

Kitab CRUD Laravel

Web Prog

Todo → membuat **aplikasi simple notes** yang bisa create notes, update notes, show notes, delete notes

Setup

First things first, pastikan environment kita udah oke yaitu

- xampp/herd/laragon
- nodes + npm
- buka dokumentasi laravel <https://laravel.com/docs>

create project → sesuaikan pathnya

```
> Bash
composer create-project laravel/laravel notesApp
```

```
> Bash
cd notesApp
```

buka di text editor kesayangan kalian kalau kalian pakai vs code bisa langsung

```
> Bash
code .
```

sebelum kita ubah apapun di dalam codingannya selalu pastikan kita settings project kita dulu dengan membuka **.env**

config envnya yang awalnya seperti ini

```
> Bash
DB_CONNECTION=sqlite
# DB_HOST=127.0.0.1
# DB_PORT=3306
# DB_DATABASE=laravel
# DB_USERNAME=root
# DB_PASSWORD=
```

menjadi seperti ini

```
> Bash
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=notesApp #nama database kalian
DB_USERNAME=root
DB_PASSWORD=
```

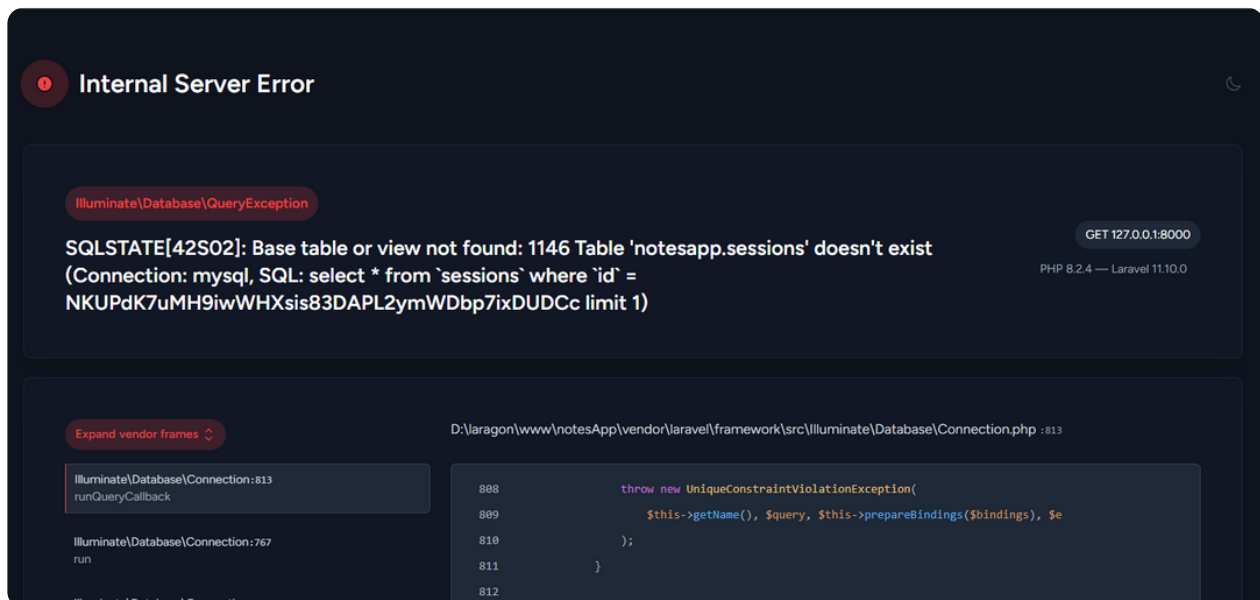
buat database notesApp → di phpMyAdmin

jalankan project

```
> Bash
php artisan serve
```

buka <http://127.0.0.1:8000> di browser kesayangan kalian

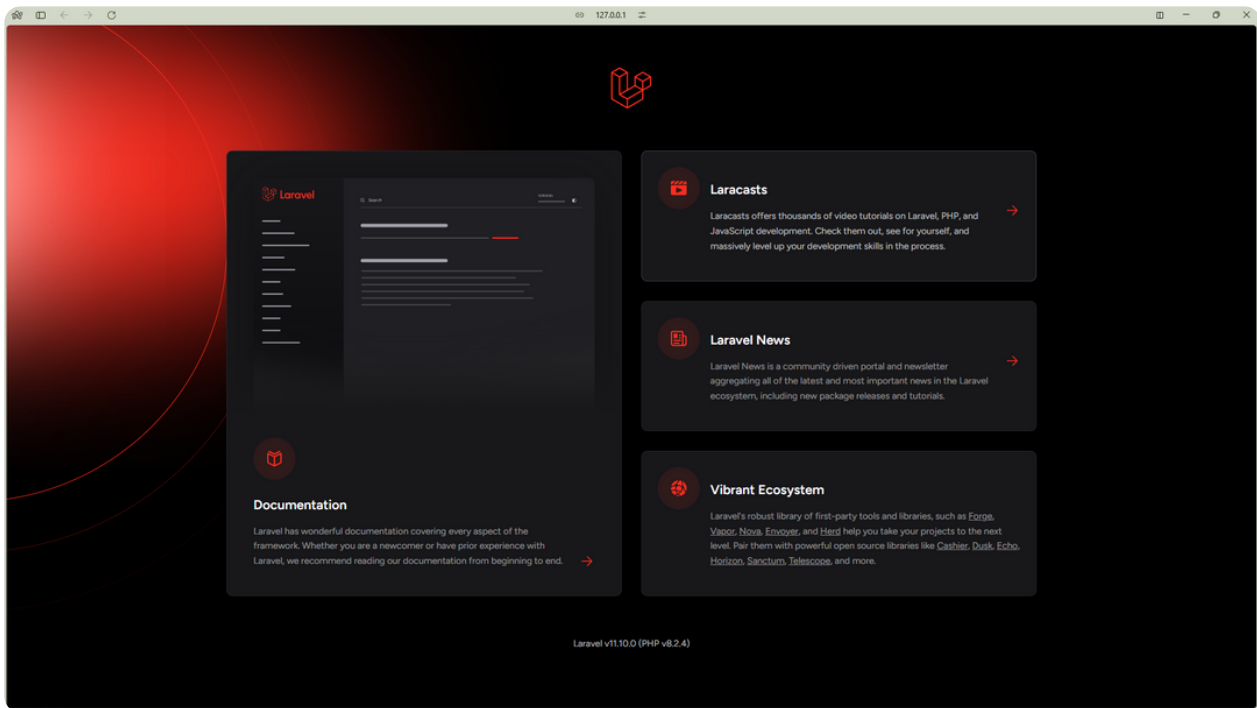
kalau kalian mendapatkan error ini tandanya kalian belum punya table session which table bawaan laravel



cara troubleshootnya kita migrate dulu friends

```
> Bash
php artisan migrate
```

taraa.. seharusnya sudah bisa



Analisis

sebelum kita mulai membuat sebuah project baiknya kita analisis dulu soalnya. di soalnya kita diminta buat apaa... disini instruksi soalnya cuman minta "simple notes application" kira kira apa aja yang harus kita pikirkan pertama?

