**Tugas 15 Mandiri**

Pengembangan Perangkat Lunak





**Pemograman Backend**

Nama : Faiz Haidar

NIM : 22255601008

**PeTIK II Jombang YBM PLN Tahun Akademik 2022/2023**

**Plosogeneng, Kec. Jombang, Kabupaten Jombang, Jawa Timur**

1. Membuat Model Produk pada App/Models dengan nama Produk.php

Isi file Produk.php

<?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',

    ];

}

1. Membuat Controller Produk, pada folder app/http/controller dengan nama ProdukController.php, membuat fungsi index untuk menampilkan data

Isi file ProdukController.php pada fungsi index

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

use App\Models\Produk;

*class* ProdukController *extends* *Controller*

{

*// Menampilkan Produk*

*public* *function* index()

    {

        $produk = Produk::all();

        return response()->json([

            'success' => true,

            'message' => 'List Semua Produk',

            'data' => $produk

        ], 200);

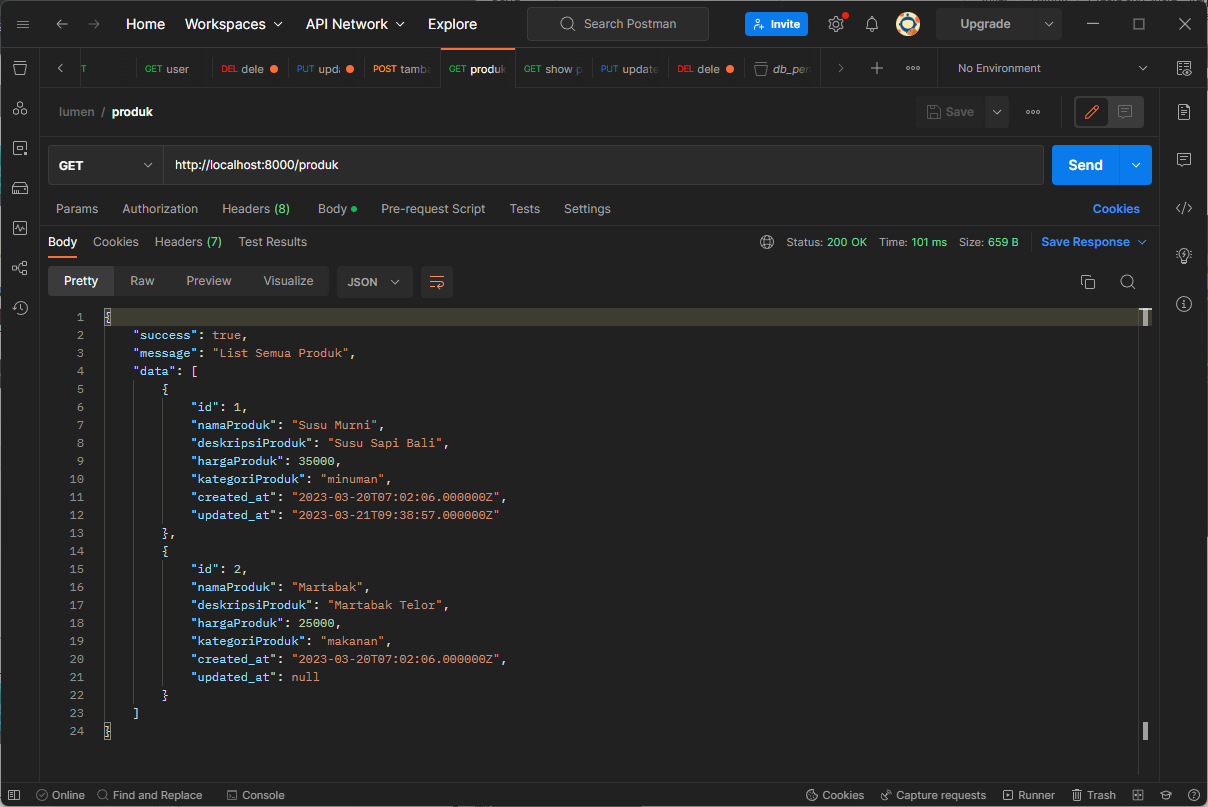
    }

1. Menambah route pada web.php untuk menjalankan fungsi index, seperti berikut

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

1. Menguji aplikasi dengan cara menjalankan server dengan perintah dibawah ini

*php -S localhost:8000 -t public,* lalu buka aplikasi Postman dan masukkan <http://localhost:8000/produk>, gunakan method GET tekan tombol SEND



