

Mempelajari Dasar-dasar Laravel (A to Z)

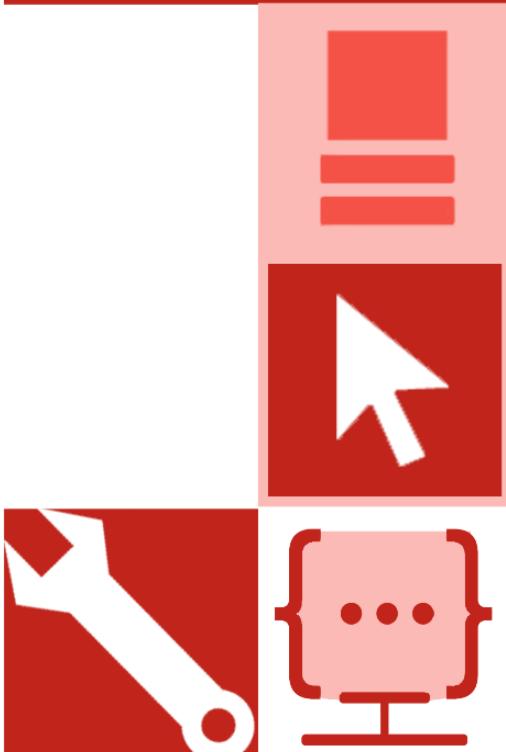
BAB 1 Client Side

- HTML
- CSS
- Bootstrap
- Javascript



BAB 2. Server Side

- PHP
- OOP PHP
- SQL Database



BAB 3 Laravel

- Laravel MVC
- Laravel Templating
- Laravel Migration
- Laravel CRUD Query Builder
- Laravel CRUD ORM + File Upload
- Laravel Auth
- Laravel Relationship



Proyek Akhir: Membuat Aplikasi Portal Berita Sederhana



Website yang berisi informasi berita terbaru dan terkini dalam berbagai topik seperti politik, bisnis, teknologi, dll.

Situs web ini dapat menampilkan berita dalam berbagai format seperti teks, gambar, video. Web ini memiliki fitur pencarian, komentar, dan bagikan.

Sertifikat Penyelesaian

Sertifikat bagi peserta yang membuat proyek akhir yang disediakan di buku.

Ada Tugas & Kunci Jawaban

Tidak hanya materi, tersedia tugas untuk praktik, jika ada kesulitan terdapat kunci jawaban di akhir buku.

Full Source Code Proyek

Disediakan full source code proyek yang diberikan untuk jadi referensi.

Gratis Download Versi Terbaru

Dalam 1 tahun setelah pembelian e-book jika terdapat update teknologi dan dibuat versi terbaru e-book, dapat diakses gratis.



Laravel

Mempelajari dasar dasar laravel A-Z

versi : 1.0

Tanggal Diterbitkan: 1 April 2023

Ditulis oleh Rezky Putra Hakkif

© 2023 Sanbercode

Hak cipta dilindungi Undang-undang. Tidak ada bagian dari manuskrip buku yang boleh diproduksi ulang, disimpan dalam sistem pengambilan, atau ditransmisikan dalam bentuk apa pun atau dengan cara apa pun di luar jumlah salinan yang dibeli, kecuali untuk cadangan tunggal atau salinan arsip.

Penulis dan penerbit telah berhati-hati dalam mempersiapkan buku ini, tetapi tidak memberikan jaminan tersurat maupun tersirat dalam bentuk apa pun dan tidak bertanggung jawab atas kesalahan atau kelalaian. Tidak ada tanggung jawab yang diasumsikan untuk kerusakan insidental atau konsekuensial sehubungan dengan atau yang timbul dari penggunaan wadah informasi atau program di sini.

Diterbitkan di Bandung, Jawa Barat oleh Sanbercode.

DAFTAR ISI

DAFTAR ISI.....	1
Pendahuluan.....	6
Kata Pengantar.....	6
Kepada Siapa Ebook Ini Ditujukan.....	6
Cakupan Ebook.....	6
Bagian 1 Client Side.....	9
Bab 1 HTML.....	9
Sejarah HTML.....	9
jadi apa itu HTML?.....	10
Struktur Dasar Penulisan HTML.....	11
element HTML.....	11
element pada HTML.....	11
Nested Element (element bersarang) di HTML.....	13
beberapa element di HTML.....	13
Headings.....	13
Paragraf.....	14
Line breaks.....	14
Label.....	15
List.....	15
Links.....	17
Input type HTML.....	17
HTML Input Attribute.....	21
Element TextArea.....	22
Element Select.....	23
FORM.....	24
Form Element.....	24
Form Attribute.....	25
Bab 2 CSS.....	27
Sejarah CSS.....	27
jadi apa itu CSS?.....	27
Selektor , Property dan Value.....	27
Cara Penulisan Kode CSS dalam HTML.....	28
Selector Di CSS.....	31
CSS Border.....	32
CSS Margin.....	37
CSS Padding.....	40
Box Model.....	43
CSS Position Property.....	44
FlexBox.....	51
Bab 3 Bootstrap.....	56
sejarah Bootstrap.....	56

Apa itu Bootstrap.....	56
Memasang bootstrap.....	57
Komponen Bootstrap.....	60
System Grid Bootstrap.....	61
Typografi.....	66
NavBar.....	67
Form.....	69
Button.....	72
Card.....	74
Carousel.....	77
Alerts.....	80
Bab 4 Javascript.....	84
sejarah Javascript.....	84
jadi apa itu Javascript?.....	85
DOM Javascript.....	85
Selector DOM.....	86
document.getElementById().....	86
Manipulasi DOM.....	88
DOM Events.....	89
Variable.....	91
Data Type.....	92
Operator Javascript.....	92
String.....	94
Function.....	102
Kondisional dengan if / else if / else.....	105
Looping.....	114
Array Javascript.....	119
Array Method.....	121
Bagian 2 Server Side.....	128
BAB 5 PHP.....	128
Sejarah PHP.....	128
Apa itu PHP?.....	128
Mempersiapkan PHP.....	129
Menyalakan XAMPP.....	129
Mulai Menulis PHP.....	132
Variabel dan titik koma ;.....	134
Output Standar.....	135
Data Types PHP.....	136
String Method.....	136
Tipe Data Array.....	139
Array.....	141
Array Asosiatif.....	141
Array Multidimentional.....	142
Function PHP.....	144

Conditional PHP.....	146
Looping PHP.....	148
BAB 6 OOP PHP.....	152
\$this OOP PHP.....	152
Intansiasi Object.....	153
Construct.....	154
Setter dan Getter.....	155
Property (properti).....	156
Method (metode).....	158
inheritance.....	159
BAB 7 SQL Database.....	161
database.....	161
database SQL dan non SQL.....	161
Database SQL.....	162
Sintax Database.....	165
Table SQL.....	166
Insert Into.....	169
Select.....	171
Order By.....	174
Join.....	175
Update.....	177
Delete.....	178
Min, Max, Sum, Avg.....	178
Max.....	178
Min.....	179
SUM.....	180
AVG.....	181
BAB 8 Command Line + Git.....	183
Command Line.....	183
Apa itu Git ?.....	187
Sejarah Git.....	187
sedangkan apa itu gitlab.....	188
Perintah Dasar Git.....	188
Latihan.....	190
1. Petunjuk Pengeraan.....	190
2. Clone repository.....	193
3. Commit Pertama.....	194
Bagian 3 Membuat aplikasi portal berita dengan Laravel.....	196
BAB 9 pengenalan Laravel dan Laravel MVC.....	196
Apa itu Laravel?.....	196
Sejarah Laravel.....	196
Laravel MVC.....	197
requirement Laravel 9.....	198
instalasi Laravel.....	198

Install Composer.....	198
Mengecek Composer.....	198
Membuat Project Laravel.....	199
fungsi file dan folder di laravel.....	201
Menjalankan Laravel.....	202
Routing Laravel.....	203
Named Routes.....	207
Controller.....	209
Route Parameter.....	210
Passing Data.....	213
BAB 10 Laravel Templating.....	218
Blade Engine Template.....	218
Templating Portal Berita.....	220
BAB 11 Laravel Migration.....	234
Apa itu Laravel Migration.....	234
file .env.....	235
Data Tipe Migration.....	236
Jenis Jenis indeks di Laravel Migration.....	237
Latihan.....	238
deskripsi table.....	240
Buat Migration.....	243
BAB 12 Laravel CRUD Query Builder.....	259
CRUD.....	259
Laravel Validation.....	261
Latihan.....	261
Create Data.....	265
Read Data.....	271
Update Data.....	276
Delete Data.....	284
BAB 13 Laravel CRUD ORM + File Upload.....	291
Apa itu ORM.....	291
Fitur-Fitur Eloquent ORM.....	291
Latihan.....	293
Persiapan.....	293
Create Data.....	304
Read Data.....	311
Update Data.....	315
Delete Data.....	325
Bab 14 Laravel Authetication.....	327
Persiapan.....	327
Register.....	342
Login.....	351
Dashboard.....	353
Halaman Dashboard.....	354

Halaman About.....	358
Edit Profile.....	361
Middleware.....	369
BAB 15 Laravel ORM Relationship.....	377
One to One.....	377
One To Many.....	382
Many to Many.....	396
Tantangan Bersertifikat.....	414
Teknis Pengumpulan.....	416

Pendahuluan

Kata Pengantar

Laravel adalah sebuah framework PHP yang populer dan sering digunakan untuk membangun aplikasi web. Framework ini dikembangkan oleh Taylor Otwell pada tahun 2011 dan sejak saat itu menjadi salah satu framework PHP paling populer di dunia. Laravel menawarkan berbagai fitur dan komponen yang sangat berguna dan dapat mempercepat pengembangan aplikasi web.

Pada ebook Laravel ini Kamu akan mempelajari bagaimana cara membuat Aplikasi web full-stack menggunakan menggunakan laravel

Setelah menyelesaikan ebook Laravel ini, Kamu akan dapat membuat aplikasi Web Custom menggunakan laravel

Kepada Siapa Ebook Ini Ditujukan

Ebook ini ditujukan kepada seseorang yang tertarik terjun dunia pemrograman bahkan yang belum memiliki pengalaman pemrograman sebelumnya dan tertarik untuk membangun web dengan cepat menggunakan framework Laravel.

Cakupan Ebook

Ebook Laravel ini akan membahas berbagai hal yang dapat membantu Kamu untuk menjadi seorang Web Developer menggunakan Laravel . Ebook ini dibagi menjadi beberapa topik besar yang disebut sebagai Bagian, kemudian beberapa topik detail yang disebut Bab, serta Latihan yang berguna untuk menguji pemahaman Kamu pada setiap topik yang dibahas.

Ebook ini dapat dibagi menjadi 3 Bagian, 16 Bab ,Masing-masing pembagian tersebut disesuaikan dengan topik-topik atau kata kunci yang akan sering terdengar oleh setiap programmer Laravel . Berikut ringkasan dari masing-masing topik yang akan dibahas.

Bab 1 - HTML, merupakan singkatan dari Hyper Text Markup Language. HTML adalah sebuah bahasa standar untuk pembuatan halaman web. atau bisa dibilang HTML adalah struktur dasar.

Bab 2 - CSS, merupakan singkatan dari (Cascading Style Sheets) adalah bahasa pemrograman yang digunakan untuk mengatur tampilan atau styling pada halaman web

Bab 3 - Bootstrap, merupakan framework css yang memiliki banyak component dan fitur yang bisa mempermudah kita untuk melakukan styling pada halaman web

Bab 4 - Javascript, adalah bahasa pemrograman yang berfungsi untuk membuat website kita lebih dinamis, disini kita akan belajar tentang data type, variabel, looping, conditional di Javascript

Bab 5 - PHP, merupakan bahasa pemrograman yang berjalan di sisi server. dan bahasa yang akan sering kita gunakan di framework laravelnya nanti, di bab ini kita akan belajar tentang data type, variabel, looping, conditional, dan array di PHP

Bab 6 - OOP PHP, OOP merupakan (Object-Oriented Programming) atau Pemrograman Berorientasi Objek adalah paradigma pemrograman yang berfokus pada konsep objek, dimana setiap objek mempunyai properti dan method yang dapat digunakan untuk memanipulasi data. di bab ini kita akan mempelajari konsep OOP di PHP

Bab 7 - SQL, merupakan bahasa basis data tempat kita menyimpan data kita atau penyimpanan elektronik kita, di bab ini kita akan belajar tentang query dasar pada SQL

Bab 8 - Command Line Interface (CLI) + Git, merupakan antarmuka untuk pengelolaan file. Pada bab ini akan dibahas terkait apa itu CLI, apa saja fungsi nya dan bagaimana cara menggunakan CLI sehingga dapat mengelola file menjadi lebih baik dan juga kita belajar git yaitu sistem yang umum digunakan oleh para programmer untuk mendokumentasikan perkembangan aplikasi dan juga berkolaborasi dengan programmer lainnya.

Bab 9 - Pengenalan Laravel dan Laravel MVC, laravel merupakan framework PHP terpopuler di dunia, di bab ini kita akan berkelantern tentang laravel dan konsep MVC di dalamnya (Model, View, Controller)

Bab 10 - Templating Laravel, di bab ini kita akan latihan memasang template pada laravel menggunakan blade engine di laravel

Bab 11 - Laravel Migration, di bab ini kita akan belajar tentang konsep Migration di Laravel yaitu cara kita memanipulasi table pada databasenya

Bab 12 - Laravel CRUD Query Builder, di bab ini kita akan belajar tentang CRUD(Create, Read, Update, Delete) di dalam laravel menggunakan konsep Query Builder

Bab 13 - Laravel CRUD ORM + File Upload, di bab ini kita akan masih belajar tentang CRUD tapi menggunakan konsep ORM Eloquent yaitu model di laravelnya dan melakukan upload gambar

Bab 14 - Laravel Auth, di bab ini kita akan membuat fitur authentication (Login, Register, dan Logout) pada laravel

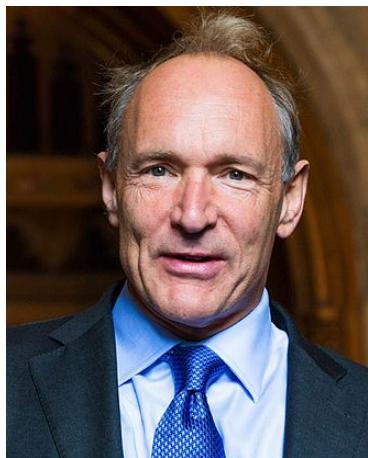
Bab 15 - Laravel ORM Relationship, di bab ini kita akan belajar memasang relasi pada modelnya menggunakan ORM Relationship Laravel

Tantangan Bersertifikat , di bagian ini akan diberikan latihan untuk menguji pemahaman materi yang telah sampaikan dengan menambahkan fitur di aplikasi portal berita. Penggeraan tantangan akan mendapatkan sertifikat

Bagian 1 Client Side

Bab 1 HTML

Sejarah HTML



sumber gambar wikipedia (https://en.wikipedia.org/wiki/Tim_Berners-Lee)

HTML pertama kali dikembangkan pada tahun 1989 oleh Tim Berners-Lee, seorang ilmuwan komputer dari CERN (Organisasi Eropa untuk Penelitian Nuklir). Berners-Lee ingin membuat sebuah sistem untuk membagikan dan mengakses informasi secara online di antara para peneliti di seluruh dunia. Dia mengembangkan HTML sebagai bagian dari proyek World Wide Web, yang merupakan protokol untuk berbagi dokumen secara online melalui internet.

