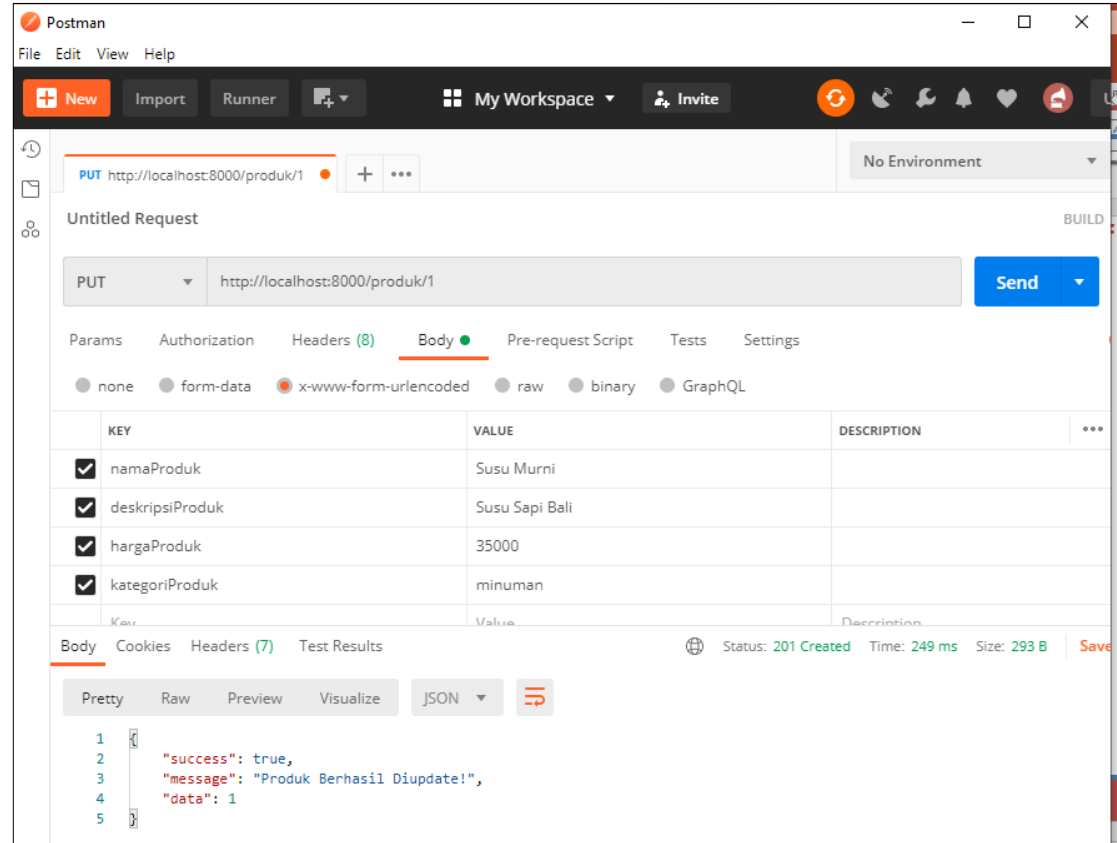
```
php -S localhost:8000 -t public
```





Tampilan Update Data REST API

- Sekarang kita bisa coba input data dengan REST API menggunakan Postman.
- Silahkan buka Postman, lalu ketikkan: `http://localhost:8000/produk/1` dan gunakan method **PUT**.
- Lalu ubah data sesuai kolom/field di tabel produk masuk ke tab **Body** dan **x-www-form-urlencoded**, lalu ubah berdasarkan field di bagian **KEY** dan ubah datanya di bagian **VALUE**.
- Setelah itu tekan tombol **SEND**.
- Setelah berhasil, di bagian paling bawah pada Postman ada keterangan Produk Berhasil Disimpan dan keterangan kolom-kolom dan isi datanya.