1. **schema database kita mau seperti apa?** asumsi ada seorang user yang mau membuat notes berarti ada table user dan notes dengan isi judul notes, body notes, dibuat kapan. **gimana dengan tampilan?** mungkin ada tampilan yang menunjukkan list notesnya, create notesnya, show detail notesnya, edit notesnya, sama delete notesnya dong yaa... **operasi ada apa aja?** store data, update data, destroy data

dari hasil itu kita bisa menyimpulkan beberapa hal yaitu

1. model user dan notes
2. views index, create, show, edit
3. controller, store, update, destroy

biasanya flow yang bisa aku lakuin :

1. model + migration + seeder
2. views
3. controller + route

Model + Migration + Seeder

kita bisa membuat model , migration dan seeder secara independent tapi disini ada command yang membantu kita membuat semuanya sekaligus

 Bash

```
php artisan make:model NamaModel -m -s
```

overview sedikit, model itu yang menghubungkan laravel dengan database, migration membantu kita membuat schema database, dan seeder adalah tools untuk menambahkan data

posisi saat ini laravel sudah membuatkan kita model user, migration untuk user seperti dibawah ini
User.php

PHP

```
<?php
```

```
namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * Get the attributes that should be cast.
     *
     * @return array<string, string>
     */
    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }
}
```

0001_01_01_000000_create_users_table.php

PHP

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });

        Schema::create('password_reset_tokens', function (Blueprint $table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });

        Schema::create('sessions', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->foreignId('user_id')->nullable()->index();
            $table->string('ip_address', 45)->nullable();
            $table->text('user_agent')->nullable();
            $table->longText('payload');
            $table->integer('last_activity')->index();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('users');
        Schema::dropIfExists('password_reset_tokens');
        Schema::dropIfExists('sessions');
    }
};
```

laravel juga membuat sebuah factory user (membuat user sama seperti seeder) yang di panggil di DatabaseSeeder.php

```
PHP
<?php

namespace Database\Seeders;

use App\Models\User;
// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        // User::factory(10)->create();

        User::factory()->create([
            'name' => 'Test User',
            'email' => 'test@example.com',
        ]);
    }
}
```

kita bisa mengulang migration kita dan menambahkan factory dengan menjalankan command ini

```
Bash
php artisan migrate:fresh --seed
```

```
alfon@Ananta MINGW64 /d/laragon/www/notesApp
$ php artisan migrate:fresh --seed

Dropping all tables ..... 40.12ms DONE

INFO Preparing database.

Creating migration table ..... 9.97ms DONE

INFO Running migrations.

0001_01_01_000000_create_users_table ..... 55.57ms DONE
0001_01_01_000001_create_cache_table ..... 13.65ms DONE
0001_01_01_000002_create_jobs_table ..... 57.87ms DONE

INFO Seeding database.
```

Server: 127.0.0.1 » Database: notesapp » Table: users

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

`SELECT * FROM `users``

☐ Profiling
 [\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PHP code \]](#)
[\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

	id	name	email	email_verified_at	password
<input type="checkbox"/> Edit Copy Delete	1	Test User	test@example.com	2024-06-13 06:52:08	\$2y\$12\$pi

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all | Number of rows: 25 | Filter rows:

yuk kita buat model migration dan seeder untuk notes

```
Bash
php artisan make:model Note -m -s
```

maka akan terbuat 3 file yaitu model Note.php, seeder NoteSeeder.php, sama create notes table selanjutnya apa?

1. setup migration
2. setup seeder (kalau ujian tida dibutuhkan skip aja)
3. setup model

Setup migrations

 Bash

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('notes', function (Blueprint $table) {
            $table->id();
            $table->string('Title');
            $table->longText('body');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('notes');
    }
};
```

ada macem" tipe column yang bisa kita isi ada integer, string, longtext,unsignedBigInteger
di laravel primaryKey selalu 'id' dan autoincrement yaa tapi kita tetep bisa customize ko

migrate saja langsung

 Bash

```
php artisan migrate:fresh
```

Setup Seeder

mengupdate seeder supaya bisa langsung isi notes

 Bash

```
<?php
```

```
namespace Database\Seeders;
```

```
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
```

```
use Illuminate\Database\Seeder;
```

```
use Illuminate\Support\Facades\DB;
```

```
class NoteSeeder extends Seeder
```

```
{
```

```
    /**
```

```
     * Run the database seeds.
```

```
     */
```

```
    public function run(): void
```

```
    {
```

```
        DB::table('notes')->insert([
```

```
            'Title' => 'Notes 1',
```

```
            'Body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit?',
```

```
        ]);
```

```
    }
```

```
}
```

kalau kalian jalanin php artisan migrate:fresh --seed si seeder notes ini tidak akan jalan temen" karena dia nda di call di DatabaseSeeder jadi kita modify dulu DatabaseSeeder nya

```

PHP
<?php

namespace Database\Seeders;

use App\Models\User;
// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        // User::factory(10)->create();

        User::factory()->create([
            'name' => 'Test User',
            'email' => 'test@example.com',
        ]);
        $this->call([
            NoteSeeder::class,
        ]);
    }
}

```

jangan lupa import yaa, takutnya dia ga otomatis, ini berlaku untuk semua seeder yang kalian buat yaa kalau mau pakai command

```

Bash

php artisan migrate:fresh --seed

```

boleh dijalanin lagi, command diatas

```

$ php artisan migrate:fresh --seed

Dropping all tables ..... 43.70ms DONE

INFO Preparing database.

Creating migration table ..... 10.32ms DONE

INFO Running migrations.

0001_01_01_000000_create_users_table ..... 52.01ms DONE
0001_01_01_000001_create_cache_table ..... 11.53ms DONE
0001_01_01_000002_create_jobs_table ..... 48.34ms DONE
2024_06_13_065451_create_notes_table ..... 5.93ms DONE

INFO Seeding database.

Database\Seeders\NoteSeeder ..... RUNNING
Database\Seeders\NoteSeeder ..... 3 ms DONE

```

seharusnya si noteSeeder sudah jalan ya.. kalau kalian notice factorynya ko nda ada runningnya? iaa.. karena si laravel belum cs sama factory xd

seharusnya sekarang sudah ada data di table notes kalian

Extra options

	id	Title	body	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Notes 1	Lorem ipsum dolor sit amet consectetur adipisicing...	NULL	NULL

Setup Model

Note.php

```
PHP
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    use HasFactory;

    protected $table = 'notes'; #buat pastiin tablenya bener
    protected $fillable = ['title','body']; #field apa aja yang bisa diisi
    protected $primaryKey = 'id'; #mastiin primarykeynya
}
```

protected ini template temen" ada beberapa yang harus di lihat, primarykey biasanya langsung kedetek

Views + Controller + Route

replace welcome.blade.php menjadi seperti ini

```
PHP
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Notes App</title>
</head>

<body>
    <h1>Notes App</h1>
</body>

</html>
```

seharusnya kalau kalian refresh browser kalian



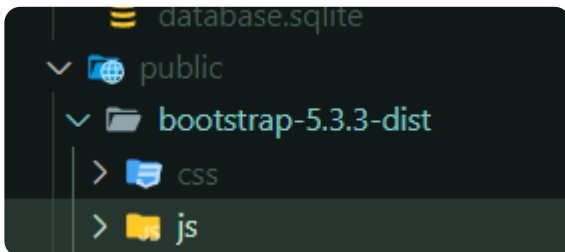
whats next? setup bootstrap

setup bootstrap ada banyak ya, cdn bisa sebagai variasi kalau kalian download lewat sini

