

PRAKTIK PENGEMBANGAN BACK END

12 November 2024

FUNGSI DELETE DI LARAVEL

a. Apa itu CRUD

CRUD adalah singkatan dari Create, Read, Update, dan Delete, yang merupakan empat fungsi dasar yang digunakan untuk mengelola data dalam basis data dan penting untuk sistem manajemen basis data. Fungsi-fungsi ini sering digunakan sebagai dasar untuk membangun aplikasi web.

Pertimbangkan panel admin atau backend apa pun, CRUD akan menjadi fungsi pentingnya untuk mengelola data.

Membuat / create

Operasi create digunakan untuk **menambahkan data baru ke dalam basis data**. Misalnya, jika Anda memiliki basis data pelanggan dan ingin menambahkan pelanggan baru ke dalam basis data, Anda dapat menggunakan operasi create. Hal ini dapat melibatkan penambahan rekaman baru ke tabel pelanggan dengan nama pelanggan, alamat, dan informasi lainnya.

Membaca / read

Operasi baca digunakan untuk **mengambil data dari basis data**. Misalnya, jika Anda ingin mengambil daftar semua pelanggan dalam basis data, Anda akan menggunakan operasi baca. Ini dapat melibatkan permintaan ke tabel pelanggan untuk mengambil semua rekaman dan menampilkannya di situs web atau aplikasi.

Memperbarui / update

Operasi pembaruan digunakan untuk **mengubah data yang ada dalam basis data**. Misalnya, jika Anda ingin memperbarui alamat pelanggan dalam basis data, Anda akan menggunakan operasi pembaruan. Ini dapat melibatkan perubahan catatan dalam tabel pelanggan dengan informasi alamat baru.

Menghapus / delete

Operasi hapus digunakan untuk **menghapus data dari basis data** . Misalnya, jika Anda ingin menghapus pelanggan dari basis data, Anda akan menggunakan operasi hapus. Ini dapat melibatkan penghapusan rekaman dari tabel pelanggan.

b. Apa Itu Model

Model menjadi salah satu bagian penting dari konsep [MVC](#) (Model-View-Controller) pada framework Laravel yang bertanggung jawab untuk mengatur interaksi antara aplikasi dengan database. Dalam konsep MVC, Model dipadukan dengan Controller dan View untuk membentuk sebuah fitur atau halaman pada aplikasi Laravel kita. Model bertanggung jawab untuk memproses data dari database dan mengembalikan data tersebut ke Controller.

c. Apa itu Controller

Dalam konteks Laravel, controller adalah kelas PHP yang bertanggung jawab untuk memproses permintaan HTTP dari pengguna dan mengembalikan respons yang sesuai. Controller menyatukan logika aplikasi dan memastikan bahwa permintaan dari pengguna diarahkan dengan benar ke model yang sesuai atau ke tampilan yang tepat.

d. Apa itu Routes / routing

Routing adalah salah satu komponen inti dalam aplikasi web yang memungkinkan Anda mendefinisikan alamat URL yang akan dipetakan ke controller tertentu dalam aplikasi Anda. Dalam Laravel, routing digunakan untuk mengarahkan semua request HTTP ke handler / method yang tepat.

Routing di Laravel sangat fleksibel dan memberi Anda kontrol penuh atas setiap request ditangani oleh aplikasi Anda. Ini

memungkinkan struktur yang rapi dan mudah dikelola untuk aplikasi yang kompleks.

e. Konsep MVC (Model, View, Controller)