Menghapus Data Berdasarkan ID



Terakhir dari operasi CRUD, kita akan menghapus data berdasarkan ID dengan REST API di Lumen. Silahkan buka file PegawaiController.php di dalam folder app/Http/Controllers dan silahkan tambahkan fungsi **destroy(\$id)** tepat di bawah fungsi update seperti kode program di bawah ini:

```
public function destroy($id)
{
    $produk = Produk::whereId($id)->first();
    $produk->delete();
    if ($produk) {
        return response()->json([
            'success' => true,
            'message' => 'Produk Berhasil Dihapus!',
        ], 200);
    }
}
```





Menambahkan Routing Baru dan Menjalankan Server

Selanjutnya tambahkanlah routing baru dengan method **delete** untuk menjalankan fungsi **destroy** yang berada di Produkcontroller. Buka file **routes/web.php** lalu tambahkan kode di bawah ini:

```
$router->delete('/produk/{id}', 'ProdukController@destroy');
```

Setelah kita menambahkan route baru, sekarang kita bisa menguji aplikasi kita. Jalankan server project kita melalui terminal/CMD, silahkan masuk ke direktori project kita, lalu jalankan perintah di bawah ini :

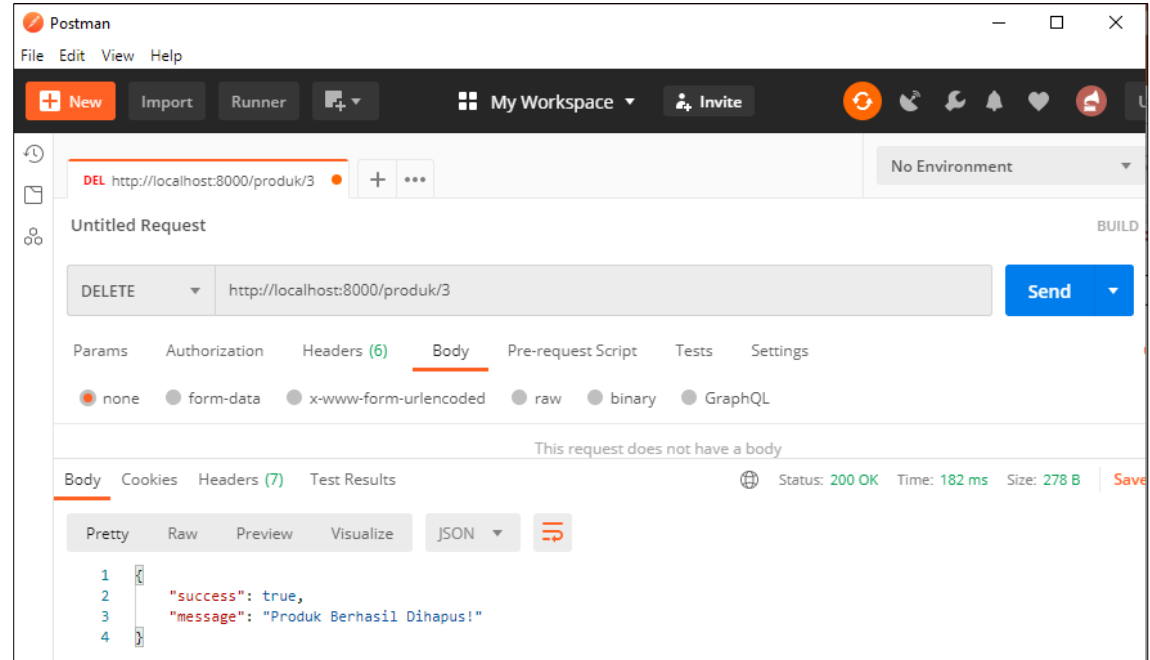
```
php -S localhost:8000 -t public
```





Tampilan Hapus Data Berdasarkan ID

- Sekarang kita akan hapus sebuah data produk berdasarkan ID dengan REST API menggunakan Postman.
- Silahkan buka Postman, lalu ketikkan:
`http://localhost:8000/produk/3` dan gunakan method **DELETE**.
- Setelah itu tekan tombol **SEND**.
- Setelah berhasil, di bagian paling bawah pada Postman ada keterangan Produk Berhasil Dihapus.



TERIMA KASIH
ATAS SEGALA PERHATIAN
SEMOGA BERMANFAAT...