<https://github.com/twbs/bootstrap/releases/download/v5.3.3/bootstrap-5.3.3-dist.zip>

extract here dan taruh di public project laravel kalian

import css, sama jsnya



PHP

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Notes App</title>
  <link rel="stylesheet" href="/bootstrap-5.3.3-dist/css/bootstrap.min.css">
</head>

<body>
  <h1>Notes App</h1>
  <script src="/bootstrap-5.3.3-dist/js/bootstrap.min.js"></script>
</body>

</html>
```

sesuaikan dengan path dan nama folder kaliankalian

PHP

```
<link rel="stylesheet" href="/bootstrap-5.3.3-dist/css/bootstrap.min.css">
```

PHP

```
<script src="/bootstrap-5.3.3-dist/js/bootstrap.min.js"></script>
```

kalau ada customer css?

sama ya tinggal di tambahin aja

PHP

```
<link rel="stylesheet" href="/style.css">
```

sekarang kita bakal pake component dari bootstrap kayak navbar dan footer dan mengimplementasikan include, yields dan section

settings navbar footer seperti ini

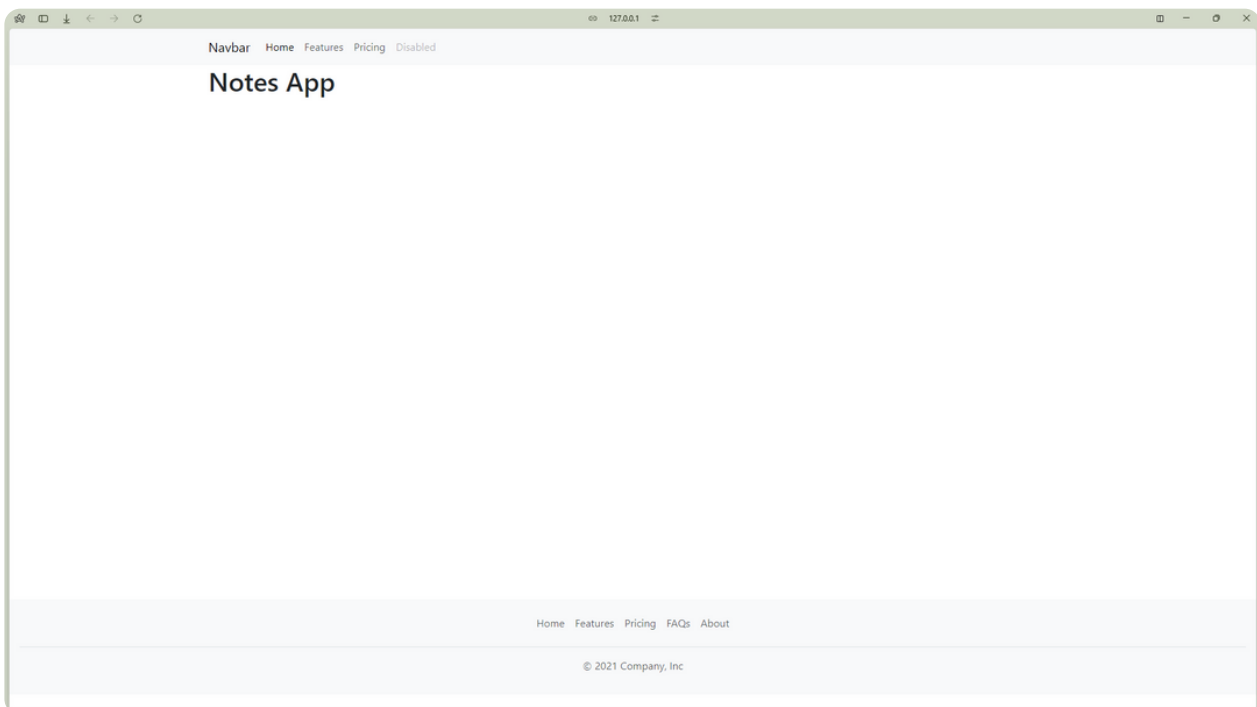
```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-
bs-target="#navbarNav"
      aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" aria-disabled="true">Disabled</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
<main>
  <div class="container">
    <h1>Notes App</h1>
  </div>
</main>
<footer class="py-3 my-4">
  <ul class="nav justify-content-center border-bottom pb-3 mb-3">
    <li class="nav-item"><a href="#" class="nav-link px-2 text-muted">Home</a>
  </li>
    <li class="nav-item"><a href="#" class="nav-link px-2 text-
muted">Features</a></li>
    <li class="nav-item"><a href="#" class="nav-link px-2 text-muted">Pricing</a>
  </li>
    <li class="nav-item"><a href="#" class="nav-link px-2 text-muted">FAQs</a>
  </li>
    <li class="nav-item"><a href="#" class="nav-link px-2 text-muted">About</a>
  </li>
  </ul>
  <p class="text-center text-muted">&copy; 2021 Company, Inc</p>
</footer>
```

add style.css juga supaya footer selalu dibawah

PHP

```
body {  
    display: flex;  
    flex-direction: column;  
    min-height: 100vh;  
}  
  
main {  
    flex: 1;  
}  
  
footer {  
    padding: 1rem;  
    margin-top: auto;  
    background-color: #f8f9fa; /* Example background color */  
}
```

kira-kira tampilannya seperti ini



supaya code views kita lebih bagus kita bisa pakai templating engine dari laravel seperti include dan yield temen"

cut code nav dan buat nav.blade.php begitu juga dengan footer

kira" seperti ini

PHP

```
<body>
  @include('nav') # nav -> dari file nav.blade.php
  <main>
    <div class="container">
      <h1>Notes App</h1>
    </div>
  </main>
  @include('footer') # footer -> dari file footer.blade.php
  <script src="/bootstrap-5.3.3-dist/js/bootstrap.min.js"></script>
</body>
```

buat file baru dengan nama index.blade.php

dan cut h1 nya dan tambahkan @yield setelah div dengan container container

file welcome.blade ini akan kita jadikan template / parent temen" yang akan di extend oleh page" lainnya. sekarang rename welcome.blade.php menjadi template.blade.php

ubah route ('/') menjadi views menjadi index seperti dibawah ini

PHP

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('index');
});
```

seharusnya kalau kita refresh maka jadi seperti ini



tidak ada navbarnya dan footernya temen" karena di dalam index.blade.php kita belum ngenalin parent bladennya jadi buka index.blade.php dan update isinya menjadi seperti berikut

index.blade.php

```
PHP
@extends('template')

@section('content')
    <h1>Notes App</h1>
@endsection
```

intinya sambungin dengan extends file template.php, dengan isi sectionnya yaitu content kalau kalian notice sectionnya itu namanya sama kayak yang di @yield

selanjutnya kita bisa buka index.blade.php dan tambah code table untuk menampilkan data