Maka akan tampil seperti berikut

1. Input data REST API di Lumen, dengan menambahkan fungsi store pada ProdukController.php dengan menuliskan kode dibawah ini

*public* *function* store(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'namaProduk' => 'required',

            'deskripsiProduk' => 'required',

            'hargaProduk' => 'required',

            'kategoriProduk' => 'required',

        ]);

        if ($validator->fails()) {

            return response()->json([

                'success' => true,

                'message' => 'Semua Kolom Wajib Diisi',

                'data' => $validator->errors()

            ], 401);

        } else {

            $produk = Produk::create([

                'namaProduk' => $request->input('namaProduk'),

                'deskripsiProduk' => $request->input('deskripsiProduk'),

                'hargaProduk' => $request->input('hargaProduk'),

                'kategoriProduk' => $request->input('kategoriProduk'),

            ]);

            if ($produk) {

                return response()->json([

                    'success' => true,

                    'message' => 'Produk Berhasil Disimpan',

                    'data' => $produk

                ], 201);

            } else {

                return response()->json([

                    'success' => false,

                    'message' => 'Produk Gagal Disimpan',

                ], 400);

            }

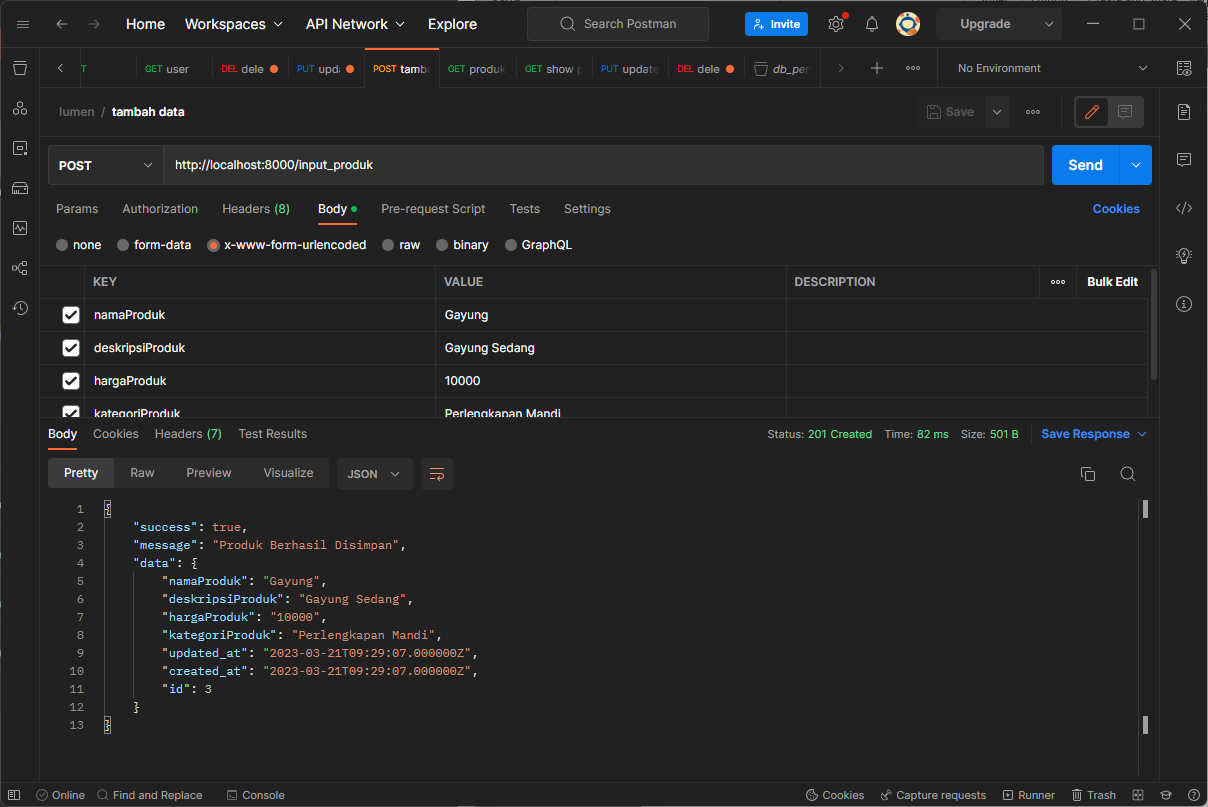
        }

    }

1. Menambah route pada web.php untuk menjalankan fungsi store, seperti berikut

$router->post('/input\_produk', 'ProdukController@store');