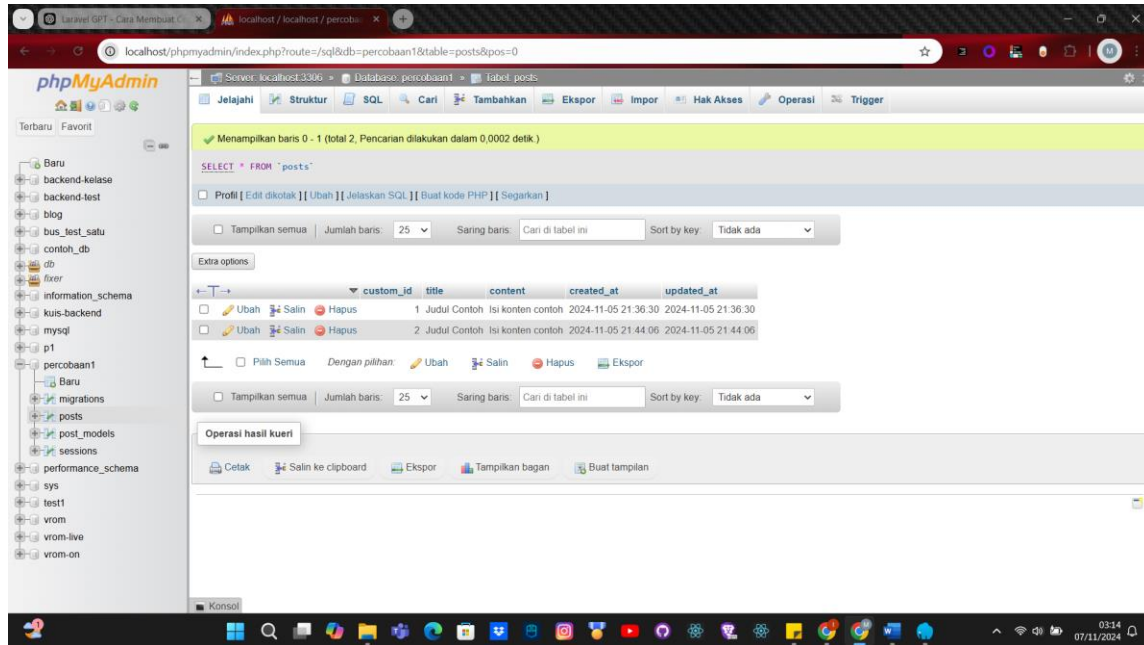
MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, controller, dan user interface.

1. Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. Controller, Controller merupakan bagian yang menjembatani model dan view.

e. Proses Pembuatan Fungsi/Function delete di Laravel

1. Pertama, cek terlebih dahulu apakah table/data yang ingin ditunjukkan pastikan sudah ada, dan di table tersebut terdapat variabelnya :

Contoh :



2. Kemudian pergi Kembali ke controller di laravelnya, dan buatlah function untuk delete, dengan kode ini :

- Function delete berdasarkan id yang dipilih

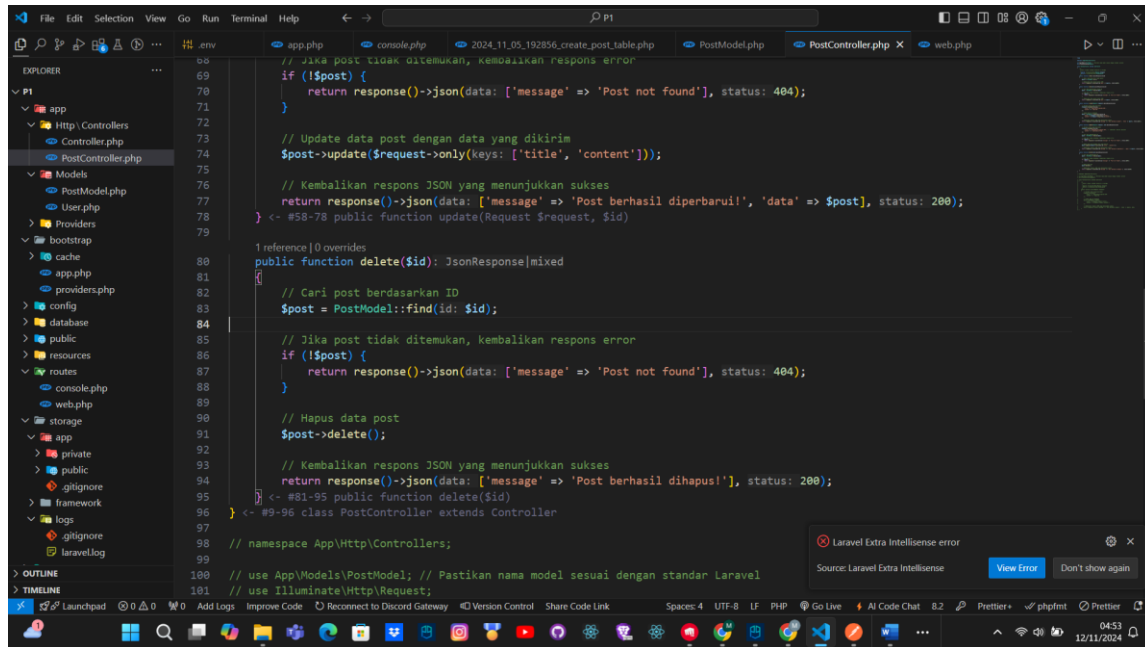
```
public function delete($id)
{
    // Cari post berdasarkan ID
    $post = PostModel::find($id);

    // Jika post tidak ditemukan, kembalikan respons error
    if (!$post) {
        return response()->json(['message' => 'Post not found'], 404);
    }

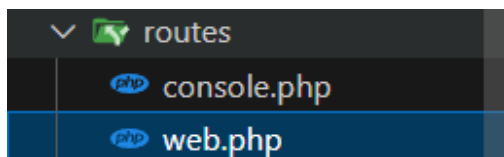
    // Hapus data post
    $post->delete();
}
```

```
// Kembalikan respons JSON yang menunjukkan sukses
return response()->json(['message' => 'Post berhasil dihapus!'], 200);
}
```