PHP

```
@extends('template')

@section('content')
    <h1>Notes App</h1>
    <button type="button" class="btn btn-primary">Add Data</button>
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">First</th>
                <th scope="col">Last</th>
                <th scope="col">Handle</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th scope="row">1</th>
                <td>Mark</td>
                <td>Otto</td>
                <td>@mdo</td>
                <td>
                    <button type="button" class="btn btn-warning">Edit</button>
                    <button type="button" class="btn btn-danger">Delete</button>
                </td>
            </tr>
        </tbody>
    </table>
@endsection
```

sekarang saatnya membuat controller dan route untuk menampilkan data, create, update, delete serta membuat viewsnya

jalankan code ini

 Bash

```
php artisan make:controller NoteController
```

seharusnya file NoteController.php terbuat

PHP

```
<?php
```

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models>Note;

class NoteController extends Controller
{
    public function index()
    {

    }

    public function create()
    {

    }

    public function store(Request $request)
    {

    }

    public function show(Note $note)
    {

    }

    public function edit(Note $note)
    {

    }

    public function update(Request $request, Note $note)
    {

    }

    public function destroy(Note $note)
    {

    }
}
```

method ini template, index untuk menampilkan jadi tidak ada parameter, create → views menambahkan data, store → method save data, show → menampilkan spesifik data, edit → views edit, update → method update data, destroy → delete data

karena sudah membuat controller mari kita tampilkan data dari seeder yang kita buat

```
PHP
public function index()
{
    $notes = Note::all();
    return view('index', ['notes' => $notes]);
}
```

menggunakan query builder kita meminta semua data di model atau table note dan assign di variable notes, setelah itu me return views yang akan dikirimkan array asosiatif notes yang isinya hasil querynya

ubah route supaya bisa mengakses controller notes dan method index

```
PHP
Route::get('/', [NoteController::class, 'index']);
```

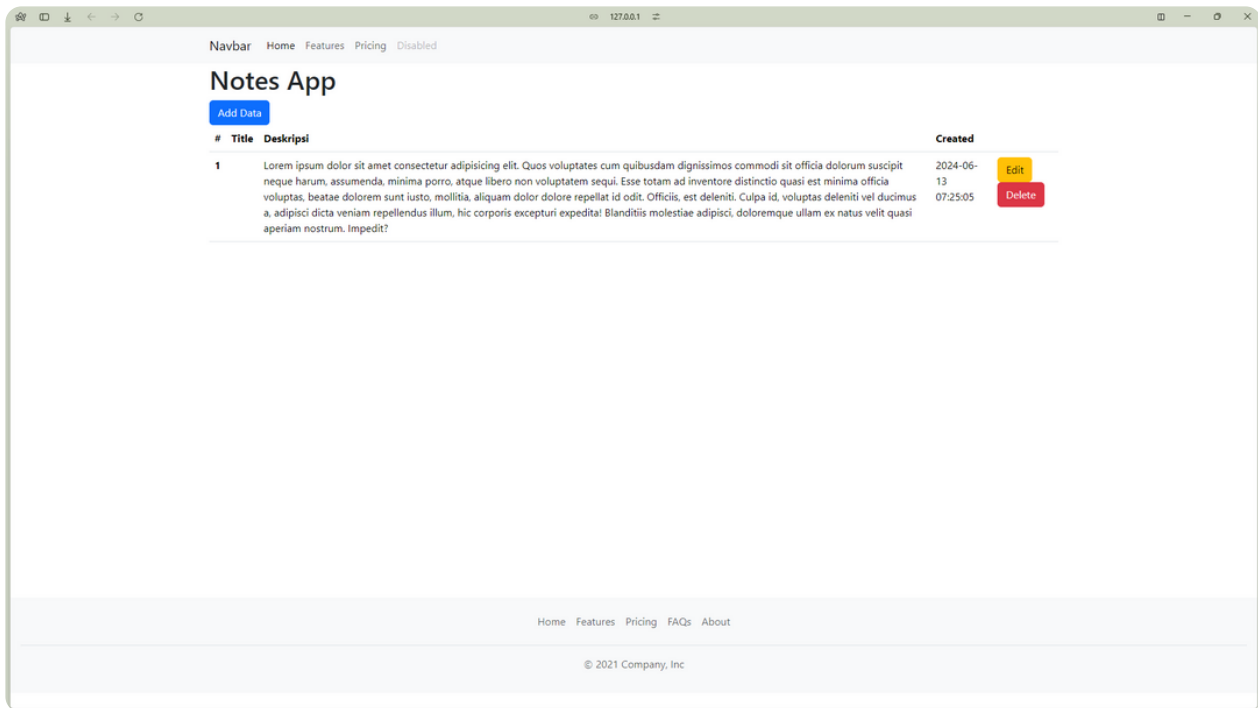
mengubah views kita supaya menampilkan data yang kita kirim

PHP

```
@extends('template')

@section('content')
    <h1>Notes App</h1>
    <button type="button" class="btn btn-primary">Add Data</button>
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Title</th>
                <th scope="col">Deskripsi</th>
                <th scope="col">Created</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody>
            @foreach ($notes as $note)
                <tr>
                    <th scope="row">{{ $note->id }}</th>
                    <td>{{ $note->title }}</td>
                    <td>{{ $note->body }}</td>
                    <td>{{ $note->created_at }}</td>
                    <td>
                        <button type="button" class="btn btn-warning">Edit</button>
                        <button type="button" class="btn btn-danger">Delete</button>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
@endsection
```

intinya foreach array asosiatif dengan nama note, \$note->id itu di dasari sama nama column



hasilnya akan seperti ini

gimana cara menambahkan data?

add controllernya untuk bisa menambahkan data

PHP

```
public function create()
{
    return view('create');
}
```

buat views create.blade.php, jangan lupa extend dan section dan end sectionnya

PHP

```
@extends('template')

@section('content')
    <h1>Create Notes</h1>
    <form action="" method="POST" enctype="multipart/form-data">
        @csrf
        <div class="mb-3">
            <label for="title" class="form-label">Title</label>
            <input type="text" class="form-control" id="title" placeholder="title"
name="title">
        </div>
        <div class="mb-3">
            <label for="body" class="form-label">Body</label>
            <textarea class="form-control" id="body" rows="3" name="body"></textarea>
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>
@endsection
```

selanjutnya menambahkan link dan route untuk create notes ubah di index bagian button replace menjadi tag a

PHP

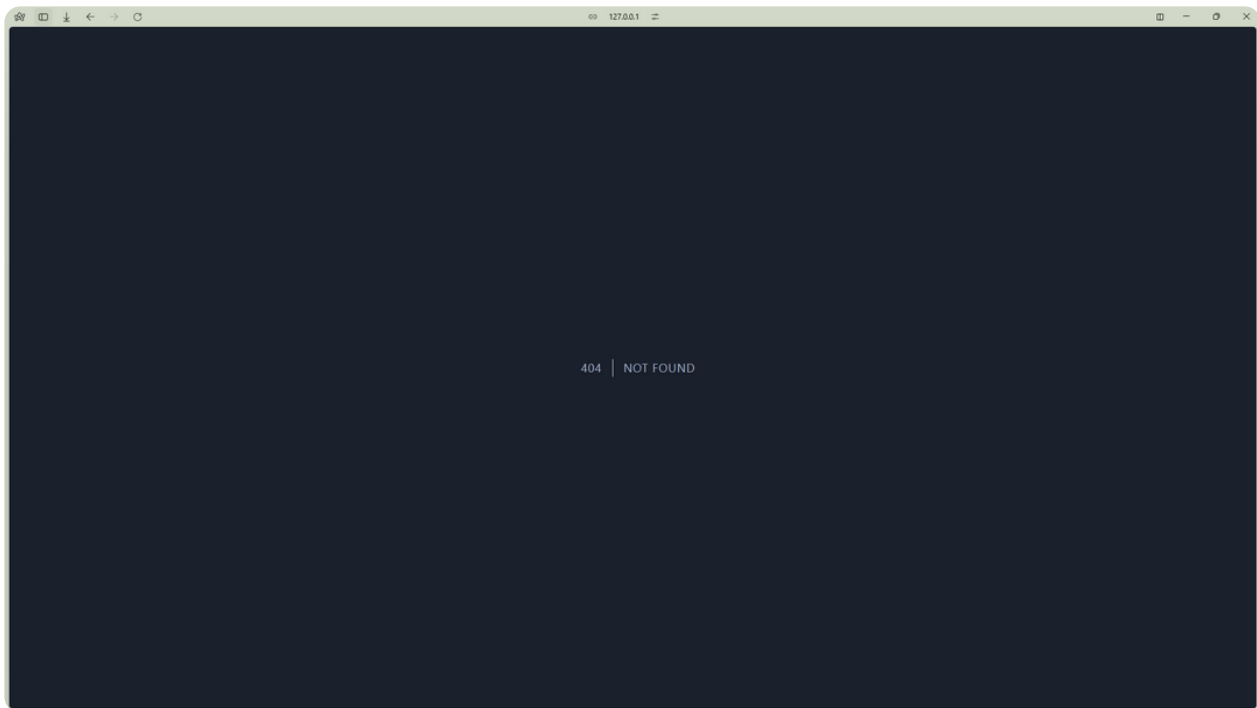
```
<a href="/create" class="btn btn-primary">Add Data</a>
```


PHP