1. Sekarang kita bisa input data dengan REST API yaitu masukkan <http://localhost:8000/input_produk>, gunakan method POST dan input data sesuai kolom, masuk ke tab body x-www-form-urlencoded isi bagian pada value, lalu klik tombol SEND



1. Membuat tampilan detail produk dengan menambahkan fungsi show($id) pada file ProdukController.php

*public* *function* show($id)

    {

        $produk = Produk::find($id);

        if ($produk) {

            return response()->json([

                'success' => true,

                'message' => 'Detail Produk',

                'data' => $produk

            ], 200);

        } else {

            return response()->json([

                'success' => false,

                'message' => 'Produk Tidak Ditemukan',

            ], 404);

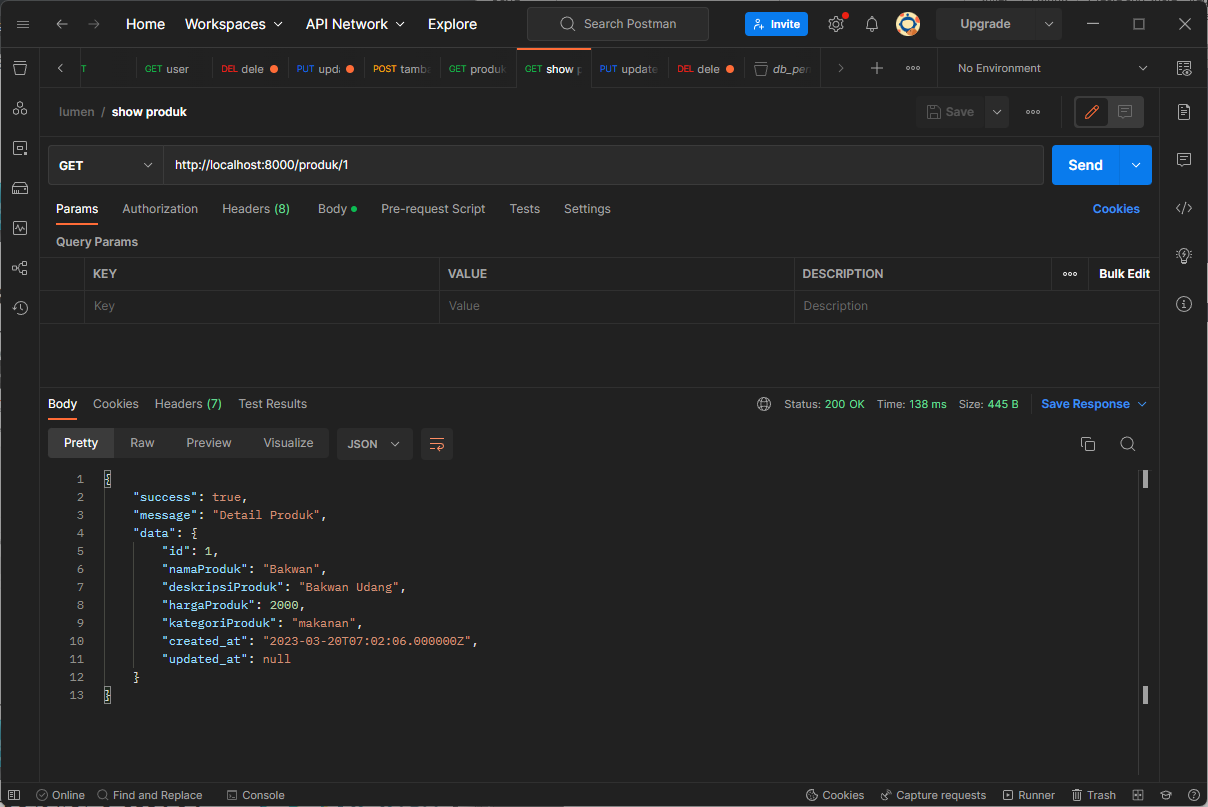
        }

    }

1. Menambahkan route fungsi show pada web.php

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

1. Menampilkan detail produk pada Postman dengan mengetikkan <http://localhost:8000/produk/1,> dan menggunakan method GET lalu klik tombol SEND, maka akan tampil seperti berikut