- Contoh penerapan :



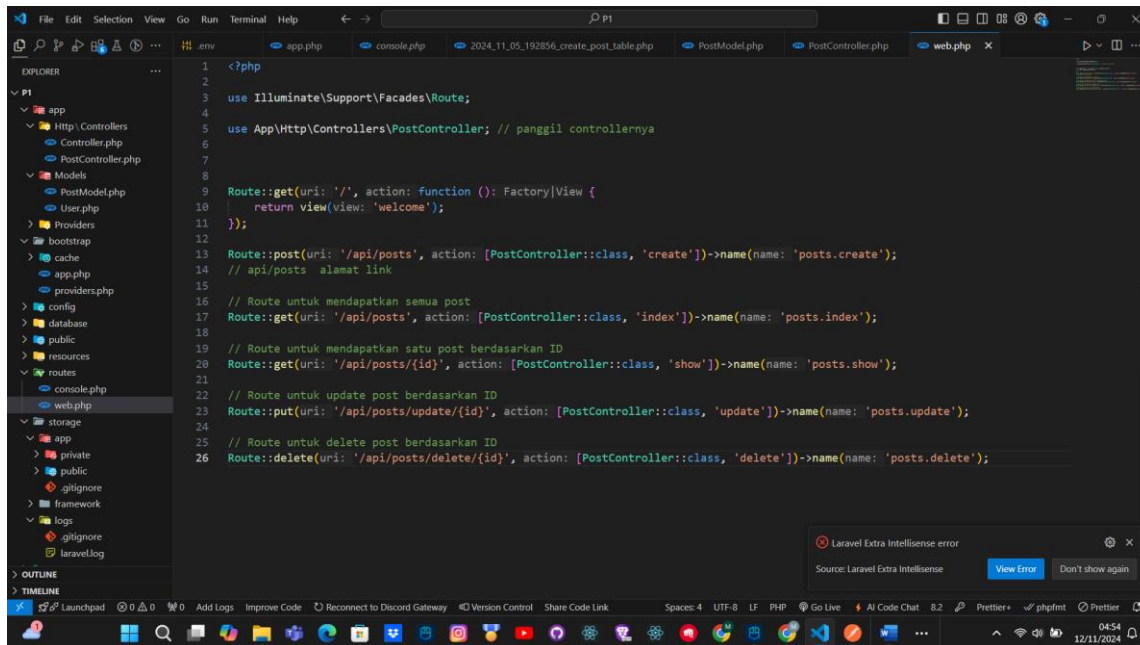
3. Kemudian setelah controller dibuat, kemudian buatlah routesnya, buka file routes di routes\web.php



4. Kemudian jika sudah dibuat, bisa ditambahkan code routes untuk delete berdasarkan dari controller yang dibuat, contoh seperti ini :

```
// Route untuk delete post berdasarkan ID
Route::delete('/api/posts/delete/{id}', [PostController::class, 'delete'])->name('posts.delete');
```