Versi pertama HTML, HTML 1.0, diterbitkan pada tahun 1993. Versi ini masih sangat sederhana, hanya memiliki sekitar 20 elemen HTML dasar seperti `<p>` (untuk paragraf), `<h1>` (untuk heading), dan `<a>` (untuk hyperlink). Kemudian pada tahun 1995, HTML 2.0 diterbitkan, yang menambahkan beberapa elemen baru seperti `<table>` dan `<form>`.

HTML 3.2 diterbitkan pada tahun 1997 dan merupakan versi HTML pertama yang disetujui oleh World Wide Web Consortium (W3C) sebagai standar web. Versi ini menambahkan lebih banyak elemen HTML dan fitur seperti frame dan tabel berlapis.

HTML4, yang merupakan versi HTML yang paling banyak digunakan pada masa lalu, diterbitkan pada tahun 1999. HTML4 menambahkan kemampuan untuk membuat tampilan web yang lebih kompleks dan interaktif dengan menggunakan CSS (Cascading Style Sheets) dan JavaScript.

Versi terbaru HTML adalah HTML5, yang diterbitkan pada tahun 2014. HTML5 memiliki beberapa fitur baru seperti canvas, video, dan audio, dan memungkinkan pengembang web untuk membuat aplikasi web yang lebih kompleks dan responsif. HTML5 juga memperkenalkan beberapa elemen dan atribut baru seperti <header>, <footer>, dan <nav>.

Selama beberapa dekade terakhir, HTML telah menjadi dasar dalam pengembangan web dan memungkinkan para pengembang untuk menciptakan berbagai jenis situs web dari yang sederhana hingga yang kompleks. HTML terus berkembang dan disesuaikan dengan perkembangan teknologi web yang baru, seperti mobile dan responsive design, untuk memastikan bahwa situs web tetap berfungsi dengan baik di semua perangkat dan layar yang berbeda.

jadi apa itu HTML?

HTML merupakan singkatan dari Hyper Text Markup Language. HTML adalah sebuah bahasa standar untuk pembuatan halaman web. atau bisa dibilang HTML adalah struktur dasar. sebagai contoh kita bisa menagap website sebagai sebuah bagunan dimana HTML berperan sebagai pondasi dasar dari sebuah bangunan tersebut dan CSS untuk mempercantik tampilan dari bagunan seperti memberi warna, posisi dll, kemudian Javascript sebagai gerak dari semua yang ada dalam bangunan tersebut seperti gerak dari pintu ditarik atau di dorong

Struktur Dasar Penulisan HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

- element <!DOCTYPE html> mengartikan bahwa dokumen ditulis dengan versi HTML5
- elemen <html> adalah elemen induk atau elemen utama yang sering disebut juga root element dari sebuah halaman HTML.
- elemen <head> berisi informasi tentang halaman HTML yang sedang dibuat
- elemen <title> adalah judul dari halaman HTML yang akan tampil di tab browser.
- elemen <body> merupakan elemen yang akan ditampilkan ke halaman web

element HTML

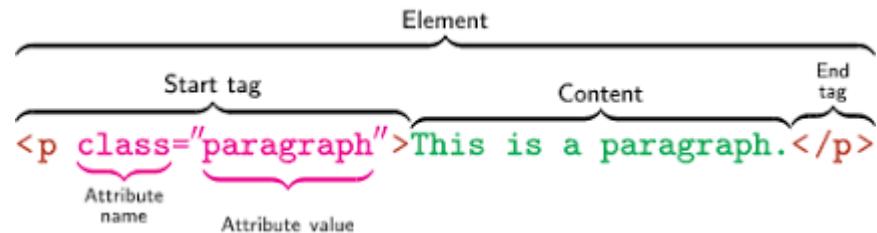
element pada HTML

Elemen atau sering disebut dengan tag pada HTML adalah penyusun dari sebuah halaman web yang dapat dikenali oleh browser dan ditampilkan oleh browser sesuai karakteristik masing-masing.

terdapat 2 macam element pertama **element dengan tag penutup** dan **element yang tanpa tag penutup(empty element)**

element dengan tag

Sebuah elemen HTML terdiri dari tag pembuka(open tag), atribut(attribute), isi(content), dan tag penutup(close tag).



pada ilustrasi di atas merupakan sebuah elemen paragraph dibuka dengan tag pembuka `<p>` . Di dalam tag pembuka tersebut terdapat atribut `class` yang disematkan ke dalam tag. Isi atau content dari sebuah tag `paragraph` `<p>` akan ditampilkan sebagai paragraf. lalu terakhir sebuah tag ditutup dengan sebuah tag penutup `</p>`.

element tanpa tag penutup(empty element)

beberapa tag yang disebut dengan empty element seperti :

- `
` berfungsi untuk membuat baris baru
- `<input>` berfungsi untuk membuat inputan
- `<link>` berfungsi untuk memanggil external content/file
- `` berfungsi untuk menampilkan sebuah gambar
- dll

sumber : <https://www.geeksforgeeks.org/what-are-empty-elements-in-html/>

Nested Element (element bersarang) di HTML

nested element di HTML adalah cara kita memasukan element di dalam element didalam HTML

contoh

```
<ul>
    <li>List 1</li>
    <li>List 2</li>
    <li>List 3</li>
</ul>
```

Pada contoh di atas terdapat element `` yang menbungkus element `` didalamnya

beberapa element di HTML

Headings

Heading HTML didefinisikan dengan tag `<h1>` hingga `<h6>` dengan cara penulisan menggunakan tag penutup dengan `h1` sebagai level tertinggi dan `h6` paling rendah yang digunakan untuk membuat sebuah judul, atau sub judul yang akan di tampilkan di web browsernya

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
```

dimana jika nilai dari heading lebih rendah `<h1>` maka ukuran font nya akan lebih besar sebaliknya jika nilai heading nya lebih besar `<h6>` maka ukuran fontnya akan lebih kecil

Heading level 1

Heading level 2

Heading level 3

Heading level 4

Heading level 5

Heading level 6

Paragraf

Paragraf HTML didefinisikan dengan tag `<p>` dengan cara penulisan menggunakan tag penutup yang digunakan untuk membuat sebuah tampilan paragraf pada web browser

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed a quam libero. In interdum vitae odio eu mattis. Sed at lorem vel sapien eleifend interdum in vel est.</p>
```

```
<p>Vivamus in quam ut neque tristique bibendum tempor vitae tellus. Suspendisse auctor mauris sit amet ex tincidunt molestie. Integer ac risus rhoncus, pharetra odio nec, auctor erat. Integer cursus nisi nec convallis dapibus. </p>
```

Lore ipsum dolor sit amet, consectetur adipiscing elit. Sed a quam libero. In interdum vitae odio eu mattis. Sed at lorem vel sapien eleifend interdum in vel est.

Vivamus in quam ut neque tristique bibendum tempor vitae tellus. Suspendisse auctor mauris sit amet ex tincidunt molestie. Integer ac risus rhoncus, pharetra odio nec, auctor erat. Integer cursus nisi nec convallis dapibus.

Line breaks

Line Breaks didefinisikan dengan tag `
` atau `
` dengan cara penulisan menggunakan tanpa tag penutup yang digunakan untuk membuat baris baru

```
<p> Hallo Semua <br /> Nama Saya Adalah <br /> Rezky Putra  
Akkif</p>
```

Hallo Semua
Nama Saya Adalah
Rezky Putra Akkif

dari contoh diatas kita dapat menggunakan nested element pada penulisan sebuah paragraf yang ingin dibuat baris baru di tiap kalimatnya

Label

Label didefinisikan dengan tag `
` atau `
` dengan cara penulisan menggunakan tag penutup yang mewakili keterangan untuk item dalam tampilan web biasanya digunakan jika kita ingin membuat text sebelum inputan contoh

```
<label for="fname">First name:</label><br>  
<input type="text" id="fname" name="fname"><br>
```

List

terdapat dua macam list di HTML **unordered List** dan **ordered List** yaitu list yang berurut dan list yang tidak berurut

unorganized List

unorganized List(list yang tidak berurut) didefinisikan dengan tag `` dengan cara penulisan menggunakan tag penutup dan menggunakan nested element didalamnya yaitu element `` yang digunakan untuk membuat berapa banyak list yang akan ditampilkan

```
<ul>  
  <li>Makan</li>  
  <li>Minum</li>  
  <li>Ibadah</li>
```

```
</ul>
```

- Makan
- Minum
- Ibadah

ordered List

ordered List(list yang berurut) didefinisikan dengan tag `` dengan cara penulisan menggunakan tag penutup dan menggunakan nested element didalamnya yaitu element `` yang digunakan untuk membuat berapa banyak list yang akan di tampilkan

```
<ol>
  <li>Ardi</li>
  <li>Budi</li>
  <li>Cantika</li>
</ol>
```

1. Ardi
2. Budi
3. Cantika

Links

Link didefinisikan dengan tag `<a>` dengan cara penulisan menggunakan tag penutup digunakan untuk membuat sebuah link

```
<a href="https://sanbercode.com/">sanbercode.com</a>
```

sanbercode.com

dari contoh diatas kita dapat menggunakan attribute yaitu `href` yang bertujuan untuk arah dari link menuju pada saat link nya di klik isi/value dari `href` nya bisa berupa link url atau lokal browsernya contoh `href="subfolder/table.html"`

Input type HTML

Input didefinisikan dengan tag `<input>` dengan cara penulisan menggunakan tanpa menggunakan tag penutup beberapa **element input** berdasarkan **attribute type** yang dapat digunakan

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`

- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text">
- <input type="time">
- <input type="url">
- <input type="week">

disini kita akan mencantoh kan beberapa attribute type pada element input HTML

input type text

digunakan untuk membuat sebuah field inputan

```
<label for="fname">First name:</label><br>
<input type="text" id="fname" name="fname"><br>
```

First name:

rezky

input type number

digunakan untuk membuat sebuah field inputan angka saja

```
<label for="age">Age</label><br>
<input type="number" id="age" name="age"><br>
```

Age

27

Input type password

digunakan untuk membuat sebuah field inputan password

```
<label for="password">Password</label><br>
<input type="password" id="password" name="password"><br>
```

Password

input type radio

digunakan untuk membuat inputan pilihan yang hanya bisa dipilih salah satunya saja

```
<label for="gender">Gender</label><br>
<input type="radio" id="gender" value="1" name="gender">
Male<br>
<input type="radio" id="gender" value="2" name="gender">
Female<br>
```

attribute value berfungsi sebagai nilai dari inputannya

attribute name harus dibuat sama di setiap inputan agar hanya dapat dipilih salah satu saja

Gender

- Male
- Female

input type checkbox

digunakan untuk membuat inputan berupa checklist yang bisa di pilih lebih dari satu

```
<label for="skill">Skill</label><br>
<input type="checkbox" id="skill" value="1" name="skill">
Laravel<br>
<input type="checkbox" id="skill" value="2" name="skill">
VueJS<br>
<input type="checkbox" id="skill" value="3" name="skill">
ReactJs<br>
```

Skill

- Laravel
- VueJS
- ReactJs

input type submit

digunakan untuk membuat inputan input yang berfungsi mentriger attribute action pada tag form

```
<form action="/action_page.php">
<label for="fname">First name:</label><br>
<input type="text" id="fname" name="fname" value="John"><br>
<input type="submit" value="Submit">
</form>
```

First name:

Submit

HTML Input Attribute

value Attribute

Atribut value pada inputan digunakan untuk menentukan nilai awal pada sebuah inputan

```
<label for="fname">First name:</label><br>
<input type="text" id="fname" name="fname" value="Rezky"><br>
```

First name:

Rezky

disabled attribute

attribute disable pada inputan digunakan jika inputan tersebut hanya ingin di tampilkan tanpa harus ubah lagi nilai nya dan nilai/value dari inputan disabled tidak akan terkirim pada saat inputan submit di klik

```
<label for="fname">First name:</label><br>
<input type="text" id="fname" name="fname" value="Rezky"
disabled><br>
```

First name:

Rezky

placeholder attribute

attribute placeholder pada inputan digunakan untuk membuat hint/petunjuk sebelum inputan tersebut diberi nilai attribute placeholder hanya bisa di gunakan pada inputan dengan attribute type (text, search, url, tel, email, and password)

```
<label for="fname">First name:</label><br>
<input type="text" id="fname" name="fname"
placeholder="masukan nama depan"><br>
```

First name:

masukan nama depan

required attribute

attribute required di gunakan untuk membuat pesan alert pada inputan jika inputan tersebut kosong pada saat input submit di jalankan pada form (validasi required)

```
<form action="form.html">
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="halo" required><br><br>
    <input type="submit" value="send">
</form>
```

First name:



Please fill out this field.

Element TextArea

Elemen text area <textarea> menggunakan penulisan dengan penutup tag untuk mendefinisikan kontrol input teks multi-baris.

Elemen <textarea> sering digunakan dalam membuat inputan yang lebih luas seperti komentar, ulasan, Address, dll.

Area inputan teks dapat menampung jumlah karakter yang tidak terbatas,

Ukuran area teks ditentukan oleh attribute **cols** (jumlah kolom) dan **rows** (jumlah baris) (atau dengan CSS).

Atribut **name** diperlukan untuk mereferensikan data inputan setelah data dikirimkan (jika Anda menghilangkan atribut name, tidak ada data dari area teks yang akan dikirimkan). Atribut **id** diperlukan untuk mengasosiasikan textarea dengan label yang menggunakan attribute **for**.

```
<label for="address">Address</label> <br>
<textarea id="address" name="address" rows="10"
cols="50"></textarea>
```



Element Select

Element **<select>** menggunakan penulisan dengan penutup tag yang di gunakan untuk membuat sebuah drop-down list yang mana di dalam element **<select>** terdapat element **<option>** sebagai opsi opsi apa saja yang tedapat di dalam inputannya

Atribut **name** diperlukan untuk mereferensikan data inputan setelah data dikirimkan (jika Anda menghilangkan atribut name, tidak ada data dari area teks yang akan dikirimkan). tag **<option>** pada element select menggunakan penulisan dengan tag penutup didefinisikan sebagai opsi opsi apa saja yang ada pada drop-down list

Atribut **id** diperlukan untuk mengasosiasikan textarea dengan label yang menggunakan attribute **for**.

attribute **value** pada option berfungsi sebagai nilai pada tag **<option>**

```
<label for="cars">Mobil</label> <br>

<select name="cars" id="cars">
    <option value="1">Volvo</option>
    <option value="2">Saab</option>
    <option value="3">Opel</option>
    <option value="4">Audi</option>
</select>
```

Mobil



FORM

Form Element

element form pada HTML menggunakan penulisan dengan tag penutup dimana berfungsi sebagai inputan pengguna/user

```
<form>
    .
    form elements
    .
</form>
```

element form adalah sebagai wadah berbagai jenis element inputan seperti inputan text, password, radio button, checkbox, dropdown-list, textarea, input submit

```

<form action="form.html">
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
</form>

```

First name:
John
Last name:
Doe
Submit

Form Attribute

Attribute Action

Atribut action menentukan tindakan yang akan dilakukan saat form di submit.

Biasanya, data formulir dikirim ke file di server (server-side) saat pengguna mengklik tombol **submit**.

Pada contoh di bawah ini, data form dikirim ke file bernama "tambah.php". File ini berisi skripsi server(server site) yang menangani data form:

```

<form action="tambah.php">
    <label for="fname">Nama Depan</label><br>
    <input type="text" id="fname" name="fname" ><br>
    <label for="lname">Nama Belakang</label><br>
    <input type="text" id="lname" name="lname" ><br><br>
    <input type="submit" value="Submit">
</form>

```

Attribute method

Atribut method menentukan metode HTTP yang akan digunakan saat mengirimkan form data.