```
@extends('template')

@section('content')
    <h1>Notes App</h1>
    <a href="/create" class="btn btn-primary">Add Data</a>
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Title</th>
                <th scope="col">Deskripsi</th>
                <th scope="col">Created</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody>
            @foreach ($notes as $note)
                <tr>
                    <th scope="row">{{ $note->id }}</th>
                    <td>{{ $note->title }}</td>
                    <td>{{ $note->body }}</td>
                    <td>{{ $note->created_at }}</td>
                    <td>
                        <button type="button" class="btn btn-warning">Edit</button>
                        <button type="button" class="btn btn-danger">Delete</button>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
@endsection
```

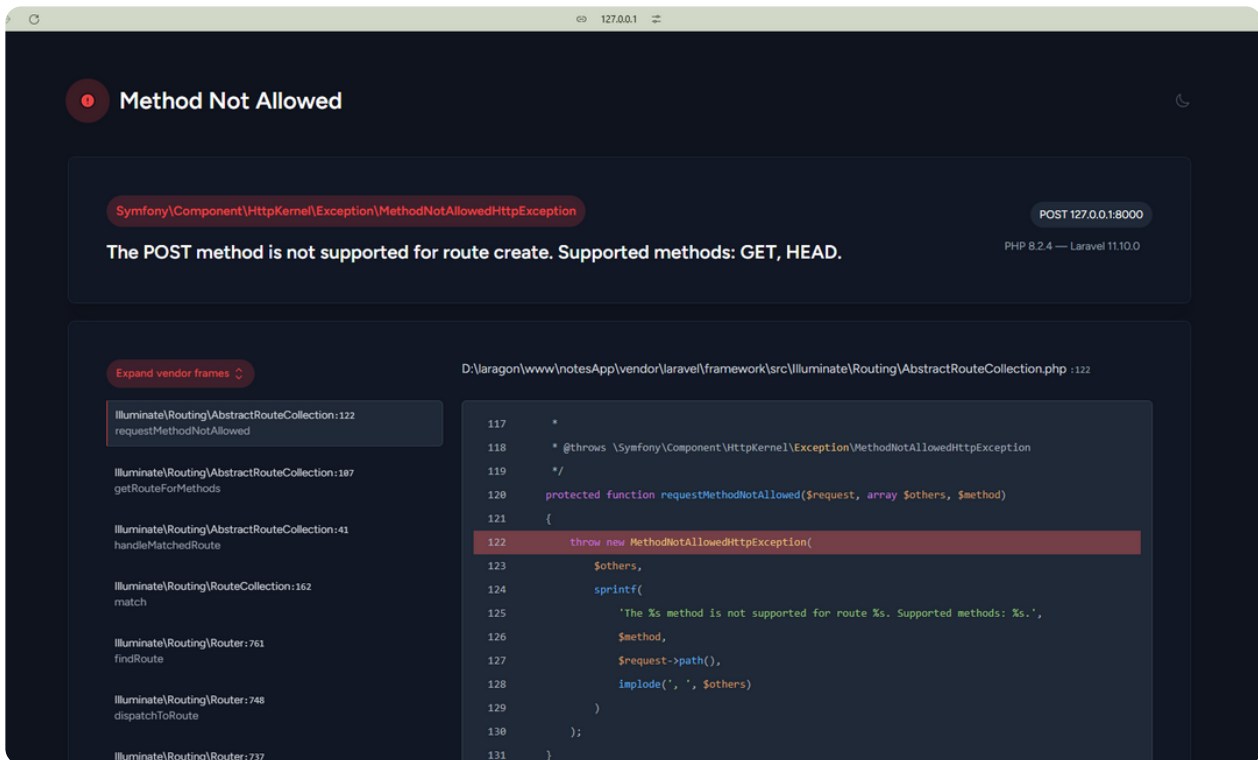
kalau kalian pencet akan error temen"



karena belum ada routenya mari kita tambahkan route di web.php



kalau kalian isi bakal error seperti ini karena belum ada route , action pada form dan methodnya



jadi mari kita benahi satu"

```

section('content')
    <h1>Create Notes</h1>
    <form action="/store" method="POST" enctype="multipart/form-data">
        @csrf
        <div class="mb-3">
            <label for="title" class="form-label">Title</label>
            <input type="text" class="form-control" id="title">
        </div>
        <div class="mb-3">
            <label for="body" class="form-label">Body</label>
            <textarea class="form-control" id="body" rows="3">
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>

```

tambahkan action ke /store

dan buat route dengan methodid post yang merujuk ke method store di controller note

PHP

```
Route::post('/store', [NoteController::class, 'store']);
```

mengupdate store method di NoteController

```

PHP
public function store(Request $request)
{
    $validatedData = $request->validate([
        'title' => 'required|max:255',
        'body' => 'required',
    ]);

    Note::create($validatedData);

    return redirect('/');
}

```

disini kita memvalidasi data teman" dimana title require dan max 255

kalian bisa check validation di halaman ini tapi yang paling penting kita bisa validasi dikit" dulu yaa kayak required dan max

selanjutnya kita minta model dengan query builder untuk create datanya

lalu return redirect ke halaman index

seharusnya jika sudah maka data berhasil di insert

#	Title	Deskripsi	Created	
1		Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit?	2024-06-13 09:19:30	Edit Delete
2	test		2024-06-13 09:19:37	Edit Delete

kalau kita mau nampilin error messagenya gimana?

berikut caranya

menambahkan @error('namafield') is-invalid @enderror di class input supaya munculin warna error menambahkan

```

PHP
@error('namafield')
    {{ $message }}
@enderror

```

untuk memunculkan error message

full codenya sebagai berikut

PHP

```
@extends('template')