Contoh Penerapan:



```
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\PostController; // panggil controllernya

Route::get(uri: '/', action: function () { return view('welcome'); });

Route::post(uri: '/api/posts', action: [PostController::class, 'create'])->name(name: 'posts.create');
// api/posts alamat link

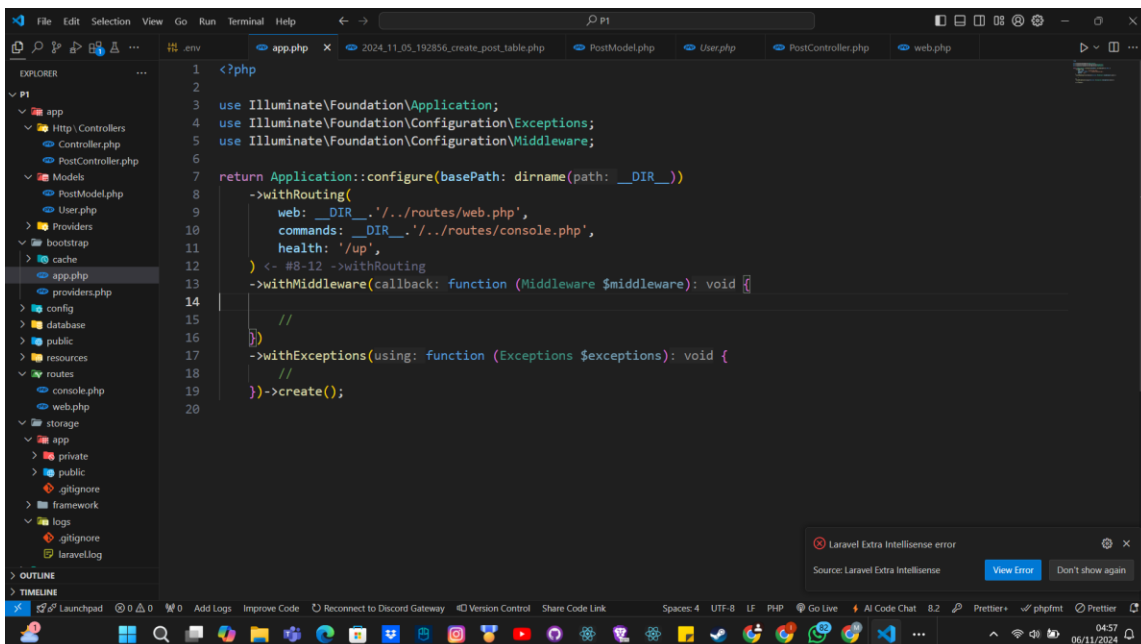
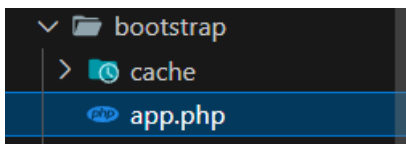
// Route untuk mendapatkan semua post
Route::get(uri: '/api/posts', action: [PostController::class, 'index'])->name(name: 'posts.index');

// Route untuk mendapatkan satu post berdasarkan ID
Route::get(uri: '/api/posts/{id}', action: [PostController::class, 'show'])->name(name: 'posts.show');

// Route untuk update post berdasarkan ID
Route::put(uri: '/api/posts/update/{id}', action: [PostController::class, 'update'])->name(name: 'posts.update');

// Route untuk delete post berdasarkan ID
Route::delete(uri: '/api/posts/delete/{id}', action: [PostController::class, 'delete'])->name(name: 'posts.delete');
```

5. Setelah membuat routesnya, kemudian buka bootstrap\app.php



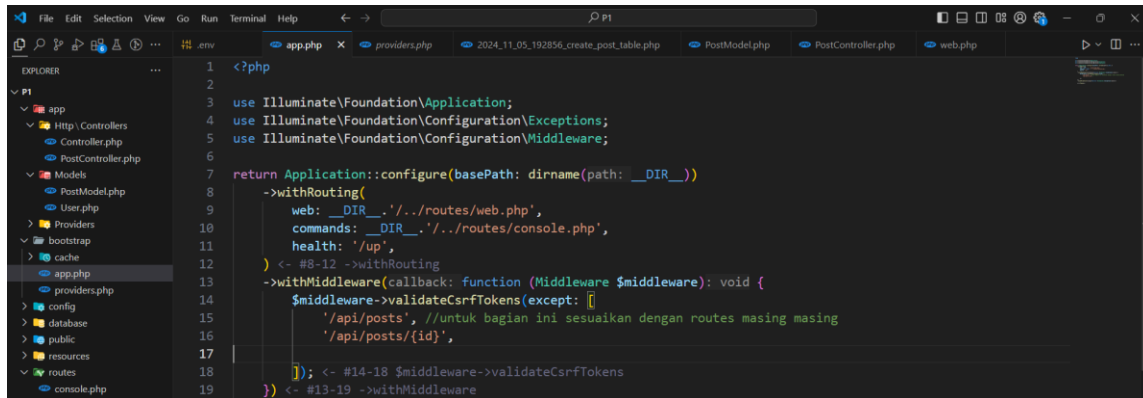
```
<?php
use Illuminate\Foundation\Application;
use Illuminate\Foundation\Configuration\Exceptions;
use Illuminate\Foundation\Configuration\Middleware;

return Application::configure(basePath: dirname(path: __DIR__))
    ->withRouting(
        web: __DIR__.'/../routes/web.php',
        commands: __DIR__.'/../routes/console.php',
        health: '/up',
    ) <- #B-12 ->withRouting
    ->withMiddleware(callback: function (Middleware $middleware): void {
        //
    })
    ->withExceptions(using: function (Exceptions $exceptions): void {
        //
    })->create();
```

6. Setelah itu isillah bagian return yang withMiddleware, dan tambahkanlah routes yang baru saja dibuat untuk delete nya :