Form data dapat dikirim sebagai variabel URL (dengan method="get") atau sebagai post (dengan method="post").

Method default jika method tidak diisi kedalam inputannya adalah method GET
method get

```
<form action="/action_page.php" method="get">
```

method post

```
<form action="/action_page.php" method="post">
```

perbedaan method post dan get

method get

- Menambahkan data form ke URL, dalam pasangan name/value

```
https://sanbercode.com/bootcamp?kelas="laravel"
```

- JANGAN PERNAH menggunakan method GET untuk mengirim data sensitif! seperti password, email, dll (data form yang dikirimkan terlihat di URL!)
- Panjang URL terbatas (2048 karakter)
- Berguna untuk pengiriman formulir di mana user dapat menfilter data berdasarkan name dan value pada url
- Berguna untuk pengiriman formulir di mana pengguna ingin mem-bookmark hasilnya

mehod post

- mengirim data form ke sisi server agar dapat diolah seperti form login, register dll (data form yang dikirimkan tidak ditampilkan di URL)
- POST tidak memiliki batasan ukuran, dan dapat digunakan untuk mengirim data dalam jumlah besar.
- Pengiriman formulir dengan POST tidak dapat di-bookmark

Bab 2 CSS

Sejarah CSS

CSS pertama kali dikembangkan pada tahun 1996 oleh Håkon Wium Lie(gambar kiri) dan Bert Bos(gambar kanan), sebagai bagian dari proyek World Wide Web Consortium (W3C) untuk membuat standar web. Versi pertama CSS, CSS1, diterbitkan pada tahun 1996, yang mendefinisikan tata letak, warna, dan gaya teks. Kemudian pada tahun 1998, CSS2 diterbitkan dengan menambahkan kemampuan untuk mengontrol tampilan elemen seperti tabulasi, citasi, dan penomoran.

Pada tahun 1999, W3C meluncurkan CSS level 2 revision 1 atau CSS2.1, yang merupakan perbaikan dari CSS2 dengan menambahkan beberapa fitur baru dan menghilangkan beberapa yang dianggap tidak diperlukan. CSS2.1 adalah versi CSS yang digunakan secara luas oleh pengembang web dan browser modern hingga saat ini.

Pada tahun 2001, W3C mulai mengembangkan CSS3, yang merupakan versi CSS terbaru yang berisi sejumlah fitur baru seperti transformasi, animasi, dan gradien. CSS3 masih dalam tahap pengembangan dan dikelompokkan ke dalam beberapa modul yang terpisah.

jadi apa itu CSS?

jika html adalah pondasi dari sebuah website CSS adalah bahasa untuk memperindah tampilan dari website seperti mengatur posisi element, mengubah warna dll

Selektor , Property dan Value

Pada CSS ada beberapa istilah biasa kita temukan diantara : selector , properti, dan value . Kita mempelajari terlebih dahulu pengertian dari ketiga istilah tersebut.

Selektor adalah metode pengelompokan syntax didalam CSS. Selector juga dapat diartikan sebagai aturan yang digunakan untuk memisahkan beberapa syntax dengan target yang berbeda.

Properti adalah suatu jenis style CSS yang berfungsi memberikan style pada selector yang telah ditentukan. Property pada CSS sangat banyak sekali. Banyak contoh properti css seperti : background-color, color, font-family, font-size, margin, padding dan lain lain.

Value adalah nilai dari property . Dan value harus bersesuaian dengan properti, misalnya properti "font-family" tidak bisa diisi "red" karena "red" adalah value untuk property "color".

Ada bentuk baku penulisan selektor , properti dan value :



```
h2 {  
    font-family: sans;  
    color: rgb(10, 8, 8);  
}
```

selektor pada contoh diatas adalah h2

properti pada contoh diatas adalah font-family dan color

value pada contoh diatas adalah sans dan rgb(10, 8, 8);

Cara Penulisan Kode CSS dalam HTML

Internal CSS

Penulisan CSS didalam tag `<head>` atau `<body>` HTML. Penulisannya ditulis dalam tag `<style>`.

```
<!DOCTYPE html>
<html>
<head>
    <style type="text/css">
        p{
            font-family: Helvetica;
            line-height: 2.00 em;
            font-size: 20px;
            color: orange;
        }
    </style>
</head>

<body>

    <style type="text/css">
        h2 {
            font-family: sans;
            color: rgb(10, 8, 8);
        }
    </style>

    <h2>Hello CSS</h2>
    <p>Hello aku bisa CSS</p>
</body>
</html>
```

Inline CSS

Penulisan CSS menggunakan **atribut style** yang di masukan langsung ke element HTML.
Contoh cara pemakaian inline CSS:

```
<!DOCTYPE html>
<html>
<head>
</head>

<body>
  <h2 style="color: blue;">Hello CSS</h2>
  <p style="color: red;">Hello aku bisa CSS</p>
</body>
</html>
```

Eksternal CSS

Penulisan CSS terpisah dengan file HTML. File CSS ditulis dan disimpan dengan file berekstensi .css .Untuk memasukkan ekstensi CSS ke HTML menggunakan tag `<link href="namefile.css">`. `href` diisi direktori folder dan nama file CSS untuk mengarahkan halaman html menemukan file CSS tersebut.

index.html

```
<!DOCTYPE html>
<html>
<head>
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <h2>Hello CSS</h2>
  <p>Hello aku bisa CSS</p>
```

```
</body>  
</html>
```

style.css

```
h2 {  
    color="blue";  
}  
  
p {  
    color="red";  
}
```

Selector Di CSS

seperti yang di jelaskan di atas selector adalah metode untuk mengambil target yang akan kita styling seperti:

Simple selectors (select elements berdasarkan **nama element** , **attribute id**, **attribute class**)

contoh:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    color: red;  
}  
  
#text{  
    color: green;  
}
```

```

.text-paragraf{
    color: blue;
}

</style>
</head>
<body>

<p>pagragraf dengan selector element</p>
<p id="text">paragraf dengan selector id</p>
<p class="text-paragraf">paragraf dengan selector class</p>

</body>
</html>

```

code diatas menjelaskan bahwa selector menggunakan attribute id harus diawali dengan simbol # kemudian isi/value dari id nya kemudian selector menggunakan attribut class diawali dengan simbol titik(.) kemudian isi/value dari id nya kemudian select menggunakan element langsung menggunakan nama dari elementnya

CSS Border

border css adalah properti css untuk mengatur (ketebalan, warna, dan jenis) garis pembatas pada element yang di selector

CSS Border Style

properti border-style adalah cara kita untuk menetukan jenis garis yang akan di buat

value garis properti border style:

dotted - untuk membuat garis titik titik

dashed - untuk membuat garis putus putus

solid - untuk membuat garis lurus

double - untuk membuat garis double

groove - untuk membuat garis 3D. Efeknya tergantung pada nilai border-color

ridge - untuk membuat garis bergerigi 3D. Efeknya tergantung pada nilai border-color

inset - untuk membuat garis 3D kedalam. Efeknya tergantung pada nilai border-color

outset - untuk membuat garis 3D keluar. Efeknya tergantung pada nilai border-color
none - untuk membuat tanpa garis

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
</style>
</head>
<body>

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
```

```
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>

</body>
</html>
```

CSS Border Width

properti border-width berfungsi untuk mengatur ketebalan dari garis pembatas dari keempat sisi (kiri kanan atas bawah).

ketebalan dapat di beri nilai berupa (px, pt, cm, em, etc) atau mendefinisikan dengan values: thin, medium, atau thick:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    border-style: solid;
    border-width: 5px;
}

p.two {
    border-style: solid;
    border-width: medium;
}

p.three {
    border-style: solid;
```

```

border-width: 0.2cm;
}

</style>
</head>
<body>
<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>

</body>
</html>

```

Some text.

Some text.

Some text.

CSS Border Color

properti css border color dibuat untuk memberi warna pada garis dari keempat sisi (kiri kanan atas bawah).

The color can be set by:

name - menggunakan nama warna pada value-nilai (harus menggunakan bahasa inggris), seperti "red"

HEX - menggunakan kode HEX pada value-nilai, seperti "#ff0000"

RGB - menggunakan kode RGB pada value-nilai , seperti "rgb(255,0,0)"

```

<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    border-style: solid;
    border-color: red;
}

```

```
}

</style>

</head>

<body>

<p class="one">A solid red border</p>

</body>

</html>
```

A solid red border

CSS border side

border side Di CSS, yaitu cara kita untuk mengatur garis di setiap sisinya (atas, kanan, bawah, dan kiri) baik itu border-style, border-color, maupun border width

```
<!DOCTYPE html>

<html>
<head>
<style>

p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
}

</style>
</head>
<body>

<p>Border Side</p>

</body>
</html>
```

Border Side

Border Shorthand

border shorthand adalah properti penulisan singkat dari bordernya yang mengabungkan border-style, border-color, dan border-width menggunakan properti bernama **border**

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    border: 5px solid red;
}
</style>
</head>
<body>
<p>border</p>
</body>
</html>
```

border

CSS Margin

Margin adalah properti CSS yang digunakan untuk menciptakan jarak di sekitar elemen yang di selector , di luar border.

Margin - Individual Sides

properti css untuk unutk memberi jarak di setiap sisinya :

margin-top

margin-right

margin-bottom

margin-left

margin properti diatas dapat diisi dengan value:

auto - browser akan menghitung jarak margin

length - menentukan margin dengan value px, pt, cm, etc.

% - menentukan margin dalam % dari lebar elemen yang bergantung dari lebar device nya juga

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
    background-color: lightblue;
}
</style>
</head>
<body>

<h2>margin example</h2>

<div>element yang diberi margin</div>

</body>
</html>
```

margin example

element yang diberi margin

shorthand margin

penulisan margin dengan satu properti yaitu margin yang mengabungkan margin-top, margin-left, margin-right, margin-bottom

dengan 4 nilai/value

```
p {  
    margin: 25px 50px 75px 100px;  
}
```

margin atas **25px**

margin kanan **50px**

margin bawah **75px**

margin kiri **100px**

dengan 3 nilai/value

```
p {  
    margin: 25px 50px 75px;  
}
```

margin atas **25px**

margin kiri dan kanan **50px**

margin bawah **75px**

dengan 2 nilai/value

```
p {
```

```
margin: 25px 50px;  
}
```

margin atas dan bawah **25px**

margin kiri dan kanan **50px**

dengan 1 nilai/value

```
p {  
    margin: 25px;  
}
```

margin ke empat sisi(kiri, kanan, atas bawah) **25px**

CSS Padding

Padding adalah properti CSS yang digunakan untuk membuat jarak di sekitar content elemen, di dalam border.

Padding- Individual Sides

property css untuk memberi jarak di setiap sisinya :

- padding-top
- padding-right
- padding-bottom
- padding-left

padding property diatas dapat diisi dengan value:

auto - browser akan menghitung jarak padding

length - menentukan padding dengan value px, pt, cm, etc.

% - menentukan padding dalam % dari lebar elemen yang bergantung dari lebar device nya juga

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    background-color: lightblue;
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
</style>
</head>
<body>

<h2>CSS Padding</h2>

<div>Contoh CSS Padding</div>

</body>
</html>
```

CSS Padding

Contoh CSS Padding

shorthand padding

penulisan padding dengan satu property yaitu padding yang mengabungkan padding-top, padding-left, padding-right, padding-bottom

dengan 4 nilai/value

```
p {  
    padding: 25px 50px 75px 100px;  
}
```

padding atas **25px**

padding kanan **50px**

padding bawah **75px**

padding kiri **100px**

dengan 3 nilai/value

```
p {  
    padding: 25px 50px 75px;  
}
```

padding atas **25px**

padding kiri dan kanan **50px**

padding bawah **75px**

dengan 2 nilai/value

```
p {  
    padding: 25px 50px;  
}
```

padding atas dan bawah **25px**

padding kiri dan kanan **50px**

dengan 1 nilai/value

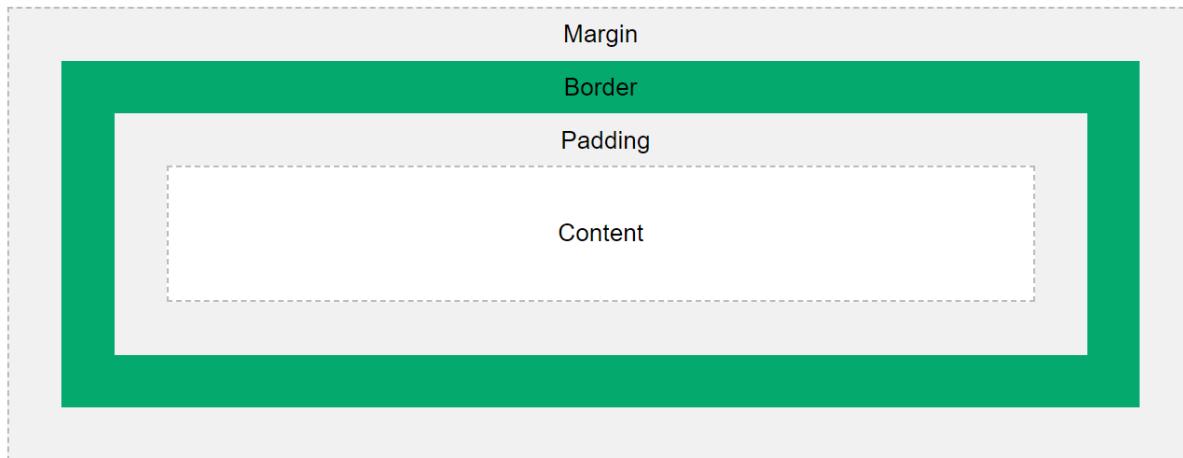
```
p {
```

```
padding: 25px;  
}
```

padding ke empat sisi(kiri, kanan, atas bawah) **25px**

Box Model

Box Model CSS pada dasarnya adalah sebuah kotak yang membungkus setiap elemen HTML. Ini terdiri dari: margin, border, padding, dan Content didalamnya



Content - element yang di selector

Padding - batas jarak content ke border

border - berfungsi memberi garis pada content

Margin - berfungsi memberi jarak di luar border

CSS Position Property

position property adalah cara kita menentukan posisi dari element nya
ada 5 macam value dari property position diantaranya:

- static
- relative
- fixed
- absolute
- sticky

position static

position static adalah position default dari element nya di mana kita tidak bisa mengatur
property top, right, bottom, left

```
<!DOCTYPE html>
<html>
<head>
<style>
.static {
    position: static;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>

<div class="static">
ini position static
</div>

</body>
</html>
```

ini position static

position relative

position relative sama seperti position static bedanya position relative kita bisa mengatur
properti top, right, bottom, left

```
<!DOCTYPE html>
<html>
<head>
<style>
p.relative {
    position: relative;
    left: 10px;
    top: 40px;
    border: 3px solid #73AD21;
}
.box{
    margin:10px;
}
</style>
</head>
<body>

<h2>position: relative;</h2>

<div class="box">

<p class="relative">
ini position relative
</p>

</div>
```

pada kode diatas ada jarak sejauh 40px; dibagian atas dan 10px di bagian kiri

position: relative;

ini position relative

position absolute

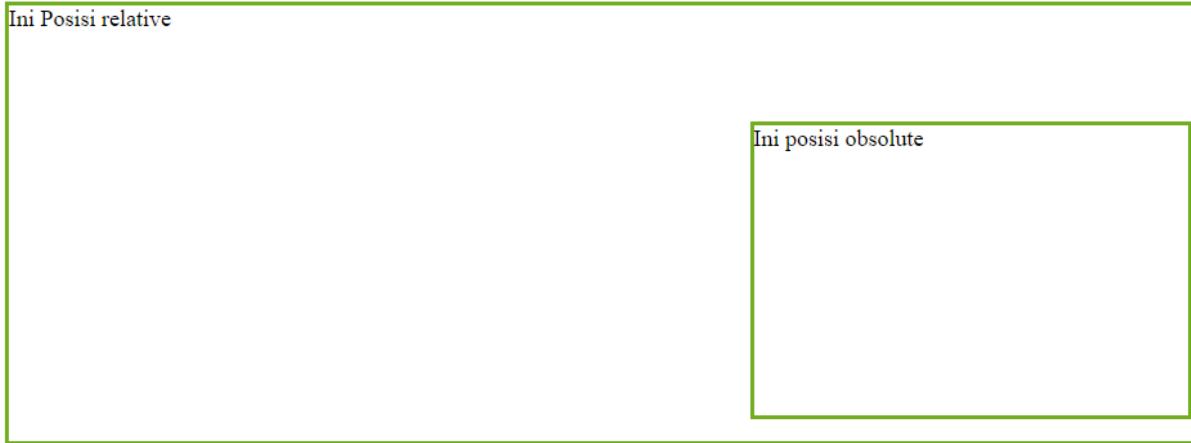
position absolute adalah posisi dimana dia menupuk ke posisi element lainnya

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
    position: relative;
    height: 300px;
    border: 3px solid #73AD21;
}

div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 300px;
    height: 200px;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>
<div class="relative">Ini Posisi relative
    <div class="absolute">Ini posisi absolute</div>
</div>

</body>
</html>
```

dari contoh diatas position element dengan class absolute nemunpuk denagn element dengan class relative



position fixed

position fixed adalah posisi dimana element tetap berada di posisi yang sama bahkan jika halaman dari web tersebut kita scroll kebawah dengan mengatur properti top, left, right, bottom pada element nya

```
<!DOCTYPE html>
<html>
<head>
<style>
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
}

body{
    background-color: lightgrey;
}
</style>
</head>
<body>
```

```
<div class="fixed">  
ini position fixed  
</div>  
  
</body>  
</html>
```



dari gambar di atas position dari fixednya berada di kiri bawah bedasarkan properti dari bottom dan right

contoh lainnya

A screenshot of a website's user interface. On the left, there is a sidebar with various navigation items: "Dasbor", "Overview", "Kelas Saya" (which is highlighted in blue), "Daftar Kelas", "Job Listing", "Afiliasi", and "Kupon". The main content area shows two course cards. The first card is for "Wordpress – CMS Web (Batch 38)" and the second for "Python Backend Development(Batch 36)". Both cards have a "Detail Kelas" button. The top right corner of the screen shows a user profile with the email "yogesworo@gmail.com" and the text "Meet Course 6". The overall layout demonstrates how a fixed header or sidebar remains in place while other content scrolls.

bagian header dari halaman website diatas masih berada di posisi yang sama bahkan saat halaman nya di scroll kebawah

position stinky

position stinky adalah gabungan antar position relatif dan fixed dimana pada saat position sudah offset maka posisi dari element nya akan tetap berada di posisi yang sudah ditentukan

```
<!DOCTYPE html>
<html>
<head>
<style>
div.sticky {
    position: -webkit-sticky;
    position: sticky;
    top: 0;
    padding: 5px;
    background-color: #cae8ca;
    border: 2px solid #4CAF50;
}
</style>
</head>
<body>

<p>coba di <b>scroll</b></p>

<div class="sticky">posisi stinky</div>

<div style="padding-bottom:2000px">
</div>

</body>
</html>
```

coba di scroll

posisi stinky



`position: -webkit-sticky;` di buat agar fungsi position sticky dapat berjalan di web browser safari

FlexBox

flexbox atau flexible box merupakan konsep layout di css yang di gunakan untuk mengatur element beserta item-item didalamnya

persiapan :

```
<!DOCTYPE html>
<html>
<head>
<style>
header{
    display:flex;
    background-color:lightgrey;
}
div{
    background-color:lightblue;
    margin: 10px;
    padding: 20px;
}
</style>
</head>
<body>

<header>
    <div>Item 1</div>
        <div>Item 2</div>
    <div>Item 3</div>
</header>

</body>
</html>
```

dari code diatas kita menambah property display flex kedalam element headernya disini kita mengatur element header nya menggunakan property flex-direction, justify-content

property flex-direction

flex-direction:row;

value row pada flex directionnya yaitu default dari property flex-direction dimana item berurut dari kiri ke kanan

```
header{  
    display:flex;  
    flex-direction:row;  
    background-color:lightgrey;  
}
```



flex-direction:row-reverse;

value row-reverse dari flex directionnya yaitu kebalikan dari value row dimana item berurut dari kanan ke kiri

```
header{  
    display:flex;  
    flex-direction:row-reverse;  
    background-color:lightgrey;  
}
```

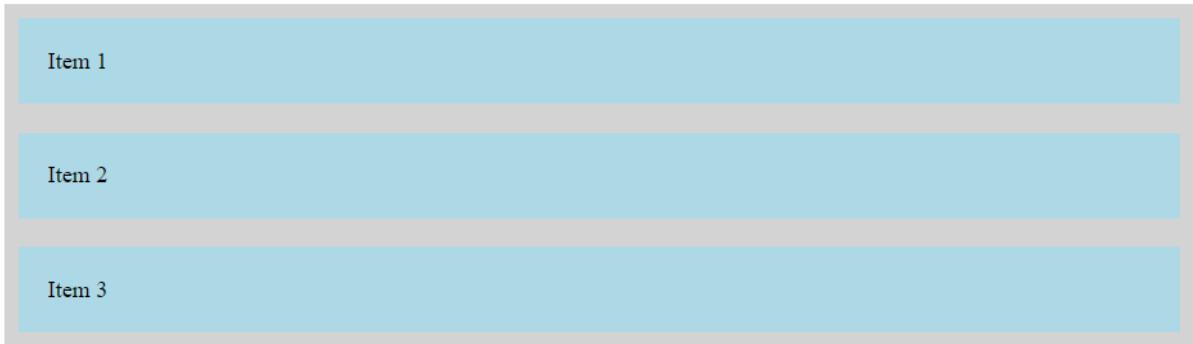


flex-direction:column;

dari tampilan dibawah kita menggunakan value column dari flex directionnya yaitu mengurutkan item pada headernya dari atas ke bawah

```
header{  
    display:flex;
```

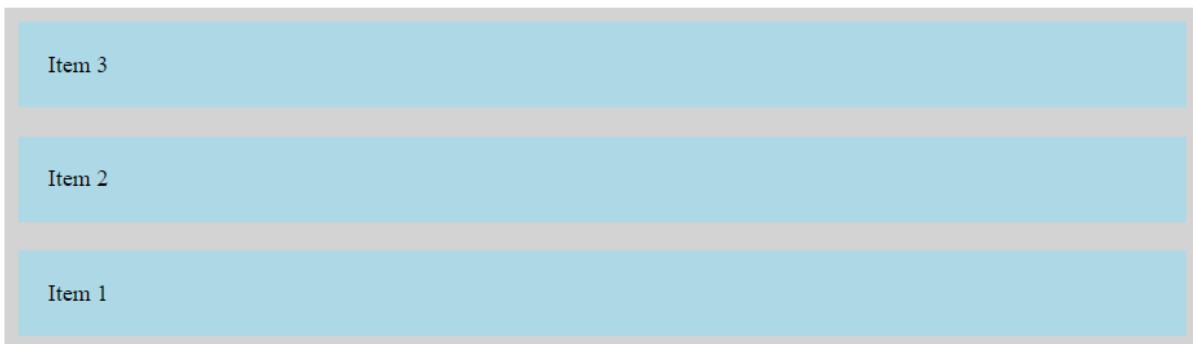
```
flex-direction:column;  
background-color:lightgrey;  
}
```



flex-direction:column-reverse;

value column-reverse adalah kebalikan dari column dimana mengurutkan item dari bawah ke atas

```
header{  
display:flex;  
flex-direction:column-reverse;  
background-color:lightgrey;  
}
```



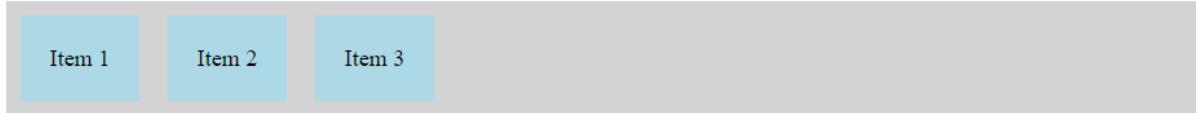
property justify-content

justify-content yaitu cara kita untuk mengatur posisi dan jarak dari item yang ada di dalam elementnya

justify-content:flex-start;

value flex-start yaitu cara kita untuk mengatur semua itemnya berada di ujung kiri yang mana adalah position defaultnya

```
header{  
    display:flex;  
    justify-content:flex-start;  
    background-color:lightgrey;  
}
```



justify-content:flex-end;

value flex-end yaitu cara kita untuk mengatur semua itemnya berada di ujung kanan

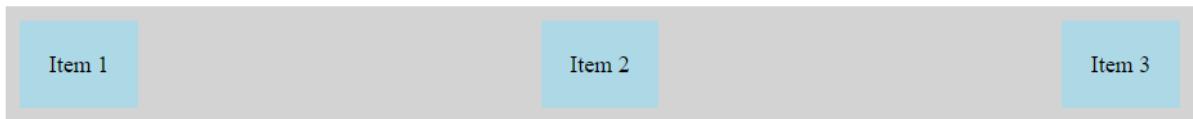
```
header{  
    display:flex;  
    justify-content:flex-end;  
    background-color:lightgrey;  
}
```



justify-content:space-between;

value space-between yaitu cara kita untuk mengatur jarak itemnya dimana item pertama akan berada di ujung kiri(flex-start) dan item terakhir akan berada di ujung kanan (flex-end)

```
header{  
display:flex;  
justify-content:space-between;  
background-color:lightgrey;  
}
```



justify-content:space-around;

value space-around yaitu jarak di setiap itemnya akan dibagi secara merata

```
header{  
display:flex;  
justify-content:space-around;  
background-color:lightgrey;  
}
```



Bab 3 Bootstrap

sejarah Bootstrap

Bootstrap awalnya dibuat oleh Mark Otto dan Jacob Thornton sebagai alat bantu untuk mempercepat pengembangan tampilan halaman web di Twitter.

Pada awalnya, Bootstrap dikenal sebagai Twitter Blueprint, namun kemudian diubah menjadi Bootstrap dan dirilis ke publik sebagai proyek open-source pada tahun 2011. Bootstrap memungkinkan para pengembang web untuk dengan mudah membuat tampilan halaman web yang responsif dan konsisten di berbagai perangkat, dari desktop hingga perangkat seluler.

Bootstrap menjadi sangat populer di kalangan pengembang web karena mudah digunakan dan memiliki dokumentasi yang lengkap. Selain itu, Bootstrap juga terus diperbarui dan dikembangkan oleh komunitas pengembang web yang terus bertumbuh.

Pada saat ini, Bootstrap telah mencapai versi 5.0 yang dirilis pada tahun 2021. Bootstrap 5.0 membawa banyak perubahan dan peningkatan, termasuk dukungan untuk CSS Grid dan mode gelap (dark mode), serta memperbaiki performa dan penggunaan kelas CSS yang lebih fleksibel.

Bootstrap digunakan oleh banyak situs web terkenal seperti Airbnb, Spotify, dan Microsoft, dan terus menjadi salah satu kerangka kerja paling populer untuk pengembangan tampilan halaman web.

Apa itu Bootstrap

Bootstrap adalah kerangka kerja (framework) front-end yang terdiri dari HTML, CSS, dan JavaScript. Bootstrap digunakan untuk mempercepat proses pengembangan tampilan halaman web yang responsif dan konsisten di berbagai perangkat, dari desktop hingga perangkat seluler.

Bootstrap menyediakan berbagai macam komponen, seperti bottom, form, tabel, navigation, dan banyak lagi. Komponen-komponen ini dapat digunakan dan disesuaikan dengan mudah untuk mempercepat proses pengembangan. Selain itu, Bootstrap juga menyediakan sistem grid yang fleksibel, yang memungkinkan para pengembang untuk merancang tata letak halaman web yang responsif dan sesuai dengan kebutuhan.

Bootstrap juga memiliki dokumentasi yang lengkap dan mudah dipahami, serta mendukung berbagai browser modern. Bootstrap sangat populer di kalangan pengembang web karena mudah digunakan, memiliki banyak fitur, dan terus diperbarui oleh komunitas pengembang yang aktif.

Dalam pengembangan aplikasi web, Bootstrap dapat membantu pengembang untuk menghemat waktu dan tenaga dalam membuat tampilan halaman web yang responsif dan konsisten. Oleh karena itu, Bootstrap menjadi salah satu kerangka kerja front-end yang paling populer di dunia.

Memasang bootstrap

Dalam dokumentasi ini akan digunakan Bootstrap 4 sebagai contoh kasus. Bootstrap 4 memiliki halaman dokumentasi tersendiri yang dipakai sebagai pedoman pemakaian di link berikut.

<https://getbootstrap.com/docs/4.6/getting-started/introduction/>

Terdapat beberapa cara untuk memasangkan Bootstrap pada halaman web atau html:

- memasangkan file CDN
- mendownload source file Bootstrap

Memasang File CDN

File CDN yaitu segala file yang dibutuhkan untuk menggunakan Bootstrap yang bersumber dari suatu pranala atau link tertentu. Keuntungannya adalah cara ini merupakan cara termudah untuk memasangkan Bootstrap pada project web kita dan tidak usah

mendownload apapun. Kekurangannya adalah dalam pengembangan project web kita harus selalu terhubung dengan internet karena file CDN tersebut diperoleh dari website luar.

Di halaman dokumentasi bootstrap kita dapat copy bagian CSS:

<https://getbootstrap.com/docs/4.6/getting-started/introduction/#starter-template>

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link
      rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.
2/dist/css/bootstrap.min.css"
      integrity="sha384-xOolHFLeh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0
nLMwNLD69Npy4HI+N" crossorigin="anonymous">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: jQuery and Bootstrap Bundle (includes Popper)
-->
    <script
      src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.m
in.js"
      integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+I
bbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0J
z8D6OP9dO5Vg3Q9ct" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS --&gt;
&lt;!--
&lt;script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-Dfxdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+I
bbVYUew+OrCXaRkfj" crossorigin="anonymous"&gt;&lt;/script&gt;
&lt;script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoY1FhbGU+6
BZp6G7niu735Sk71N" crossorigin="anonymous"&gt;&lt;/script&gt;
&lt;script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.min.js"
integrity="sha384-sLIOodYLS7CIRQpBjl+C7nPvqq+FbNUBDunl/OZv93DB7L
n/533i8e/mZXLi/P+" crossorigin="anonymous"&gt;&lt;/script&gt;
--&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>

```

Komponen Bootstrap

Ada beberapa komponen yang sering digunakan di Bootstrap, di antaranya:

- Grid system: Sistem grid Bootstrap memungkinkan Anda untuk membagi halaman web ke dalam kolom dan baris, yang memungkinkan Anda untuk membuat tata letak responsif dan fleksibel.
- Typography: Bootstrap memiliki aturan gaya yang telah ditentukan sebelumnya untuk berbagai jenis teks seperti judul, paragraf, dan link.
- Navbar: Navbar Bootstrap adalah komponen navigasi yang dapat disesuaikan yang dapat ditempatkan di bagian atas atau bawah halaman web Anda.
- Forms: Bootstrap menyediakan komponen untuk membuat form seperti input, textarea, select, dan lainnya.
- Buttons: Bootstrap menyediakan banyak jenis tombol yang dapat digunakan dalam website, seperti tombol default, tombol outline, tombol berwarna, tombol block, dan lainnya.
- Card: Komponen card Bootstrap digunakan untuk menampilkan informasi seperti artikel, produk, atau profil pengguna.
- Carousel: Komponen carousel Bootstrap digunakan untuk membuat slideshow gambar atau konten lainnya.
- Alerts: Komponen alert Bootstrap digunakan untuk memberi tahu pengguna tentang informasi penting atau kesalahan di situs web.