1. Membuat update data produk dengan menambahkan fungsi update(Request $request, $id) dengan mengisikan kode sebagai barikut

*public* *function* update(Request $request, $id)

    {

        $validator = Validator::make($request->all(), [

            'namaProduk' => 'required',

            'deskripsiProduk' => 'required',

            'hargaProduk' => 'required',

            'kategoriProduk' => 'required',

        ]);

        if ($validator->fails()) {

            return response()->json([

                'success' => true,

                'message' => 'Semua Kolom Wajib Diisi',

                'data' => $validator->errors()

            ], 401);

        } else {

            $produk = Produk::whereId($id)->update([

                'namaProduk' => $request->input('namaProduk'),

                'deskripsiProduk' => $request->input('deskripsiProduk'),

                'hargaProduk' => $request->input('hargaProduk'),

                'kategoriProduk' => $request->input('kategoriProduk'),

            ]);

            if ($produk) {

                return response()->json([

                    'success' => true,

                    'message' => 'Produk Berhasil Diupdate',

                    'data' => $produk

                ], 201);

            } else {

                return response()->json([

                    'success' => false,

                    'message' => 'Produk Gagal Diupdate',

                ], 400);

            }

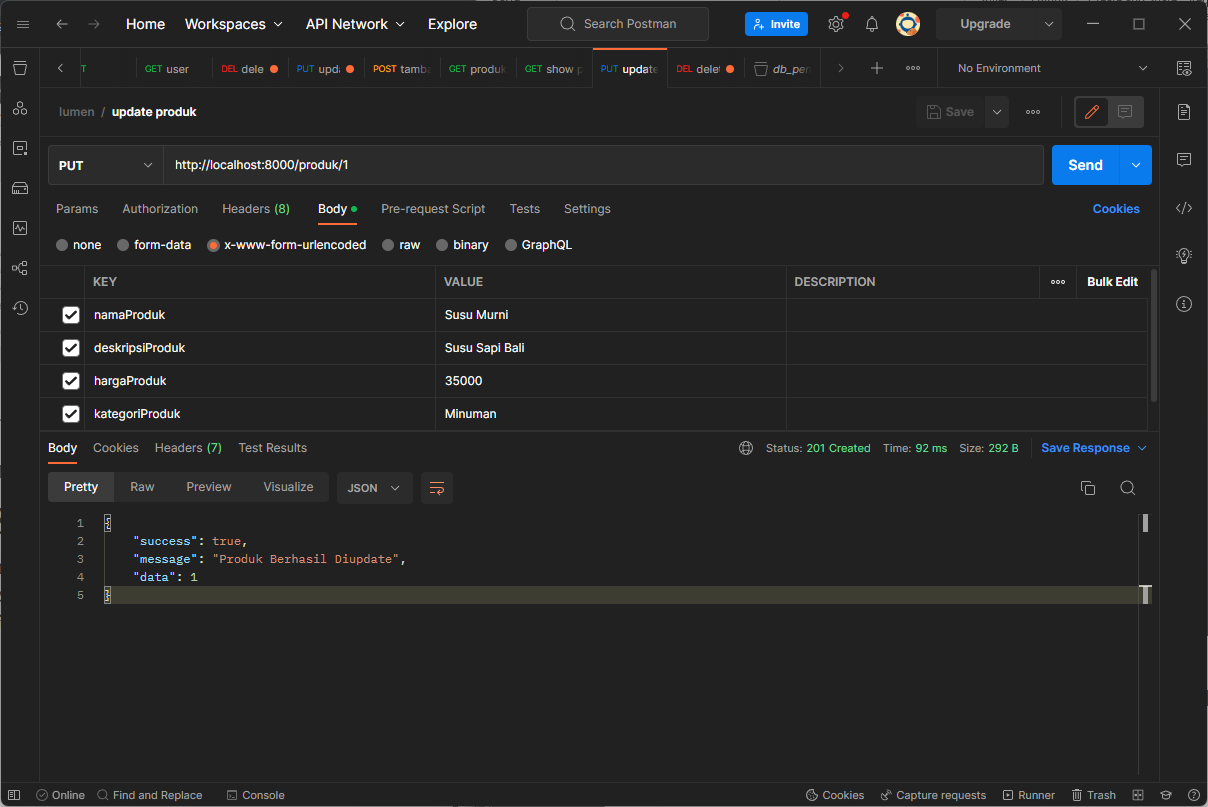
        }

    }

1. Menambahkan route fungsi update pada web.php

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

1. Mengupdate data produk pada REST API menggunakan POSTMAN dengan mengetikkan <http://localhost:8000/produk/1>, lal menggunakan method PUT, ubah kolom denagn masuk ke tab Body *x-www-form-urlencoded* ubah data pada bagian value, maka akan tampil seperti berikut