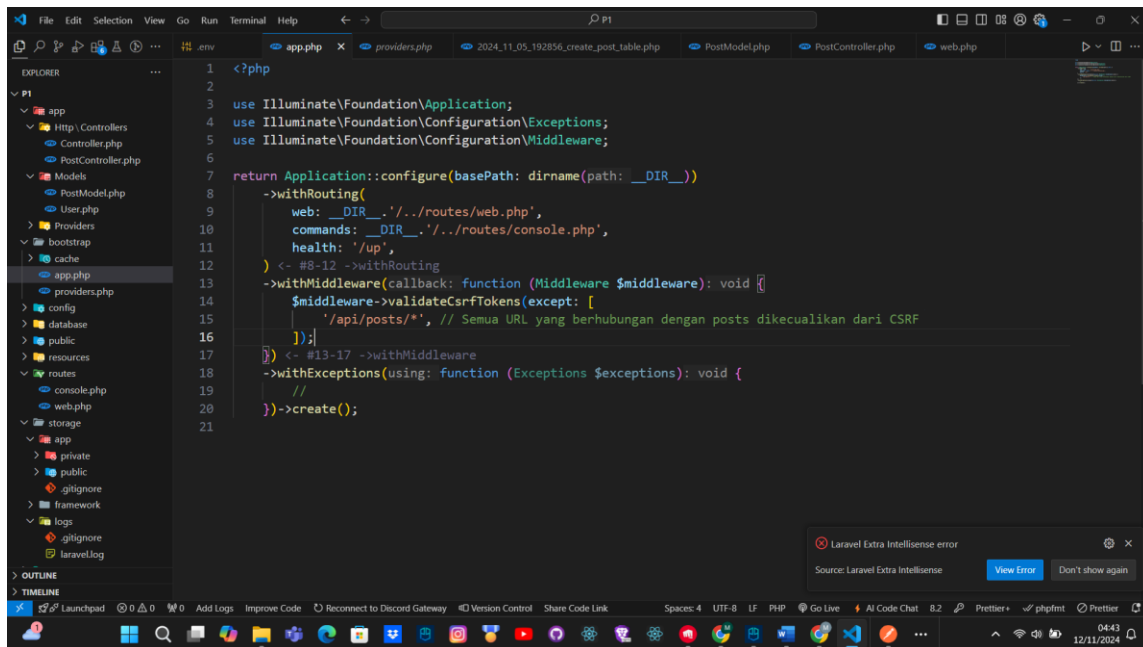
```
$middleware->validateCsrfTokens(except: [  
    '/api/posts/*', // Semua URL yang berhubungan dengan posts dikecualikan dari CSRF  
]);
```

Sebelum :



```
<?php  
use Illuminate\Foundation\Application;  
use Illuminate\Foundation\Configuration\Exceptions;  
use Illuminate\Foundation\Configuration\Middleware;  
  
return Application::configure(basePath: dirname(__DIR__))  
    ->withRouting(  
        web: __DIR__.'/../routes/web.php',  
        commands: __DIR__.'/../routes/console.php',  
        health: '/up',  
    )  
    <- #8-12 ->withRouting  
    ->withMiddleware(callback: function (Middleware $middleware): void {  
        $middleware->validateCsrfTokens(except: [  
            '/api/posts', //untuk bagian ini sesuaikan dengan routes masing masing  
            '/api/posts/{id}',  
        ]);  
    });  
    <- #14-18 $middleware->validateCsrfTokens  
    <- #13-19 ->withMiddleware
```

Sesudah :



```
<?php  
use Illuminate\Foundation\Application;  
use Illuminate\Foundation\Configuration\Exceptions;  
use Illuminate\Foundation\Configuration\Middleware;  
  
return Application::configure(basePath: dirname(__DIR__))  
    ->withRouting(  
        web: __DIR__.'/../routes/web.php',  
        commands: __DIR__.'/../routes/console.php',  
        health: '/up',  
    )  
    <- #8-12 ->withRouting  
    ->withMiddleware(callback: function (Middleware $middleware): void {  
        $middleware->validateCsrfTokens(except: [  
            '/api/posts/*', // Semua URL yang berhubungan dengan posts dikecualikan dari CSRF  
        ]);  
    });  
    <- #13-17 ->withMiddleware  
    ->withExceptions(using: function (Exceptions $exceptions): void {  
        //  
    })->create();
```

Laravel Extra Intellisense error
Source: Laravel Extra Intellisense
View Error Don't show again