@section('content')
<h1>Create Notes</h1>
<form action="/store" method="POST" enctype="multipart/form-data">
    @csrf
    <div class="mb-3">
        <label for="title" class="form-label @error('title') is-invalid
@enderror">Title</label>
        <input type="text" class="form-control" id="title" placeholder="title"
name="title">
        @error('title')
            <div class="invalid-feedback m-1">
                {{ $message }}
            </div>
        @enderror
    </div>
    <div class="mb-3">
        <label for="body" class="form-label @error('title') is-invalid
@enderror">Body</label>
        <textarea class="form-control" id="body" rows="3" name="body"></textarea>
        @error('title')
            <div class="invalid-feedback m-1">
                {{ $message }}
            </div>
        @enderror
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
</form>
@endsection
```

kadang ada case kita udah input title tapi lupa input body atau salah satu field, tapi inputan value kita ilang seperti ini

Create Notes

Title

Body

The body field is required.

gimana caranya? kasih value="{{ old('namafield') }}"

khusus text area langsung di dalamnya ya karena tidak ada valuenya untuk text area

PHP

```
@extends('template')

@section('content')
<h1>Create Notes</h1>
<form action="/store" method="POST" enctype="multipart/form-data">
    @csrf
    <div class="mb-3">
        <label for="title" class="form-label @error('title') is-invalid
@enderror">Title</label>
        <input type="text" class="form-control" id="title" placeholder="title"
name="title"
        value="{{ old('title') }}">
        @error('title')
        <div class="invalid-feedback m-1">
            {{ $message }}
        </div>
        @enderror
    </div>
    <div class="mb-3">
        <label for="body" class="form-label @error('body') is-invalid
@enderror">Body</label>
        <textarea class="form-control" id="body" rows="3" name="body">{{ old('body')
}}</textarea>
        @error('body')
        <div class="invalid-feedback m-1">
            {{ $message }}
        </div>
        @enderror
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
</form>
@endsection
```

Title

adsfa

Body

The body field is required.

Submit

selanjutnya edit dan update yang harus kita siapin yaitu, route, method nampilin views, method updatenya dan link editnya

pertama link updatenya

di index.blade.php

ubah button edit

PHP

```
<a href="/edit/{{ $note->id }}" class="btn btn-warning">Edit</a>
```

buat routesnya

tandanya kita minta parameter id

PHP

```
Route::get('/edit/{note:id}', [NoteController::class, 'edit']);
```

method edit di NoteController

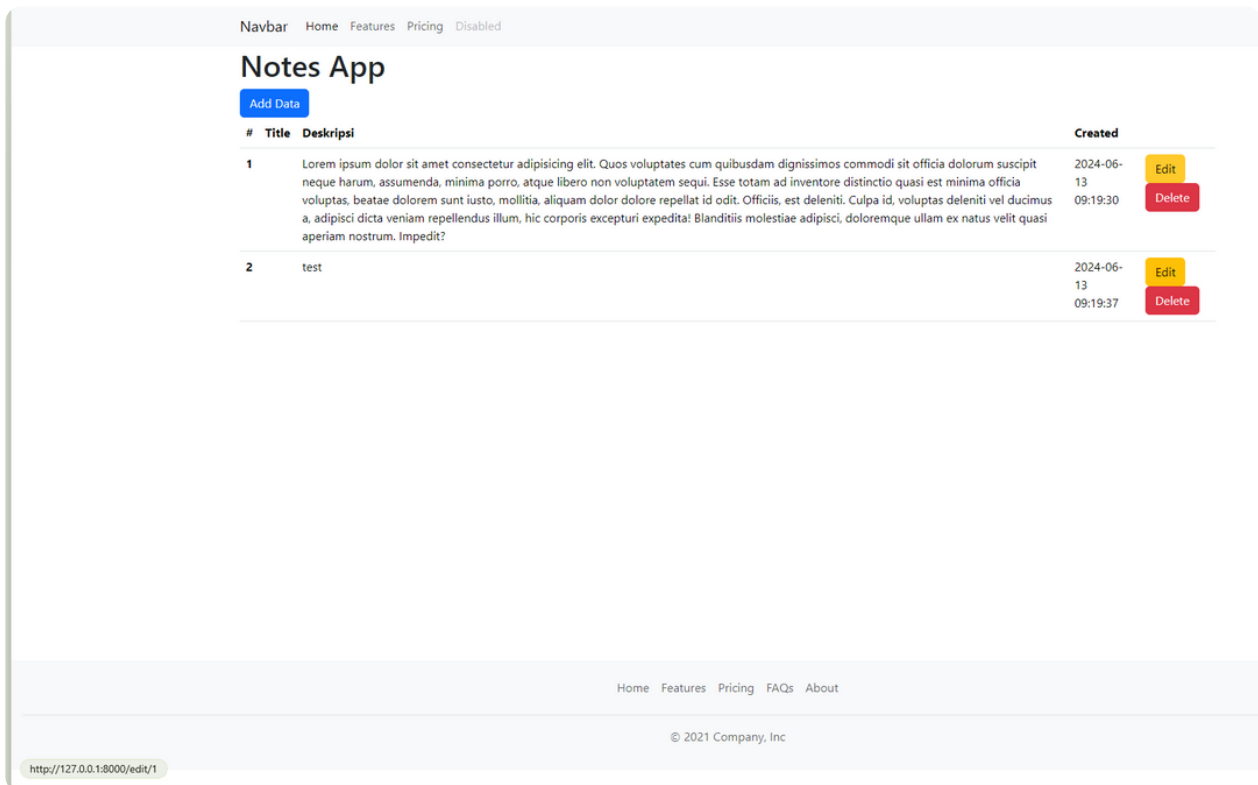
PHP

```
public function edit(Note $note)
{
    return view('edit', ['note' => $note]);
}
```

view edit, dengan passing datanya notenya

jangan lupa buat edit.blade.php → duplicate saja dan rename dari create

seharusnya kalau kita hover button edit



akan muncul link berikut , kalau udah pencet

ubah menjadi edit notes anehnya kita belum tampilin value awalnya, gimana caranya?

PHP

```
<input type="text" class="form-control" id="title" placeholder="title" name="title"
      value="{{ old('title', $note->title) }}">
```

Edit Notes

Title

Notes 1

Body

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit?

Submit

aku baru sadar... titlenya seharusnya Tnya gede ges jadi kita jalanin migration ulang yaa

`$note->title` di views edit sama index

jangan lupa untuk mengubah action ke path update dengan `@method('put')`

```
<h1>Edit Notes</h1>
<form action="/update/{{ $note->id }}" method="POST" enctype="multipart/form-data">
    @method('put')
    @csrf
    {{-- {{ dd($note) }} --}}
```

siapkan routes put sesuai dengan action yang kita buat

PHP

```
Route::put('/update/{note:id}', [NoteController::class, 'update']);
```

method updatenya gimana?

PHP

```
public function update(Request $request, Note $note)
{
    $validatedData = $request->validate([
        'title' => 'required|max:255',
        'body' => 'required',
    ]);

    Note::where('id', $note->id)
        ->update($validatedData);
    return redirect('/');
}
```

seharusnya bisa

Notes App

Add Data		
#	Title	Deskripsi
1	Notes 1 update	Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit?
		Created 2024-06-13 15:24:00
		<button>Edit</button> <button>Delete</button>

gimana dengan delete?

pertama kita siapkan buttonnya terlebih dahulu

di index.blade.php

PHP