System Grid Bootstrap

Grid di Bootstrap adalah sistem layout grid yang terdiri dari 12 kolom yang dapat digunakan untuk membagi dan mengatur tata letak halaman web. Grid ini memungkinkan pengembang untuk membuat tata letak responsif dan fleksibel dengan mudah, sehingga halaman web dapat diakses dan dilihat dengan baik pada berbagai perangkat dan ukuran layar.

Dalam grid Bootstrap, sebuah halaman dibagi menjadi 12 kolom yang sama lebar, dengan setiap kolom diberi nomor indeks dari 1 hingga 12. Pengembang dapat menentukan jumlah kolom yang ingin digunakan untuk setiap elemen pada halaman web dengan memberikan kelas CSS yang sesuai.

Contohnya, untuk membuat 3 kolom yang sama lebar di halaman web, pengembang dapat menggunakan class "col-6" untuk masing-masing elemen, yang akan membagi halaman menjadi dua kolom yang sama lebar. Pengembang juga dapat menambahkan class tambahan seperti "col-md-6" untuk menentukan jumlah kolom yang akan digunakan pada perangkat dengan lebar layar yang lebih besar.

	xs ≤576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container <small>max-width</small>	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

Berikut adalah contoh kode HTML sederhana yang menggunakan grid Bootstrap untuk membuat 3 kolom yang sama lebar:

```
.....  
<div class="row">  
  <div class="col-4">  
    <p>Ini adalah kolom 1</p>
```

```
</div>
<div class="col-4">
    <p>Ini adalah kolom 2</p>
</div>
<div class="col-4">
    <p>Ini adalah kolom 3</p>
</div>
</div>
....
```

output



Ini adalah kolom 1 Ini adalah kolom 2 Ini adalah kolom 3

Dalam kode di atas, kelas "row" digunakan untuk membuat sebuah baris yang akan berisi tiga kolom. Kemudian, kelas "col-4" digunakan untuk membuat masing-masing kolom dengan lebar yang sama.

disini kita bisa mangatur bisa mengatur lebar dari kolom nya asal **pembagian nilai kolom tidak boleh lebih dari 12**

Berikut adalah contoh kode menggunakan grid Bootstrap untuk membuat 3 kolom yang mana kolom 2 lebih lebar dibangi kolom 1 dan 3 sama lebar:

```
<div class="row">
  <div class="col-2">
    <p>Ini adalah kolom 1</p>
  </div>
  <div class="col-8">
    <p>Ini adalah kolom 2</p>
  </div>
  <div class="col-2">
    <p>Ini adalah kolom 3</p>
  </div>
</div>
```

Ini adalah kolom 1 Ini adalah kolom 2 Ini adalah kolom 3

di dalam grid nya kita juga dapat masukan component apa saja di dalam setiap kolomnya contoh:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-xOolHFLeh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0
nLMwNLD9Npy4HI+N" crossorigin="anonymous">
    <title>Hello, world!</title>
  </head>
  <body>
<div class="row">
  <div class="col-2">
    <ul class="list-group">
      <li class="list-group-item">Phyton</li>
      <li class="list-group-item">React JS</li>
      <li class="list-group-item">Laravel</li>
      <li class="list-group-item">Digital Marketing</li>
      <li class="list-group-item">Desain Grafis</li>
      <li class="list-group-item">UI/UX</li>
      <li class="list-group-item">Desain Grafis</li>
      <li class="list-group-item">VueJS</li>
      <li class="list-group-item">NodeJS</li>
      <li class="list-group-item">Golang</li>
    </ul>
  </div>
  <div class="col-8">
```

```

<div class="jumbotron">
    <h1 class="display-4">Hello, world!</h1>
    <p class="lead">Selamat Datang Disanber code tempat
kalian belajar pemrograman</p>
</div>
</div>
<div class="col-2">
    <ul class="list-group">
        <li class="list-group-item">SanberCode</li>
        <li class="list-group-item">SanberHub</li>
        <li class="list-group-item">Sanbersy</li>
    </ul>
</div>
</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NTUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIxVXM" crossorigin="anonymous"></script>

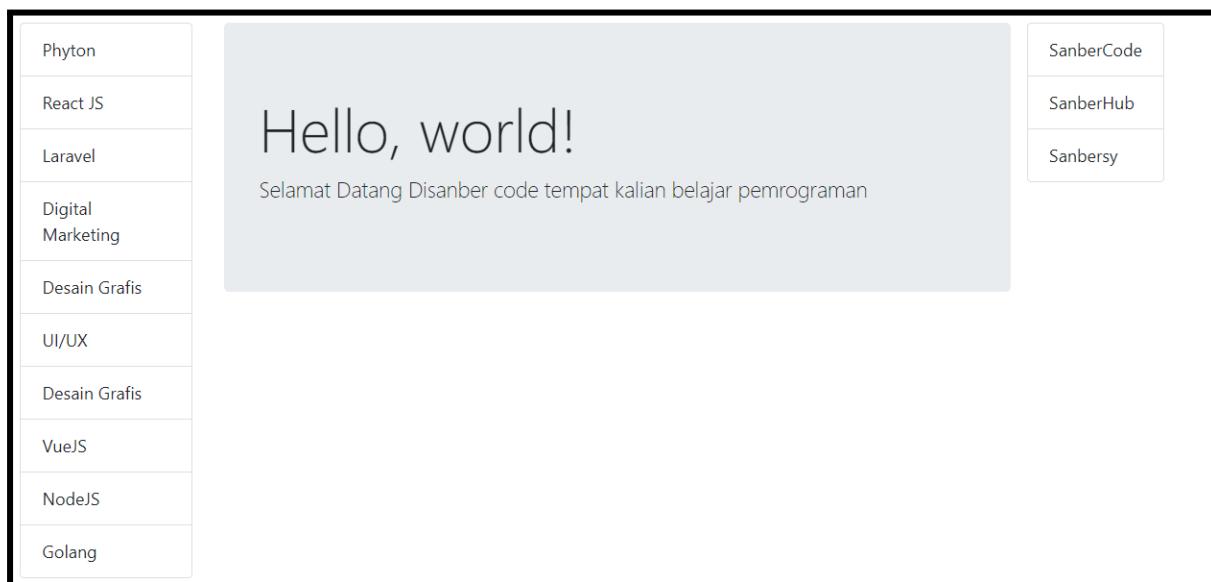
<!-- Option 2: Separate Popper and Bootstrap JS -->
<!--
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"
integrity="sha384-IQsoLX15PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
integrity="sha384-cVKIPhGWic2Al4u+LWgxKTRIcfu0JTxR+EQDz/bgldoEyl

```

```
4H0zUF0QKbrJ0EcQF" crossorigin="anonymous">></script>
-->
</body>
</html>
```

pada kode diatas kolom 1 dan kolom 3 didiisi dengan list group sedangkan kolom 2 diisi dengan jumbotron

output



Typografi

Bootstrap 4 menyediakan beberapa kelas untuk mengatur tipografi pada halaman web. Berikut adalah beberapa kelas Bootstrap 4 yang umum digunakan untuk mengatur tipografi:

1. h1 - h6: Kelas ini digunakan untuk membuat judul pada halaman web. Semakin tinggi angka yang digunakan, semakin kecil ukuran hurufnya. Contoh: <h1>Judul 1</h1>, <h2>Judul 2</h2>, dan seterusnya.
2. lead: Kelas ini digunakan untuk membuat teks utama atau penjelasan pada halaman web. Contoh: <p class="lead">Teks utama</p>.
3. display-1 - display-4: Kelas ini digunakan untuk membuat teks besar dan mencolok. Semakin tinggi angka yang digunakan, semakin besar ukuran hurufnya. Contoh: <h1>

class="display-1">Teks besar</h1>, <h2 class="display-2">Teks besar</h2>, dan seterusnya.

4. font-weight-bold: Kelas ini digunakan untuk membuat teks menjadi tebal. Contoh: <p class="font-weight-bold">Teks tebal</p>.
5. text-uppercase: Kelas ini digunakan untuk membuat teks menjadi huruf kapital. Contoh: <p class="text-uppercase">TEKS KAPITAL</p>.
6. text-lowercase: Kelas ini digunakan untuk membuat teks menjadi huruf kecil. Contoh: <p class="text-lowercase">teks kecil</p>.
7. text-muted: Kelas ini digunakan untuk membuat teks menjadi abu-abu atau pudar. Contoh: <p class="text-muted">Teks pudar</p>.
8. text-danger, text-success, text-warning, text-info: Kelas-kelas ini digunakan untuk memberikan warna pada teks. Contoh: <p class="text-danger">Teks merah</p>, <p class="text-success">Teks hijau</p>, <p class="text-warning">Teks kuning</p>, <p class="text-info">Teks biru</p>.

Dalam penggunaan kelas tipografi Bootstrap 4, sebaiknya kita menyesuaikan dengan kebutuhan desain dan konten pada halaman web yang dibuat.

NavBar

Navbar adalah salah satu komponen penting dalam Bootstrap 4 yang sering digunakan untuk membuat navigasi pada halaman web

<https://getbootstrap.com/docs/4.6/components/navs/>

<https://getbootstrap.com/docs/4.6/components/navbar/>

beberapa contoh navigasi Horizontal

```
<ul class="nav">
  <li class="nav-item">
    <a class="nav-link" href="#">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">About</a>
  </li>
  <li class="nav-item">
```

```
<a class="nav-link" href="#">Contact Us</a>
</li>
</ul>
```

Home About Contact Us

navigasi Horizontal rata tengah

```
<ul class="nav justify-content-center">
  <li class="nav-item">
    <a class="nav-link" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">About</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Contact Us</a>
  </li>
</ul>
```

Home About Contact Us

navigasi horizontal rata kanan

```
<ul class="nav justify-content-end">
  <li class="nav-item">
    <a class="nav-link" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">About</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Contact Us</a>
  </li>
</ul>
```

```
</li>  
</ul>
```

Home About Contact Us

navigasi vertical

```
<ul class="nav flex-column">  
    <li class="nav-item">  
        <a class="nav-link" href="#">Home</a>  
    </li>  
    <li class="nav-item">  
        <a class="nav-link" href="#">About</a>  
    </li>  
    <li class="nav-item">  
        <a class="nav-link" href="#">Contact Us</a>  
    </li>  
</ul>
```

Home
About
Contact Us

Form

Bootstrap 4 menyediakan beberapa kelas dan komponen untuk mempermudah pembuatan form pada halaman web. Berikut adalah langkah-langkah untuk membuat form di Bootstrap 4:

1. Siapkan struktur HTML untuk form. Form harus diletakkan di dalam tag `<form>` dan terdiri dari beberapa elemen seperti `<div>`, `<input>`, dan `<button>`. Contoh:

```

<form>

    <div class="form-group">
        <label for="exampleInputEmail1">Email address</label>
        <input type="email" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp">
            <small id="emailHelp" class="form-text text-muted">We'll
never share your email with anyone else.</small>
    </div>

    <div class="form-group">
        <label for="exampleInputPassword1">Password</label>
        <input type="password" class="form-control"
id="exampleInputPassword1">
    </div>

    <div class="form-group form-check">
        <input type="checkbox" class="form-check-input"
id="exampleCheck1">
            <label class="form-check-label" for="exampleCheck1">Check me
out</label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
</form>

```

2. Tambahkan kelas form-group pada setiap elemen form untuk mengelompokkan elemen yang memiliki label. Anda juga dapat menambahkan atribut for pada tag <label> dan mengatur nilai sesuai dengan atribut id pada tag <input> untuk menghubungkan label dengan elemen form. Contoh:

```

<div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp">
        <small id="emailHelp" class="form-text text-muted">We'll
never share your email with anyone else.</small>
</div>

```

3. Gunakan kelas form-control pada tag <input> untuk membuat tampilan elemen form menjadi lebih konsisten dan responsif. Anda juga dapat menambahkan atribut placeholder pada tag <input> untuk memberikan petunjuk pada pengguna. Contoh:

```
<input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
```

4. Tambahkan kelas form-check pada tag <div> yang mengelompokkan elemen checkbox atau radio. Anda juga dapat menambahkan kelas form-check-input pada tag <input> dan kelas form-check-label pada tag <label> untuk membuat tampilan checkbox atau radio menjadi lebih konsisten dan responsif. Contoh:

```
<div class="form-group form-check">
  <input type="checkbox" class="form-check-input" id="exampleCheck1">
  <label class="form-check-label" for="exampleCheck1">Check me out</label>
</div>
```

5. Gunakan kelas btn dan kelas btn-primary pada tag <button> untuk membuat tampilan tombol form menjadi lebih konsisten dan responsif. Contoh:

```
<button type="submit" class="btn btn-primary">Submit</button>
```

The image shows a screenshot of a web form. At the top, there is a text input field with the placeholder "Email address". Below it is a larger text input field with the placeholder "We'll never share your email with anyone else.". Underneath these fields is a password input field with the placeholder "Password". To the left of the password field is a checkbox labeled "Check me out". At the bottom of the form is a blue rectangular button with the word "Submit" in white text.

Button

Bootstrap 4 menyediakan beberapa kelas dan komponen untuk membuat button yang responsif dan mudah disesuaikan. Berikut adalah beberapa kelas yang dapat digunakan untuk membuat button di Bootstrap 4:

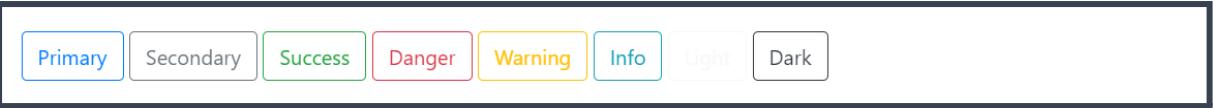
- btn: kelas dasar untuk membuat button. Contoh: `<button class="btn">Default</button>`
- btn-primary: kelas untuk membuat button dengan warna biru yang menonjol. Contoh: `<button class="btn btn-primary">Primary</button>`
- btn-secondary: kelas untuk membuat button dengan warna abu-abu yang lebih netral. Contoh: `<button class="btn btn-secondary">Secondary</button>`
- btn-success: kelas untuk membuat button dengan warna hijau yang menunjukkan aksi yang berhasil dilakukan. Contoh: `<button class="btn btn-success">Success</button>`
- btn-danger: kelas untuk membuat button dengan warna merah yang menunjukkan aksi yang gagal dilakukan. Contoh: `<button class="btn btn-danger">Danger</button>`
- btn-warning: kelas untuk membuat button dengan warna kuning yang menunjukkan peringatan. Contoh: `<button class="btn btn-warning">Warning</button>`
- btn-info: kelas untuk membuat button dengan warna biru muda yang menunjukkan informasi. Contoh: `<button class="btn btn-info">Info</button>`
- btn-light: kelas untuk membuat button dengan warna putih yang lebih netral. Contoh: `<button class="btn btn-light">Light</button>`
- btn-dark: kelas untuk membuat button dengan warna hitam yang lebih netral. Contoh: `<button class="btn btn-dark">Dark</button>`

Selain kelas-kelas dasar di atas, Bootstrap 4 juga menyediakan kelas-kelas lain seperti btn-outline-primary (untuk membuat button dengan border biru dan teks putih) dan btn-block (untuk membuat button memenuhi lebar elemen yang menyimpannya).

Contoh penggunaan:

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn
btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>
```

```
<button type="button" class="btn
btn-outline-primary">Primary</button>
<button type="button" class="btn
btn-outline-secondary">Secondary</button>
<button type="button" class="btn
btn-outline-success">Success</button>
<button type="button" class="btn
btn-outline-danger">Danger</button>
<button type="button" class="btn
btn-outline-warning">Warning</button>
<button type="button" class="btn
btn-outline-info">Info</button>
<button type="button" class="btn
btn-outline-light">Light</button>
<button type="button" class="btn
btn-outline-dark">Dark</button>
```