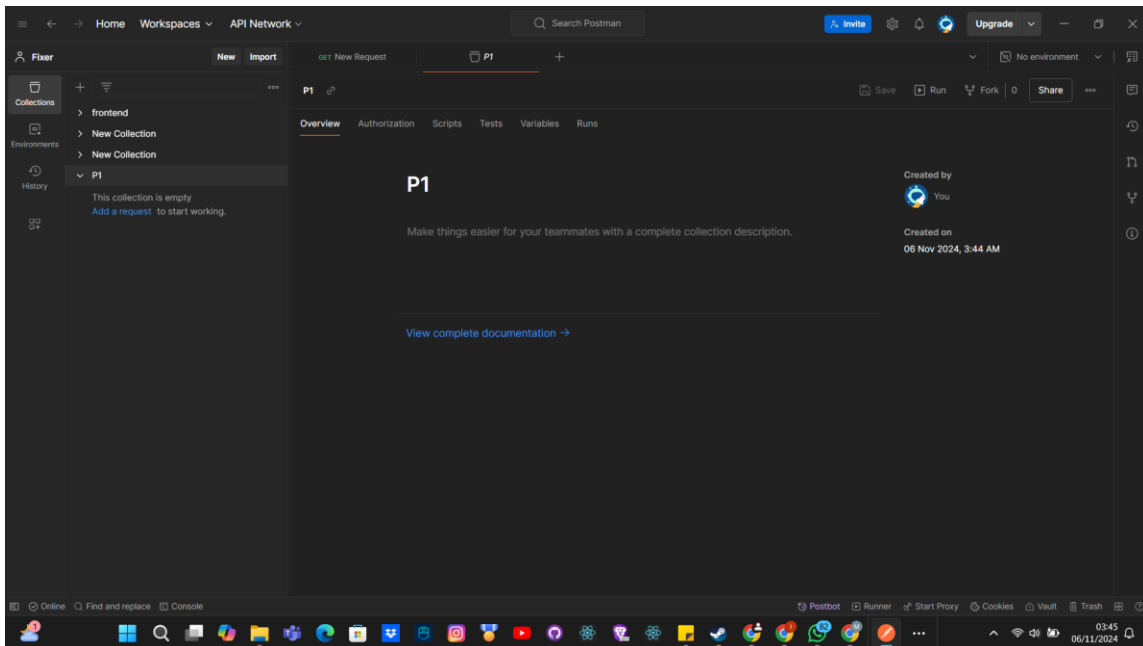
7. Kemudian buka kembali terminal dan nyalakan project laravelnya dengan command :
php artisan serve

```
D:\materi asdos\backend\P1
λ php artisan serve
forking is not supported on this platform

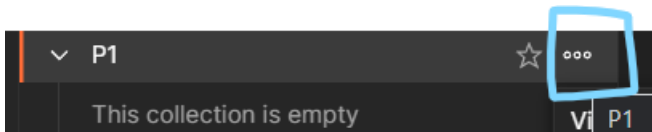
INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

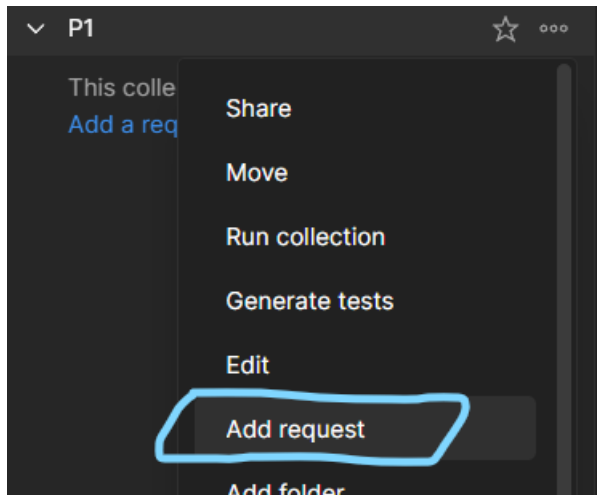
8. Kemudian buka postman, setelah itu bukalah collection yang kalian sudah buat sebelumnya
Contoh :



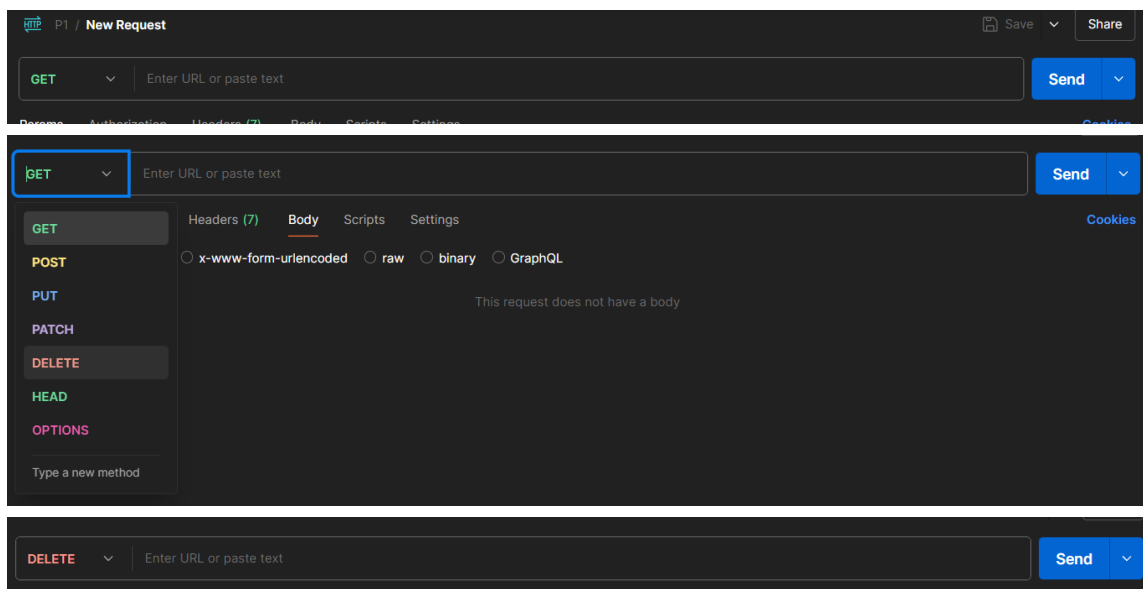
9. Setelah terbuat, klik titik tiga yang berada di sebelah nama project



10. Kemudian klik add request



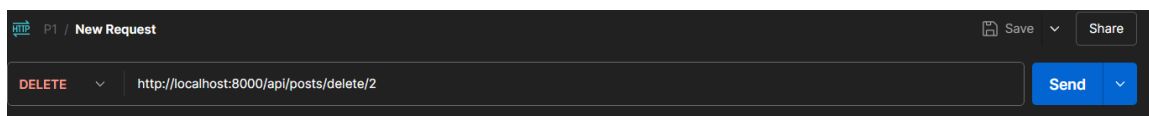
11. Setelah itu gantilah requestnya menjadi DELETE