```
<form action="/delete/{{ $note->id }}" method="post" class="d-inline">
    @method('delete')
    @csrf
    <button class="btn btn-danger w-100" onclick="return confirm('Are You sure?')">
<span
    data-feather="x-circle"></span> Delete</button>
</form>
```

siapkan routenya

PHP

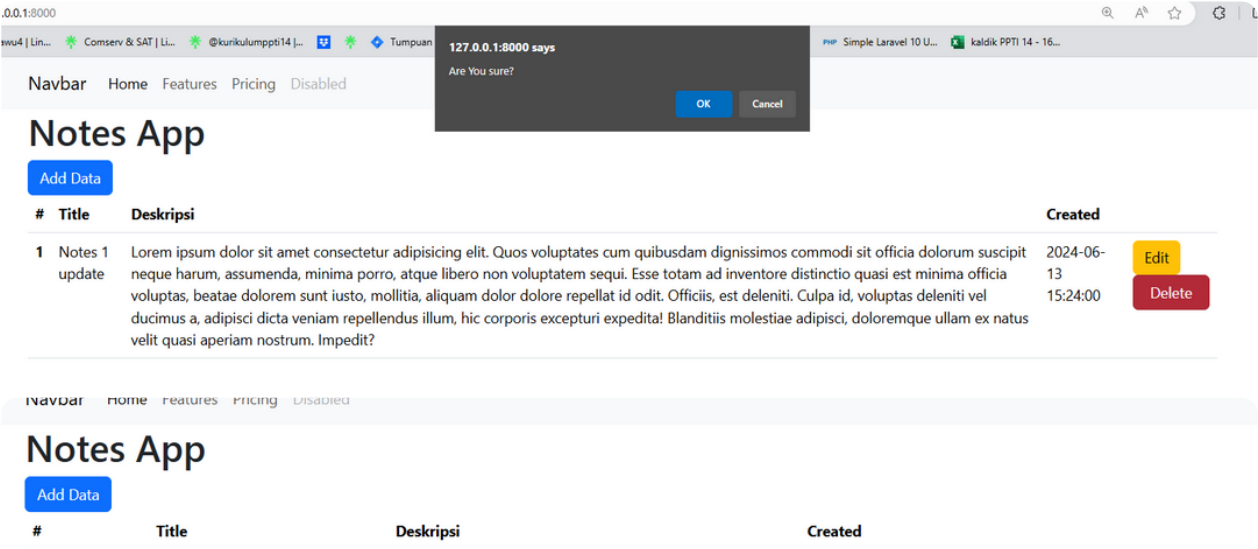
```
Route::delete('/delete/{note:id}', [NoteController::class, 'destroy']);
```

method controllernya yaitu

```
PHP

public function destroy(Note $note)
{
    Note::destroy($note->id);
    return redirect('/');
}
```

jika kita coba maka



data berhasil di delete

kita bisa menambahkan if supaya jika tidak ada data kita bisa tampilin text atau apa gitu dengan cara seperti berikut

```

@extends('template')

@section('content')
    <h1>Notes App</h1>
    <a href="/create" class="btn btn-primary">Add Data</a>

    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Title</th>
                <th scope="col">Deskripsi</th>
                <th scope="col">Created</th>
                <th scope="col"></th>
            </tr>
        </thead>
        @if ($notes->isEmpty())
            <tbody>
                @foreach ($notes as $note)
                    <tr>
                        <th scope="row">{{ $note->id }}</th>
                        <td>{{ $note->title }}</td>
                        <td>{{ $note->body }}</td>
                        <td>{{ $note->created_at }}</td>
                        <td>
                            <a href="/edit/{{ $note->id }}" class="btn btn-
warning">Edit</a>
                            <form action="/delete/{{ $note->id }}" method="post"
class="d-inline">
                                @method('delete')
                                @csrf
                                <button class="btn btn-danger w-100" onclick="return
confirm('Are You sure?')"><span
                                data-feather="x-circle"></span>
Delete</button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        @else
            <div class="row my-3 text-center">
                <h2>No Notes found</h2>
            </div>
        @endif
    </table>
@endsection

```

kalau datanya banyak gimana cara pakai paginate?

pertama kita siapkan data 10 terlebih dahulu kita bisa menggunakan seeder

awalnya notes seeder kita seperti ini

```
PHP
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class NoteSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        DB::table('notes')->insert([
            'Title' => 'Notes 1',
            'Body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit?',
            'created_at' => now(),
            'updated_at' => now(),
        ]);
    }
}
```

kalau lebih gimana?

buat array lagi seperti berikut

PHP

```
DB::table('notes')->insert([
    [
        'Title' => 'Notes 1',
        'Body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit.
Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque
harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad
inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto,
mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id,
voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic
corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus
velit quasi aperiam nostrum. Impedit?',
        'created_at' => now(),
        'updated_at' => now(),
    ],
    [
        'Title' => 'Notes 2',
        'Body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit.
Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque
harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad
inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto,
mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id,
voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic
corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus
velit quasi aperiam nostrum. Impedit?',
        'created_at' => now(),
        'updated_at' => now(),
    ]
]);
```

tugas buat kan 12 data

kalian bisa menggunakan faker jika mau

selanjutnya mengubah query builder kita sehingga bisa menggunakan paginate

PHP

```
public function index()
{
    $notes = Note::paginate(10);
    return view('index', ['notes' => $notes]);
}
```

kalian juga bisa memodifikasi yaa pakai where dan temen"

panggil links/paginate di view index kita dibagian bawah table

PHP

```
{{ $notes->links() }}
```

hasilnya jelek temen" karena belum menggunakan bootstrap, caranya gimana supaya pake yang bootstrap?

10	Notes	Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit?	2024-06-13 15:55:33	<button>Edit</button> <button>Delete</button>
----	-------	--	------------------------	--

[« Previous](#)

[Next »](#)

Showing 1 to 10 of 12 results



buka AppServiceProvider.php

```

PHP
<?php

namespace App\Providers;

use Illuminate\Pagination\Paginator;
use Illuminate\Support\ServiceProvider;

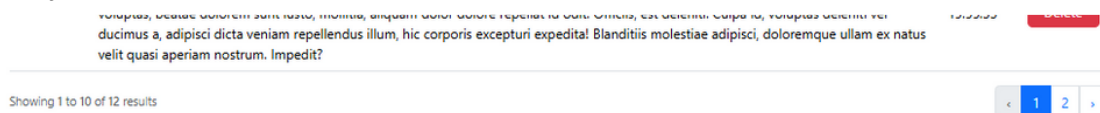
class AppServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     */
    public function register(): void
    {
        //
    }

    /**
     * Bootstrap any application services.
     */
    public function boot(): void
    {
        Paginator::useBootstrapFive();
    }
}

```

import pagination paginator dan tambahkan paginator bootstrapfive() di boot

seharusnya sudah bisa



terakhir, gimana cara relationship data? sebagai contoh sebuah notes memiliki sebuah category yang reference table categorynya

pertama kita harus siapkan schemanya

buka migration notes

tambahkan column baru

```

PHP
$table->unsignedBigInteger('idKategori');

```

buat migration model dan seeder kategori

PHP

```
php artisan make:model Kategori -m -s
```

modifikasi migration

PHP

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('kategoris', function (Blueprint $table) {
            $table->id();
            $table->string('kategori');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('kategoris');
    }
};
```

update model

PHP

<?php

```
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class kategori extends Model
{
    use HasFactory;
    protected $table = 'kategoris';
    protected $fillable = ['kategori'];
    protected $primaryKey = 'id';
}
```

buat migration lagi untuk membuat relation antara notes dan kategoris

PHP

php artisan make:migration referencekategoris

PHP

<?php

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('notes', function (Blueprint $table) {
            $table->foreign('idKategori')->references('id')->on('kategoris');
        });
    }