1. Membuat fitur delete pada REST API dengan menambahkan fungsi destroy($id) pada ProdukController.php dengan kode sebagai berikut

*public* *function* destroy($id)

    {

        $produk = Produk::whereId($id)->first();

        $produk->delete();

        if ($produk) {

            return response()->json([

                'success' => true,

                'message' => 'Produk Berhasil Dihapus!',

            ], 200);

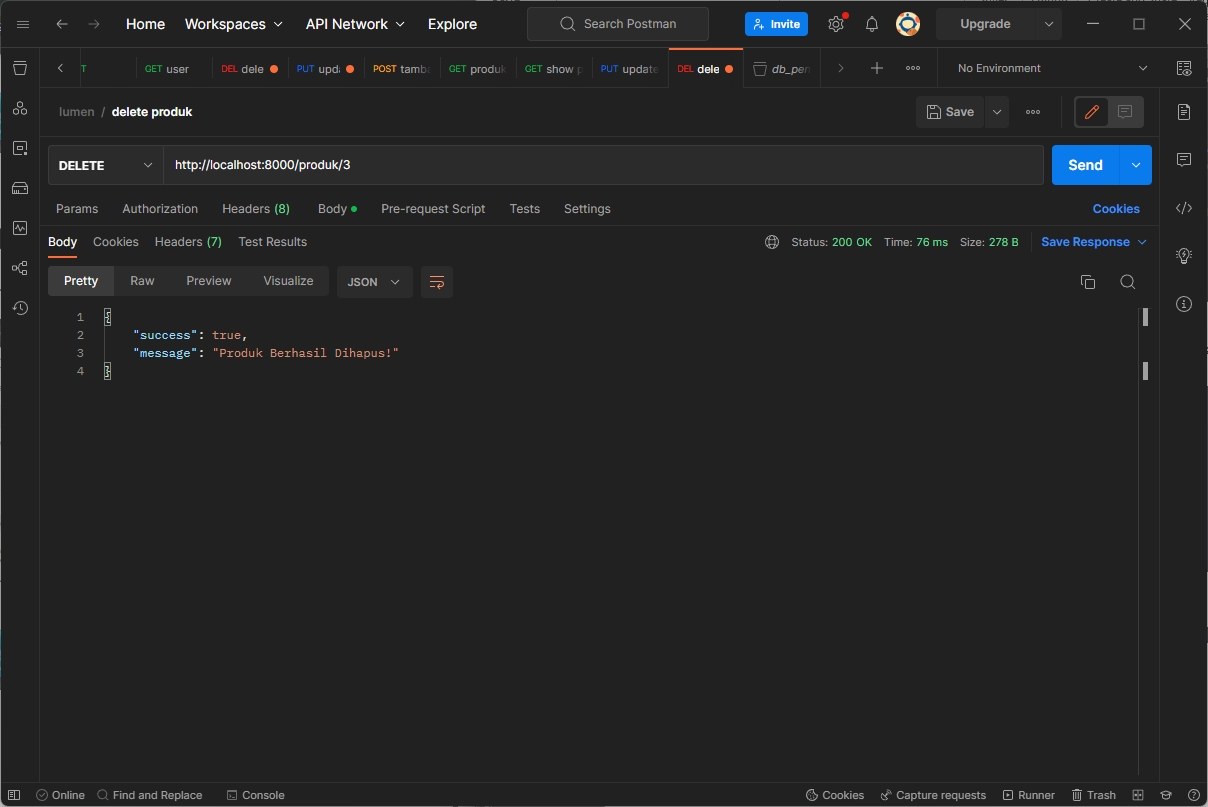
        }

    }

1. Menambahkan Route fungsi destroy pada web.php

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

1. Menghapus data produk dengan mengetikkan <http://localhost:8000/produk/3> menggunakan method DELETE lalu tekan tombol SEND



Maka data pada produk dengan id 3 berhasil terhapus