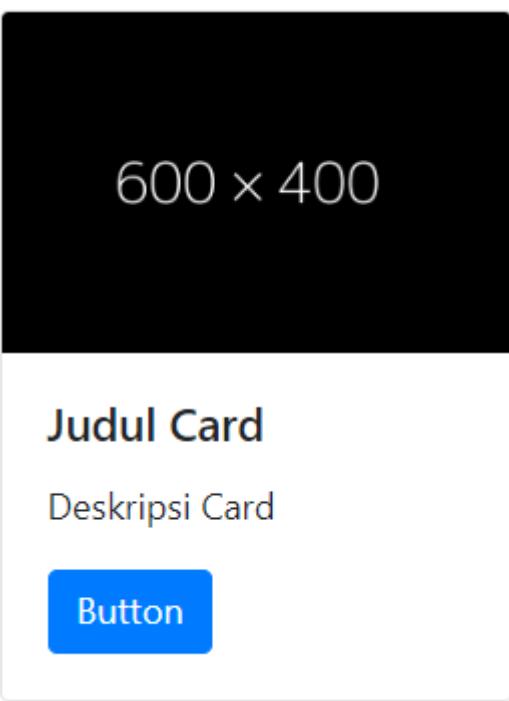


Card

Card di Bootstrap 4 adalah sebuah komponen yang berfungsi sebagai wadah atau container untuk menampilkan informasi, gambar, dan konten lainnya. Card ini bisa digunakan untuk membuat tampilan yang rapi dan modern dalam halaman web.

Berikut ini adalah struktur dasar dari card di Bootstrap 4:

```
<div class="card">
    
    <div class="card-body">
        <h5 class="card-title">Judul Card</h5>
        <p class="card-text">Deskripsi Card</p>
        <a href="#" class="btn btn-primary">Button</a>
    </div>
</div>
```



Penjelasan dari struktur di atas adalah sebagai berikut:

- div dengan kelas card adalah container utama dari card.
- img dengan kelas card-img-top adalah gambar header dari card. Anda bisa mengganti sumber gambar dengan gambar yang Anda inginkan.
- div dengan kelas card-body adalah container dari konten card seperti judul, deskripsi, dan tombol.
- h5 dengan kelas card-title adalah judul card.
- p dengan kelas card-text adalah deskripsi dari card.
- a dengan kelas btn btn-primary adalah tombol di card. Anda bisa mengubah teks tombol dan kelas lainnya sesuai kebutuhan.

Anda bisa menambahkan kelas lain seperti card-header (untuk header card), card-footer (untuk footer card), dan card-deck (untuk mengelompokkan beberapa card menjadi satu baris) sesuai kebutuhan.

Berikut ini adalah contoh card dengan header, footer, dan beberapa card dalam satu baris menggunakan kelas-kelas di atas:

```
<div class="card">
    <div class="card-header">Header Card</div>
    
    <div class="card-body">
        <h5 class="card-title">Judul Card 1</h5>
```

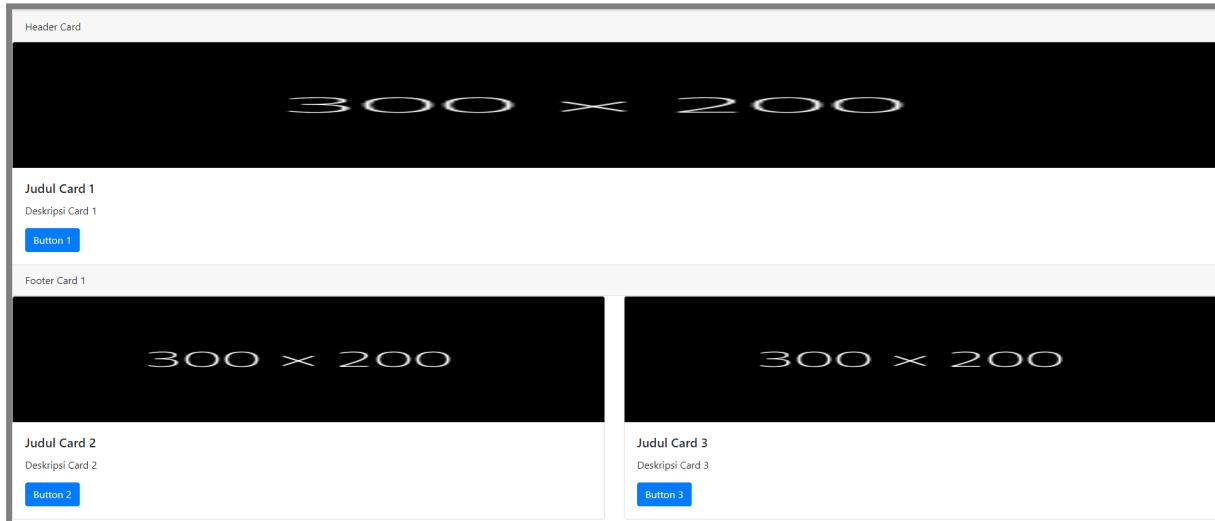
```

        <p class="card-text">Deskripsi Card 1</p>
        <a href="#" class="btn btn-primary">Button 1</a>
    </div>
    <div class="card-footer">Footer Card 1</div>
</div>

<div class="card-deck">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Judul Card 2</h5>
            <p class="card-text">Deskripsi Card 2</p>
            <a href="#" class="btn btn-primary">Button
2</a>
        </div>
    </div>
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Judul Card 3</h5>
            <p class="card-text">Deskripsi Card 3</p>
            <a href="#" class="btn btn-primary">Button
3</a>
        </div>
    </div>

```

```
</div>  
</div>
```



Carousel

Carousel di Bootstrap 4 adalah sebuah komponen yang berfungsi sebagai slider atau penayangan gambar secara otomatis dengan transisi yang halus. Carousel ini bisa digunakan untuk membuat tampilan yang menarik pada halaman web.

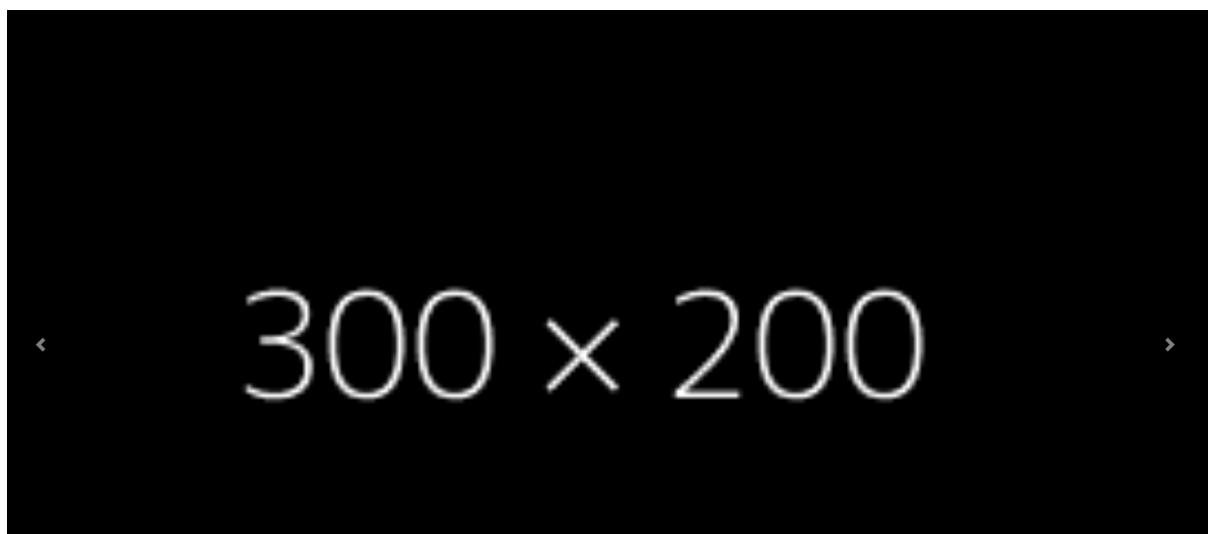
Berikut ini adalah struktur dasar dari carousel di Bootstrap 4:

```
<div  
    id="carouselExampleIndicators"  
    class="carousel slide"  
    data-ride="carousel"  
>  
    <ol class="carousel-indicators">  
        <li  
            data-target="#carouselExampleIndicators"  
            data-slide-to="0"  
            class="active"  
        ></li>  
        <li  
            data-target="#carouselExampleIndicators"
```

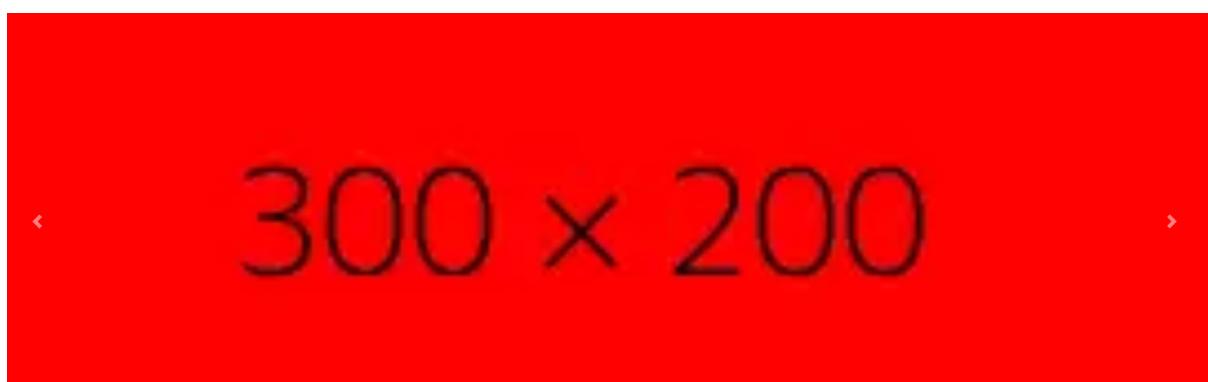
```
        data-slide-to="1"
    ></li>
<li
    data-target="#carouselExampleIndicators"
    data-slide-to="2"
></li>
</ol>
<div class="carousel-inner">
    <div class="carousel-item active">
        
    </div>
    <div class="carousel-item">
        
    </div>
    <div class="carousel-item">
        
    </div>
</div>
<a
    class="carousel-control-prev"
    href="#carouselExampleIndicators"
```

```
        role="button"
        data-slide="prev"
    >
    <span
        class="carousel-control-prev-icon"
        aria-hidden="true"
    ></span>
    <span class="sr-only">Previous</span>
</a>
<a
    class="carousel-control-next"
    href="#carouselExampleIndicators"
    role="button"
    data-slide="next"
>
    <span
        class="carousel-control-next-icon"
        aria-hidden="true"
    ></span>
    <span class="sr-only">Next</span>
</a>
</div>
```

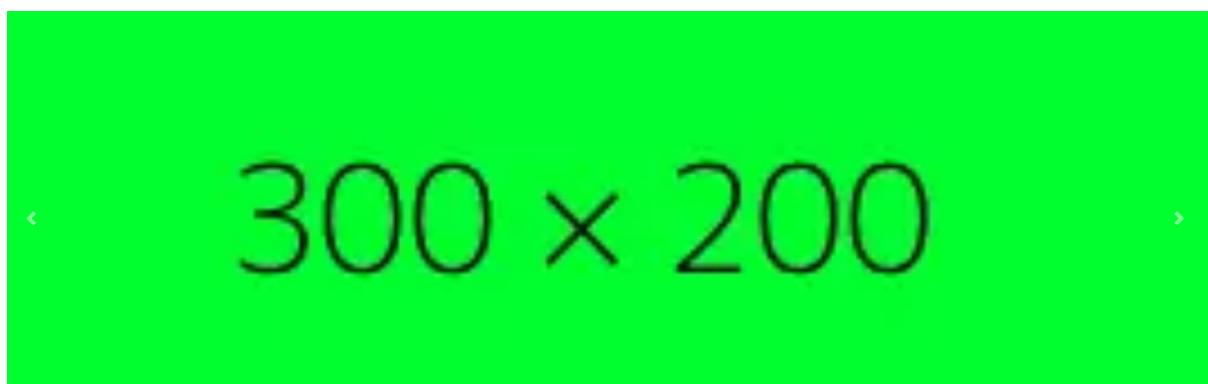
slider1



slider2



slider3



Alerts

Alerts di Bootstrap 4 adalah komponen yang berfungsi untuk menampilkan pesan atau notifikasi pada halaman web. Alerts tersedia dalam beberapa jenis, seperti primary, secondary, success, danger, warning, dan info. Anda bisa menggunakan alerts untuk

memberikan feedback atau menginformasikan pengguna tentang sesuatu yang terjadi pada halaman web.

Berikut ini adalah struktur dasar dari alerts di Bootstrap 4:

```
<div class="alert alert-primary" role="alert">  
  A simple primary alert—check it out!  
</div>
```

Penjelasan dari struktur di atas adalah sebagai berikut:

- div dengan kelas alert adalah container untuk alerts.
- div juga memiliki kelas tambahan seperti alert-primary untuk menentukan jenis alerts.
- role="alert" adalah atribut yang menandakan bahwa ini adalah sebuah alert.
- Teks di dalam div adalah pesan atau notifikasi yang ingin ditampilkan.

Anda bisa menambahkan kelas lain seperti alert-dismissible (untuk menambahkan tombol close) dan fade (untuk menambahkan efek fade in/out) sesuai kebutuhan.

Berikut ini adalah contoh penggunaan alerts dengan beberapa jenis:

```
<div class="alert alert-primary" role="alert">  
  A simple primary alert—check it out!  
</div>  
  
<div class="alert alert-secondary" role="alert">  
  A simple secondary alert—check it out!  
</div>  
  
<div class="alert alert-success" role="alert">  
  A simple success alert—check it out!  
</div>  
  
<div class="alert alert-danger" role="alert">
```

```
A simple danger alert—check it out!  
</div>  
  
<div class="alert alert-warning" role="alert">  
    A simple warning alert—check it out!  
</div>  
  
<div class="alert alert-info" role="alert">  
    A simple info alert—check it out!  
</div>  
  
<div class="alert alert-light" role="alert">  
    A simple light alert—check it out!  
</div>  
  
<div class="alert alert-dark" role="alert">  
    A simple dark alert—check it out!  
</div>
```

A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

A simple danger alert—check it out!

A simple warning alert—check it out!

A simple info alert—check it out!

A simple light alert—check it out!

A simple dark alert—check it out!