12. Setelah itu isilah url atasnya sesuai dengan routes yang dibuat :

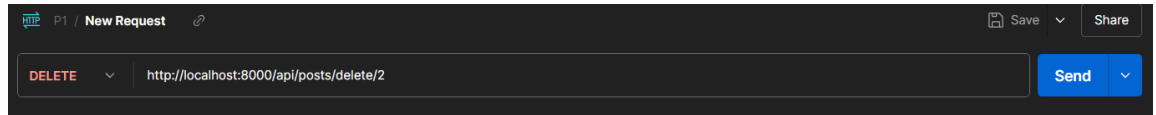
- Function delete berdasarkan id yang dipilih

URL: <http://localhost:8000/api/posts/delete/2>

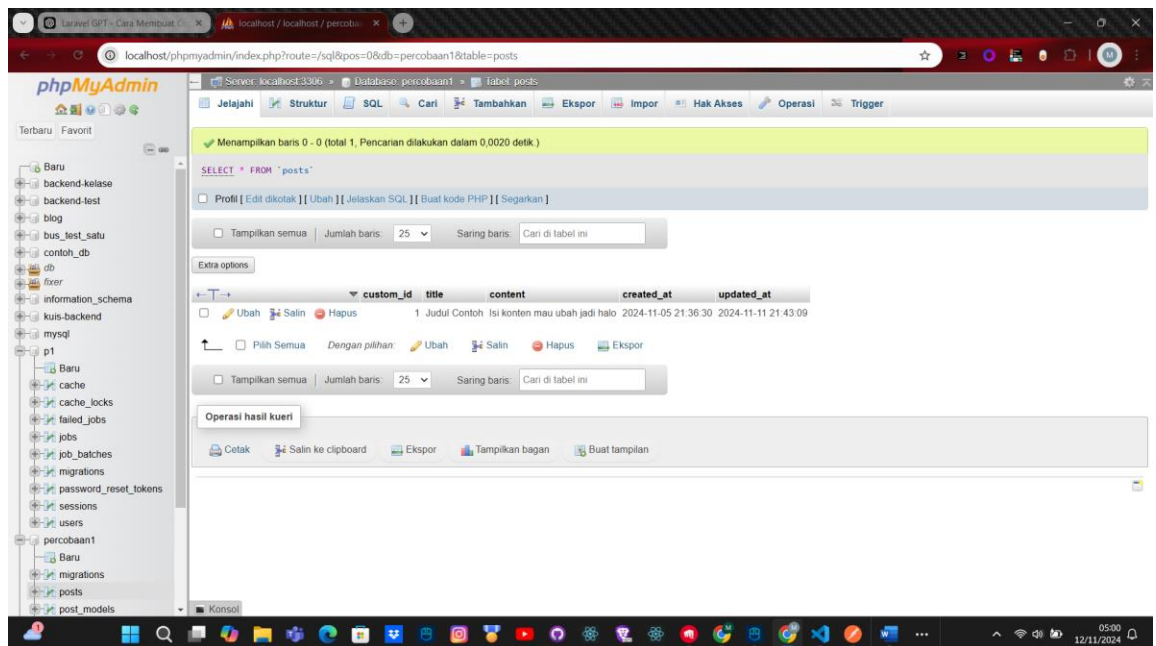
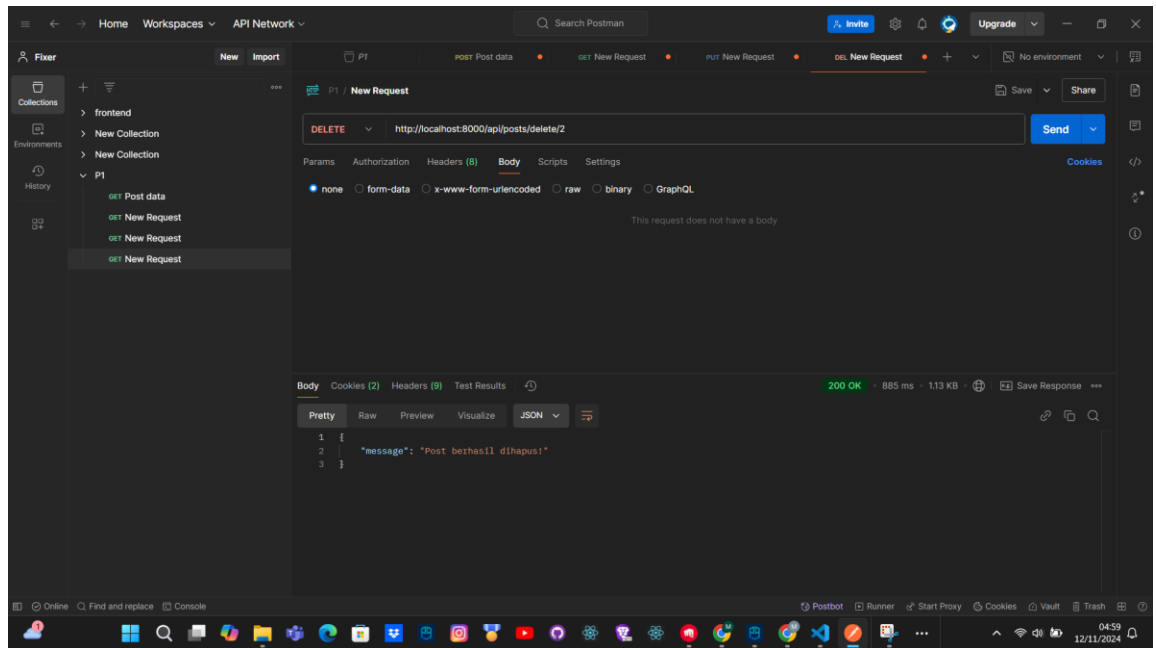


13. Kemudian klik send, dan cek di database apakah datanya sudah terhapus atau belum

- Function delete berdasarkan id yang dipilih (contoh disini mau delete data dari id 2), dan klik send



Hasil :



2. TUGAS

Jawablah Pertanyaan dari Soal ini |

1. Cobalah untuk melakukan semua percobaan diatas
2. Kemudian buatlah versi project kalian sendiri
3. minimal sudah 3 table yang bisa melakukan proses delete berdasarkan dengan id(primary key dari table tersebut)

PENGUMPULAN TUGAS DAN DEADLINE

- a. Pengumpulan Tugas di Google Drive
- b. Deadline = 13 November 2024 | 23.00 WIB

DAFTAR PUSTAKA

<https://buildwithangga.com/tips/membangun-aplikasi-crud-sederhana-menggunakan-laravel-11>

<https://baraka.uma.ac.id/mengenal-controller-pada-laravel-pusat-kendali-aplikasi-web/#:~:text=Dalam%20konteks%20Laravel%2C%20controller%20adalah,atau%20ke%20tampilan%20yang%20tepat.>

<https://buildwithangga.com/tips/apa-itu-model-pada-framework-laravel>
<https://www.php.net/downloads>

<https://parsinta.com/articles/3-metode-laravel-routing-yang-sering-digunakan-ztwbcp>

<https://medium.com/@kevinffa0107/pengertian-mvc-model-view-controller-pada-framework-laravel-20f261ccf233>