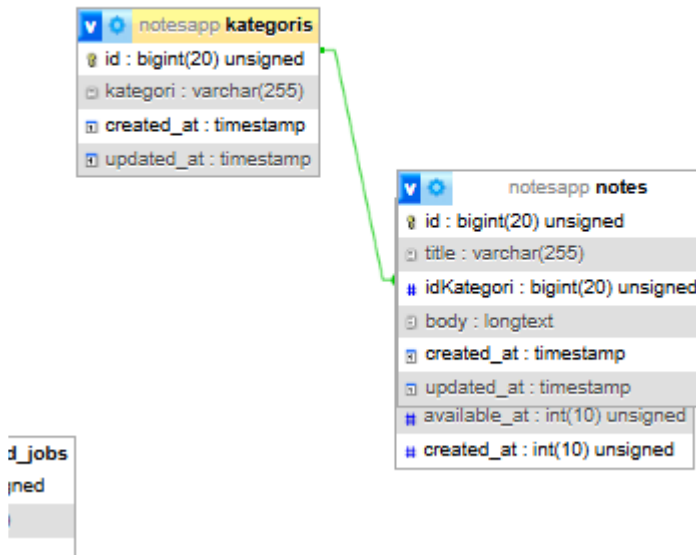
    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        //
    }
};
```

intinya kita minta laravel buat tambahkan foreign key di idkategori yang references id di table kategori

PHP

```
php artisan migrate:fresh
```

seharusnya di table views kita sudah ada relation nya



selanjutnya apa? kita info ke masing" model kalau kita ada references
notemodel

PHP

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

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

```
use Illuminate\Database\Eloquent\Relations\BelongsTo;
```

```
class Note extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table = 'notes';
```

```
    protected $fillable = ['title', 'body'];
```

```
    protected $primaryKey = 'id';
```

```
    public function kategori(): BelongsTo
```

```
    {
```

```
        return $this->belongsTo(Kategori::class, 'id');
```

```
    }
```

```
}
```

aku ada kesalahan ya temen" seharusnya Kategori::class model Kategori besar
intinya kita kasih tau bahwa satu note itu belongsto 1 to kategori

```
PHP
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;

class kategori extends Model
{
    use HasFactory;
    protected $table = 'kategoris';
    protected $fillable = ['kategori'];
    protected $primaryKey = 'id';

    public function notes(): HasMany
    {
        return $this->hasMany(Note::class, 'idKategori');
    }
}
```

intinya kita kasih tau kalau 1 kategori di miliki banyak notes, dan referencenya idKategori

gimana kita testnya kalau bener? lakuin seed kategori dulu lalu seed notesnya dengan tambahan kategorisnya

PHP

<?php

```
namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class KategoriSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        DB::table('kategoris')->insert([
            [
                'kategori' => 'kategori 1',
            ],
            [
                'kategori' => 'kategori 2',
            ],
            [
                'kategori' => 'kategori 3',
            ],
        ]);
    }
}
```

call di database seeder

PHP

```
$this->call([
    KategoriSeeder::class,
    NoteSeeder::class,
]);
```

modifikasi modelnya supaya idKategori menjadi fill able

PHP

```
protected $fillable = ['title', 'body', 'idKategori'];
```

tambahkan idKategori di seeder

PHP

```
'idKategori' => rand(1,3),
```

random saja

migrate dan seed

PHP

```
php artisan migrate:fresh --seed
```

tampilkan idkategorinya di index

PHP

```
{{ $note->idKategori }}
```

sudah bisaa

Notes App

Add Data

| # | Title | Deskripsi | Kategori | Created | |
|---|---------|--|----------|---------------------|-----------------------------------|
| 1 | Notes 1 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit? | 3 | 2024-06-13 16:29:29 | <div>Edit</div> <div>Delete</div> |
| 2 | Notes 2 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit? | 2 | 2024-06-13 16:29:29 | <div>Edit</div> <div>Delete</div> |
| 3 | Notes 3 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adioisci, doloremque ullam ex natus velit quasi aperiarn nostrum. Impeedit? | 1 | 2024-06-13 16:29:29 | <div>Edit</div> <div>Delete</div> |

tapi kan yang mau di tampilin nama kategorinya... gimana caranya?

seharusnya

PHP

```
<td {{ $note->fungsidimodel>fieldnya }}</td>
```

PHP

```
<td {{ $note->kategori->kategori }}</td>
```

Notes App

Add Data

| # | Title | Deskripsi | Kategori | Created | |
|---|---------|--|------------|---------------------|--|
| 1 | Notes 1 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit? | kategori 2 | 2024-06-13 16:56:32 | <button>Edit</button>
<button>Delete</button> |
| 2 | Notes 2 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit? | kategori 2 | 2024-06-13 16:56:32 | <button>Edit</button>
<button>Delete</button> |
| 3 | Notes 3 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit? | kategori 1 | 2024-06-13 16:56:32 | <button>Edit</button>
<button>Delete</button> |
| 4 | Notes 4 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit? | kategori 2 | 2024-06-13 16:56:32 | <button>Edit</button>
<button>Delete</button> |
| 5 | Notes | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia | kategori | 2024-06- | <button>Edit</button> |

oiya created atnya jelek temen" kita bisa pakai ini →diffForHumans()

PHP

```
<td>{{ $note->created_at->diffForHumans() }}</td>
```

maka jadi begini

Add Data

| # | Title | Deskripsi | Kategori | Created | |
|---|---------|--|------------|--------------|--|
| 1 | Notes 1 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum suscipit neque harum, assumenda, minima porro, atque libero non voluptatem sequi. Esse totam ad inventore distinctio quasi est minima officia voluptas, beatae dolorem sunt iusto, mollitia, aliquam dolor dolore repellat id odit. Officiis, est deleniti. Culpa id, voluptas deleniti vel ducimus a, adipisci dicta veniam repellendus illum, hic corporis excepturi expedita! Blanditiis molestiae adipisci, doloremque ullam ex natus velit quasi aperiam nostrum. Impedit? | kategori 2 | 1 minute ago | <button>Edit</button>
<button>Delete</button> |
| 2 | Notes | Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptates cum quibusdam dignissimos commodi sit officia dolorum | kategori | 1 minute | <button>Edit</button> |

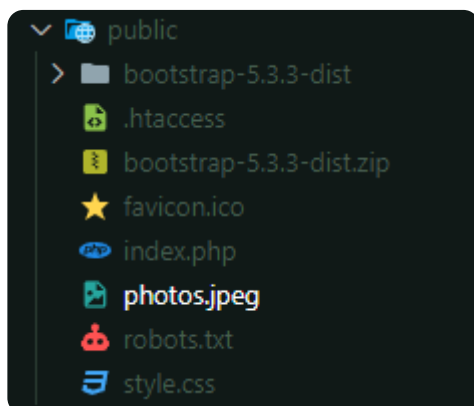
cara tampilin gambar gimana?

PHP

```

```

pastikan image berada di public yaa





taraa bisa

finish, whats next?

- factory
- custom validate
- insert image
- component laravel

June 14, 2024, 00:04 June 14, 2024