Anda juga bisa menambahkan teks tambahan seperti tombol pada alerts seperti contoh di bawah ini:

```
<div class="alert alert-success alert-dismissible fade show"  
role="alert">  
    <strong>Success!</strong> Pesan telah terkirim.
```

```
<button type="button" class="close" data-dismiss="alert"
aria-label="Close">
  <span aria-hidden="true">&times;</span>
</button>
</div>
```

Kode di atas menampilkan alerts dengan teks pesan dan tombol close. Anda bisa menambahkan kelas tambahan seperti alert-success dan alert-dismissible untuk menentukan jenis alerts dan menambahkan tombol close.

nb : untuk tambahkan script js untuk menjalankan fungsi closenya

```
<script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0J
z8D6OP9dO5Vg3Q9ct" crossorigin="anonymous"></script>
```

Bab 4 Javascript



sumber gambar wikipedia (https://en.wikipedia.org/wiki/Brendan_Eich)

sejarah Javascript

JavaScript adalah bahasa pemrograman yang pertama kali dibuat oleh Brendan Eich pada tahun 1995 ketika ia bekerja di Netscape Communications Corporation. Saat itu, Netscape sedang mencari cara untuk membuat tampilan halaman web lebih interaktif dan dinamis. Sebelumnya, tampilan web hanya terdiri dari teks dan gambar statis yang tidak bisa diubah oleh pengguna.

Eich mengembangkan JavaScript dalam waktu 10 hari dengan menggabungkan konsep-konsep dari bahasa pemrograman lain seperti Java, C, dan Perl. Nama asli dari bahasa pemrograman ini sebenarnya adalah Mocha, namun kemudian diubah menjadi LiveScript, dan akhirnya disepakati untuk menggunakan nama JavaScript untuk memanfaatkan popularitas Java pada waktu itu.

Pada awalnya, JavaScript hanya digunakan di browser Netscape Navigator, namun kemudian diadopsi oleh browser-browser lain seperti Internet Explorer, Firefox, dan Chrome. Hal ini menjadikan JavaScript sebagai bahasa pemrograman yang sangat penting dan populer dalam pengembangan aplikasi web.

Seiring berjalananya waktu, JavaScript terus berkembang dan memiliki berbagai versi yang lebih baik dan lebih canggih. Pada tahun 2015, rilis terbaru dari JavaScript yaitu ECMAScript

6 (ES6) atau sering disebut dengan ECMAScript 2015. ECMAScript 6 membawa banyak perubahan dan penambahan fitur baru pada bahasa pemrograman JavaScript, yang memungkinkan programmer untuk membuat kode yang lebih efisien dan mudah dipelajari. Saat ini, JavaScript digunakan dalam berbagai macam aplikasi, mulai dari website sederhana hingga aplikasi web kompleks dan game.

jadi apa itu Javascript?

javascript adalah bahasa pemrograman yang berfungsi untuk membuat website kita lebih dinamis, kita bisa analogikan jika HTML adalah pondasi, CSS untuk tampilannya, maka Javascript berfungsi sebagai gerak dari website nya agar lebih interaktif

Cara memasukan Javascript ke Dalam HTML

ada dua cara memasukan Javascript ke dalam HTML

1.internal Javascript

Membuat javascript langsung ke kode HTML dilakukan dengan cara menambahkan kode JavaScript di HTML kedalam tag `<script>"kode javascript"</script>`. Kode JavaScript yang ditambahkan bisa diletakkan didalam tag `<head>` atau tag `<body>`. Cara pertama ini cocok dipakai untuk penggunaan javascript untuk satu halaman

2.external Javascript

Cara kedua ini cocok dipakai kalau kita memerlukan penggunaan JavaScript di berbagai halaman. File JavaScript ditulis dengan ekstensi `namefile.js`. File ini di-reference dari dalam dokumen HTML, sama seperti mereferensikan dokumen CSS.

DOM Javascript

DOM atau Document Object Model adalah suatu antarmuka untuk merepresentasikan dokumen web sebagai objek. Dengan DOM, seluruh komponen dari halaman web dapat diakses dan dimanipulasi sesuai keinginan kita. Komponen-komponen pada halaman website yang dapat dimanipulasi mencakup elemen HTML, atribut pada elemen, text, dll.

DOM dapat dimanipulasi seperti dibuat baru, diubah, dan dihapus dengan menggunakan code javascript.

Selector DOM

Membuat selektor artinya kita ingin mengakses sebuah element atau komponen pada halaman web untuk kemudian kita dapat lakukan manipulasi di dalamnya. biasanya kita menampung hasil selektor pada sebuah variabel agar bisa digunakan kembali.

document.getElementById()

untuk mengakses sebuah elemen berdasarkan id dari elemen kita dapat gunakan selector `getElemenById()`. Hasil selektor ini adalah sebuah elemen yang dapat kita lakukan manipulasi.

cara membuat selektor dengan id yaitu dengan sintaks sbb:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1 id="judul">Berlatih DOM</h1>

    <script>
        var judul = document.getElementById("judul")
        judul.innerHTML = "Hello";
    </script>
</body>
</html>
```

code diatas kita memanipulasi object isi dari element `<h1>` yang memiliki attribute `id="judul"` disini kita menganti Text “Berlatih Dom” menjadi “Hello”

document.getElementsByClassName()

Untuk mengakses beberapa element sekaligus , kita dapat gunakan selektor menggunakan class. hasil selektor ini yaitu berupa HTML Collection atau kumpulan element HTML yang bisa kita akses sebagai array.

```
<!DOCTYPE html>
<html>
<head>
<style>
.list-menu {
    padding:8px;
}
</style>
</head>
<body>

<ul>
    <li class="list-menu">goreng tempe</li>
    <li class="list-menu">mendoan</li>
    <li class="list-menu">tahu bulat</li>
</ul>
<script>
    var listMenu = document.getElementsByClassName("list-menu")
    //mengakses array index ke 0
    listMenu[0].style.backgroundColor = "blue";
</script>
</body>
</html>
```

- goreng tempe
- mendoan
- tahu bulat

code diatas kita menambah style pada class list-menu array index ke 0 menjadi warna biru(blue)

Manipulasi DOM

Melakukan manipulasi DOM adalah salah satu cara agar tampilan web menjadi interaktif. beberapa contoh manipulasi yaitu dengan cara membuat element baru, mengubahnya atau menghapusnya. Terdapat beberapa metode untuk memanipulasi DOM di antaranya :

.innerHTML

metode innerHTML yaitu metode untuk **mengisi atau mengganti isi** dari sebuah element HTML dengan apapun yang kita inginkan. Kita dapat isi dengan sebuah teks node atau bahkan kita isi dengan element HTML di dalam DOM yang sedang kita akses.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>
<body>

    <div id="menu-makanan">
```

```

</div>

<script>
    var menuMakanan = document.getElementById("menu-makanan")

    menuMakanan.innerHTML = `<h2>Ini adalah menu makanannya</h2>
                            <ul>
                                <li class="list-menu">goreng
                                tempe</li>
                                <li
                                class="list-menu">mendoan</li>
                                <li class="list-menu">tahu
                                bulat</li>
                            </ul>`
</script>
</body>
</html>

```

dari code diatas kita memasukan element `<h2>` dan list `` kedalam element dengan attribute id menu makanan

DOM Events

Event adalah sebuah aktivitas atau sebuah kejadian yang terjadi di dalam sebuah DOM. Kejadian tersebut dapat kita tangkap atau didengarkan(listen) untuk kemudian kita lakukan sesuatu hal yang lain. hal ini tentunya membuat halaman web kita menjadi sangat interaktif.

Cara menangkap sebuah event ada 2 cara yaitu:

inline attribute dengan tag HTML

beberapa tag pada HTML memiliki attribute bawaan berupa event, contohnya `<input onchange="alert('hello')">` atau `<button onclick="alert('i am clicked')"> Klik button </button>`

menambahkan addEventListener()

1. menangkap event submit pada form

untuk menangkap kejadian ketika sebuah form disubmit kita dapat tambahkan addEventListener untuk event submit. contohnya di bawah ini sebuah form dengan id formConditional ditangkap event submit nya.

di dalam function addEventListener terdapat dua parameter yaitu nama eventnya dan function handler nya.

nama event yang ingin ditangkap adalah "submit". function handler adalah function yang dijalankan ketika event tersebut didengarkan. function handler akan punya satu parameter yaitu event. Agar form tersebut tidak merefresh halaman maka kita gunakan `event.preventDefault()`.

```
<!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>Conditional HTML</title>
</head>
<body>
    <h1>Berlatih Kondisional</h1>
    <h3>Masukkan nama dan peranmu!</h3>
    <form action="#" id="formConditional">
        <input type="text" id="name" placeholder="Isi nama di
sini">
        <input type="text" id="role" placeholder="Isi peran di
sini">
        <input type="submit" value="Mulai Permainan">
    </form>

    <h3>Jawaban dari console game: </h3>
    <div id="jawaban">

    </div>
<script>
```

```

var form = document.getElementById("formConditional")
var jawaban = document.getElementById("jawaban")

form.addEventListener("submit", function(event) {
    event.preventDefault()
    var name = document.getElementById("name").value
    var role = document.getElementById("role").value

    jawaban.innerHTML = name + " " + role;
})

</script>
</body>
</html>

```

Variable

Variable adalah suatu blok data untuk menampung sekumpulan data dengan berbagai tipe data apapun. Dengan variable kita bisa menyimpan suatu nilai untuk kemudian kita olah kembali pada program kita. Untuk deklarasi variable dalam javascript kita bisa gunakan sintaks var lalu diikuti nama variablenya.

Untuk menulis satu variabel di dalam Javascript digunakan tanda var . Aturannya adalah

- tidak boleh menggunakan angka di depan nama variabel tapi boleh menggunakan angka di tempat lain.
- tidak boleh memuat spesial karakter (~!@#\$%. dan lain lain)
- tidak boleh menggunakan spasi jika menggunakan 2 suku kata pada saat pembuatan variabel (untuk mengatasinya bisa menggunakan cara penulisan camelCase contoh: namaDepan atau snake_case contoh nama_depan)

Data Type

Data Type atau dalam bahasa indonesia Tipe Data adalah sekumpulan informasi yang memiliki nilai dan karakteristik tertentu. Beberapa contoh tipe data pada javascript di antaranya:

Number : tipe data angka

String : tipe data berupa text atau kumpulan karakter, biasanya string dibungkus dalam tanda petik ganda (double quote) atau tanda petik tunggal (single quote).

Boolean: tipe data dengan nilai true atau false

```
var name = "John" // Tipe data string  
var angka = 12 //Tipe data number  
var angkaDecimal = 2,5 //Tipe data number  
var NiceDay = true//Type data boolean
```

Operator Javascript

Operator adalah karakter khusus yang merepresentasikan sebuah tindakan. Operator terbagi ke dalam beberapa jenis:

Operator Aritmatika

Operator yang melibatkan operasi matematika seperti tambah, kurang, kali, bagi.

Tambah (+)

Kurang (-)

Kali (*)

Bagi (/)

Modulus (%)

Modulus adalah sisa bagi. Contohnya $5\%3$ hasilnya adalah 2, $100\%5$ hasilnya 0.

Operator Assignment (=)

Operator untuk mendaftarkan atau meng-assign suatu nilai ke dalam suatu variable

```
var angka = 1;  
angka = 10 // Contoh assignment variable angka yang tadinya nilai  
1 menjadi 10
```

nb: variabel diatas pemanggilannya harus sama dengan nama variabel yang dibuat baik dari huruf besar maupun huruf kecilnya contoh jika kita memanggil variabel code diatas dengan Angka = 10 maka akan error undefined karena huruf a di awal pemanggilan beda menggunakan huruf besar beda dengan nama variabel yang dibuat dimana awal a nya menggunakan huruf kecil

Operator Perbandingan

Operator yang membandingkan suatu nilai dengan nilai yang lain. Hasil dari perbandingan ini akan dikembalikan dalam tipe data boolean true atau false.

- **Equal Operator (==)**

```
var angka = 100  
console.log(angka == 100) // true  
console.log(angka == 20) // false
```

- **Not Equal (!=)**

```
var sifat = "rajin"  
console.log(sifat != "malas") // true  
console.log(sifat != "bandel") //true
```

- **Strict Equal (===)** Selain membandingkan dua nilai nya, strict equal juga membandingkan tipe datanya apakah sama atau tidak

```
var angka = 8  
console.log(angka === "8") // true, padahal "8" adalah string.  
console.log(angka === "8") // false, karena tipe data nya berbeda  
console.log(angka === 8) // true
```

- **Strict not Equal (!==)** Kebalikan dari strict equal.

```
var angka = 11
console.log(angka != "11") // false, padahal "11" adalah string
console.log(angka !== "11") // true, karena tipe datanya berbeda
console.log(angka !== 11) // false
```

- Kurang dari & Lebih Dari (<, >, <=, >=)

```
var number = 17
console.log( number < 20 ) // true
console.log( number > 17 ) // false
console.log( number >= 17 ) // true, karena terdapat sama dengan
console.log( number <= 20 ) // true
```

Operator Kondisional, Operator yang mengkombinasikan dua nilai kebenaran .

Terdapat operator AND (`&&`) dan OR (`||`)

- OR (||)

```
console.log(true || true); // true
console.log(true || false); // true
console.log(true || false || false); // true
console.log(false || false); // false


- AND ( && )


console.log(true && true); // true
console.log(true && false); // false
console.log(false && false); // false
console.log(false && true && true); // false
console.log(true && true && true); // true
```

String

String Properties

.length

mengembalikan panjang atau jumlah karakter pada suatu string.

```

<!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>Conditional HTML</title>
</head>
<body>
    <div id="jawaban">

    </div>
</body>

<script>
    var jawaban = document.getElementById("jawaban");
    var kalimat = "Javascript is awesome";

    jawaban.innerHTML = kalimat.length //21
</script>
</html>

```

.charAt([indeks])

Mengembalikan karakter pada indeks yang diinginkan

```

<!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>String Javascript</title>

```

```

</head>
<body>
    <div id="jawaban">
        </div>
    </body>

<script>
    var jawaban = document.getElementById("jawaban");
    var kalimat = "i am a string";

    jawaban.innerHTML = kalimat.charAt(3) //m
</script>
</html>

```

.concat([string])

Menggabungkan beberapa string dan mengembalikannya menjadi string baru.

```

<!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>String Javascript</title>
</head>
<body>
    <div id="jawaban">

```

```

</div>

</body>

<script>
    var jawaban = document.getElementById("jawaban");
    var string1 = 'good';
    var string2 = 'luck';

    jawaban.innerHTML = string1.concat(string2) //goodluck

</script>
</html>

```

.indexOf([string/karakter])

Mengembalikan indeks dari string/karakter yang dicari, yang pertama kali ditemukan, atau -1 apabila tidak ditemukan.

```

<!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>String Javascript</title>
</head>
<body>
    <div id="jawaban">

    </div>

```

```

</body>

<script>
    var jawaban = document.getElementById("jawaban");
    var text = 'dung dung ces!';

    jawaban.innerHTML = text.indexOf('c') //10

</script>
</html>

```

.substring([indeks awal], [indeks akhir (optional)])

Mengembalikan potongan string mulai dari indeks pada parameter pertama (indeks awal) sampai dengan indeks pada parameter kedua (indeks akhir). Bila parameter kedua tidak ditentukan, maka secara otomatis berakhir pada karakter terakhir. Karakter pada indeks yang ditentukan pada parameter kedua tidak diikutkan sebagai output.

```

<!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>String Javascript</title>
</head>
<body>
    <div id="jawaban">

    </div>

</body>

```

```

<script>
    var jawaban = document.getElementById("jawaban");
    var text = 'Lykan Hypersport';

    jawaban.innerHTML = text.substring(6, 16) //Hypersport'
</script>
</html>

```

.substr([indeks awal], [jumlah karakter yang diambil (optional)])

Mendapatkan potongan string mulai dari indeks pada parameter pertama (indeks awal) dengan jumlah indeks pada parameter kedua (jumlah karakter). Bila parameter kedua tidak ditentukan, maka secara otomatis berakhir pada karakter terakhir. Karakter pada indeks yang ditentukan pada parameter kedua tidak diikutkan sebagai output.

```

<!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>String Javascript</title>
</head>
<body>
    <div id="jawaban">

    </div>

</body>

<script>
    var jawaban = document.getElementById("jawaban");
    var text = 'Lykan Hypersport';

```

```
jawaban.innerHTML = text.substr(6,10) //Hypersport'  
  
</script>  
</html>
```

.toUpperCase()

Mengembalikan string baru dengan semua karakter yang diubah menjadi huruf kapital.

```
<!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>String Javascript</title>  
</head>  
<body>  
    <div id="jawaban">  
  
    </div>  
  
</body>  
  
<script>  
    var jawaban = document.getElementById("jawaban");  
    var letter = 'This Letter Is For You';  
  
    jawaban.innerHTML = letter.toUpperCase() //THIS LETTER IS FOR  
YOU  
  
</script>  
</html>
```

.toLowerCase()

Mengembalikan string baru dengan semua karakter yang diubah menjadi huruf kecil

```
<!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>String Javascript</title>
</head>
<body>
    <div id="jawaban">

    </div>

</body>

<script>
    var jawaban = document.getElementById("jawaban");
    var letter = 'This Letter Is For You';

    jawaban.innerHTML = letter.toLowerCase() //this letter is for
you

</script>
</html>
```

Function

Function adalah sebuah blok kode yang disusun sedemikian rupa untuk menjalankan sebuah tindakan. Blok kode ini dibuat untuk dapat bisa digunakan kembali. Cara atau bentuk penulisan function adalah sebagai berikut:

```
function nama_function(parameter 1, parameter 2, ...) {  
    [Isi dari function berupa tindakan]  
    return [expression];  
}
```

Kode di atas tidak dapat kita copy-paste kan langsung, melainkan hanya sebuah bentuk penulisan function. Sebuah function, umumnya melakukan tindakan dan sebelum function berakhir, function bisa mengembalikan nilai dengan cara menambahkan sintaks return.

Kita juga dapat mengirimkan nilai ke dalam sebuah function dengan mencantumkannya ke dalam tanda kurung dalam penulisan function. Untuk mengirimkan nilai lebih dari satu, gunakan tanda , sebagai pemisah.

Contoh Function 1: Function sederhana tanpa parameter

```
<!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>Function Javascript</title>  
</head>  
<body>  
    <div id="jawaban">  
  
    </div>
```

```

</body>

<script>
    var jawaban = document.getElementById("jawaban");
    function munculkanHello() {
        return "Hello World"
    }

    jawaban.innerHTML = munculkanHello() //Hello World
}

</script>
</html>

```

Contoh Function 2: Function dengan parameter

```

<!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>Function Javascript</title>
</head>
<body>
    <div id="jawaban">

    </div>

</body>

<script>
    var jawaban = document.getElementById("jawaban");
    function kelipatanDua(angka) {
        return 2*angka

```

```
}
```

```
jawaban.innerHTML = kelipatanDua(10) //20
```

```
</script>
```

```
</html>
```

Contoh Function 3: Pengiriman parameter lebih dari satu

```
<!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>Function Javascript</title>
</head>
<body>
    <div id="jawaban">
```

```
</div>
```

```
</body>
<script>
    var jawaban = document.getElementById("jawaban");
    function tambah(angka1, angka2) {
        return angka1 + angka2
    }

    jawaban.innerHTML = tambah(10,30) //40
</script>
</html>
```

Kondisional dengan if / else if / else

Kondisional Sangat sering digunakan Saat menulis kode, karena kita dapat mengatur tindakan apa yang akan terjadi disetiap kondisinya

kita dapat menggunakan pernyataan bersyarat dalam kode

Dalam JavaScript kami memiliki conditional bersyarat berikut diantaranya:

Gunakan **if** untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan benar

Gunakan **else** untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sama salah

Gunakan **else if** untuk menentukan kondisi baru yang akan diuji, jika kondisi pertama salah maka akan masuk ke kondisi berikutnya

Kondisional if

```
<!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>Conditional HTML</title>
</head>
<body>
    <h1>Berlatih Kondisional</h1>

    <h3>Masukkan nama dan peranmu!</h3>
    <form action="#" id="formConditional">
        <input type="text" id="name" placeholder="Isi nama di
sini">
```

```

<input type="text" id="role" placeholder="Isi peran di
sini">
<input type="submit" value="Mulai Permainan">
</form>

<h3>Jawaban dari console game: </h3>
<div id="jawaban">

</div>
</body>

<script>

var form = document.getElementById("formConditional")
var jawaban = document.getElementById("jawaban")

form.addEventListener("submit", function(e) {
    e.preventDefault()

    var name = document.getElementById("name").value
    var role = document.getElementById("role").value

    var jawabanKonsol = "";

    if(name == "" && role == "") {
        jawabanKonsol = "Nama dan role harus diisi"
    }

    jawaban.innerHTML = jawabanKonsol
} )

```

```
</script>  
</html>
```

Berlatih Kondisional

Masukkan nama dan peranmu!

[

Jawaban dari console game:

dari code diatas kondisi jika name dan role pada inputan tidak diisi maka output nya “Nama dan Role harus diisi”

Kondisional else if

```
<!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>Conditional HTML</title>  
</head>  
<body>  
    <h1>Berlatih Kondisional</h1>  
  
    <h3>Masukkan nama dan peranmu!</h3>  
    <form action="#" id="formConditional">  
        <input type="text" id="name" placeholder="Isi nama di  
sini">  
        <input type="text" id="role" placeholder="Isi peran di  
sini">  
        <input type="submit" value="Mulai Permainan">  
    </form>  
</body>
```

```

</form>

<h3>Jawaban dari console game: </h3>
<div id="jawaban">

</div>
</body>

<script>

var form = document.getElementById("formConditional")
var jawaban = document.getElementById("jawaban")

form.addEventListener("submit", function(e) {
    e.preventDefault()

    var name = document.getElementById("name").value
    var role = document.getElementById("role").value

    var jawabanKonsol = "";

    if(name == "" && role == "") {
        jawabanKonsol = "Nama dan role harus diisi"
    }else if(name != "" && role== "") {
        jawabanKonsol = "Halo " + name + " role harus
diisi"
    }else if(name != "" && role== "admin") {
        jawabanKonsol = "Halo " + name + " anda bertugas
sebagai " + role + "anda dapat mengakses semua halaman"
    }else if(name != "" && role== "operator") {
        jawabanKonsol = "Halo " + name + " anda bertugas
sebagai " + role + "hanya dapat mangakses beberapa halaman"
    }
})

```

```
jawaban.innerHTML = jawabanKonsol  
}  
  
</script>  
</html>
```

else if jika name tidak kosong atau diinput && rolenya kosong

Berlatih Kondisional

Masukkan nama dan peranmu!

rezky	Isi peran di sini	Mulai Permainan
-------	-------------------	-----------------

Jawaban dari console game:

Halo rezky role harus diisi

else if jika name tidak kosong atau diinput && rolenya adalah admin

Berlatih Kondisional

Masukkan nama dan peranmu!

rezky	admin	Mulai Permainan
-------	-------	-----------------

Jawaban dari console game:

Halo rezky anda bertugas sebagai admin anda dapat mengakses semua halaman

else if jika name tidak kosong atau diinput && rolenya adalah operator

Berlatih Kondisional

Masukkan nama dan peranmu!

rezky operator [Mulai Permainan](#)

Jawaban dari console game:

Halo rezky anda bertugas sebagai operator hanya dapat mengakses beberapa halaman

dari output diatas terdapat 3 kondisi

- jika name tidak kosong dan output kosong
- jika name tidak kosong dan role nya diisi admin
- jika name tidak kosong dan role nya diisi operator

Kondisional else

```
<!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>Conditional HTML</title>
</head>
<body>
    <h1>Berlatih Kondisional</h1>

    <h3>Masukkan nama dan peranmu!</h3>
    <form action="#" id="formConditional">
        <input type="text" id="name" placeholder="Isi nama di
sini">
        <input type="text" id="role" placeholder="Isi peran di
sini">
        <input type="submit" value="Mulai Permainan">
    </form>
</body>
</html>
```

```

</form>

<h3>Jawaban dari console game: </h3>
<div id="jawaban">

</div>
</body>

<script>

var form = document.getElementById("formConditional")
var jawaban = document.getElementById("jawaban")

form.addEventListener("submit", function(e) {
    e.preventDefault()

    var name = document.getElementById("name").value
    var role = document.getElementById("role").value

    var jawabanKonsol = "";

    if(name == "" && role == "") {
        jawabanKonsol = "Nama dan role harus diisi"
    }else if(name != "" && role== "") {
        jawabanKonsol = "Halo " + name + " role harus
diisi"
    }else if(name != "" && role== "admin") {
        jawabanKonsol = "Halo " + name + " anda bertugas
sebagai " + role + "anda dapat mengakses semua halaman"
    }else if(name != "" && role== "operator") {
        jawabanKonsol = "Halo " + name + " anda bertugas
sebagai " + role + "hanya dapat mangakses beberapa halaman"
    }else{

```

```

        jawabanKonsol = "role tidak ditemukan"
    }

    jawaban.innerHTML = jawabanKonsol

}

</script>
</html>

```

Berlatih Kondisional

Masukkan nama dan peranmu!

rezky	uoiuo	Mulai Permainan
-------	-------	-----------------

Jawaban dari console game:

role tidak ditemukan

dari kode diatas kondisi else di gunakan sebagai kondisi terakhir jika kondisi diatasnya yaitu if dan else if nya sudah bernilai salah

Kondisi Bersarang

kondisi bersarang adalah kondisi dimana didalam sebuah kondisional terdapat kondisional lagi didalamnya

```

<!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>Conditional HTML</title>
</head>
<body>
    <h1>Berlatih Kondisional</h1>

    <h3>Masukkan nama dan peranmu!</h3>
    <form action="#" id="formConditional">
        <input type="text" id="name" placeholder="Isi nama di
sini">
        <input type="text" id="role" placeholder="Isi peran di
sini">
        <input type="submit" value="Mulai Permainan">
    </form>

    <h3>Jawaban dari console game: </h3>
    <div id="jawaban">

    </div>
</body>

<script>

var form = document.getElementById("formConditional")
var jawaban = document.getElementById("jawaban")



form.addEventListener("submit", function(e) {
    e.preventDefault()

    var name = document.getElementById("name").value
    var role = document.getElementById("role").value

    var jawabankonsol = "";

```

```

        if(name == "" && role == "") {
            jawabanKonsol = "Nama dan role harus diisi"
        }else if(name != "") {
            if(role=="") {
                jawabanKonsol = "Halo " + name + " role harus
diisi"
            }else if(role=="admin") {
                jawabanKonsol = "Halo " + name + " anda
bertugas sebagai " + role + " anda dapat mengakses semua halaman"
            }else if(role=="operator") {
                jawabanKonsol = "Halo " + name + " anda bertugas
sebagai " + role + " hanya dapat mangakses beberapa halaman"
            }else{
                jawabanKonsol = "role tidak ditemukan"
            }
        }else{
            jawabanKonsol = "input tidak ditemukan"
        }

        jawaban.innerHTML = jawabanKonsol
    }
}

</script>
</html>

```

Looping

Loop/Iteration adalah tindakan mengulang / merepetisi sebuah proses, dengan tujuan untuk mendapatkan deret hasil, atau dengan tujuan mendapatkan hasil tertentu dengan repetisi. Setiap proses repetisi ini disebut sebagai Iteration atau Looping.

Untuk melakukan looping/iteration, JavaScript menyediakan beberapa jenis iteration, yaitu:

1. while-loop
2. for-loop

While-loop

While-loop adalah iterasi yang akan mengulang sebuah proses dengan sebuah kondisi tertentu.

Berikut adalah cara atau struktur penulisan while-loop:

```
while([Kondisi]) { // Kondisi yang menentukan apakah program akan
// melakukan iterasi.
// Berupa boolean atau true/false.[Proses]
// Merupakan proses yang akan dijalankan dalam satu iterasi
}
```

Pada while-loop, statement while akan mengambil sebuah nilai true atau false dari hasil kondisi yang diberikan. Apabila statement while mendapatkan nilai true, maka proses yang berada didalam curly bracket ({ }) akan dijalankan. Looping akan terus dilakukan selama kondisi while-loop masih true.

Untuk mencegah looping berjalan terus-menerus, dilakukan sebuah proses yang akan mengubah kondisi yang sedemikian rupa yang bertujuan untuk menghentikan looping dengan menghasilkan kondisi yang false.

```
<!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>Looping HTML</title>
</head>
<body>
```

```

<h1>Berlatih Looping</h1>
<div id="jawaban">

</div>
</body>

<script>
    var jawaban = document.getElementById("jawaban")

    var jawabanKonsol = "";
    var flag = 1;
    while(flag <= 10) {
        jawabanKonsol += 'Iterasi ke-' + flag + "<br>";
        flag++;
    }

    jawaban.innerHTML = jawabanKonsol
}

</script>
</html>

```

- Loop akan terus berjalan selama nilai flag masih dibawah 10
- Menampilkan nilai flag pada iterasi tertentu
- Mengubah nilai flag dengan menambahkan 1

Berlatih Looping

Iterasi ke-1
Iterasi ke-2
Iterasi ke-3
Iterasi ke-4
Iterasi ke-5
Iterasi ke-6
Iterasi ke-7
Iterasi ke-8
Iterasi ke-9
Iterasi ke-10

For-loop

For-loop adalah bentuk lain dari iterasi, dimana statement for menjadi kontrol atas loop yang dilakukan. Hal ini yang menjadi pembeda antara for-loop dengan while-loop.

Berikut adalah cara atau struktur penulisan for-loop:

```
for([Inisialisasi], [Kondisi], [Incremental/Decremental]) {  
    [Proses] // Merupakan proses yang akan dijalankan dalam satu  
    iterasi  
}
```

Pada for-loop, statement for akan menampung tiga parameter, yaitu sebut saja inisialisasi, kondisi, dan incremental/decremental. Ketiga parameter ini akan menjadi kontrol kapan loop ini harus berhenti. Pada parameter pertama, yaitu inisialisasi, sebuah variable diberikan nilai awal atau default. Pada parameter kedua, yaitu kondisi, for-loop akan terus berjalan selama kondisi ini masih terpenuhi, dengan kata lain, mengandung nilai true. Pada parameter kedua, yaitu incremental/decremental, variabel yang menjadi kontrol terhadap loop ini akan diubah nilainya.

Best Practice: Walaupun memang for-loop dapat mengubah kondisi di dalam proses, namun best practice dari penggunaan for-loop adalah seluruh kendali atau kontrol dari looping ditentukan oleh variable yang diinisialisasi, di increment/decrement, dan juga kondisi for-loop pun menggunakan variable tersebut.

Untuk memudahkan kamu mendapatkan gambaran jelas tentang penggunaan for-loop, mari kita gunakan kedua contoh while-loop dan kita tulis ulang dalam bentuk for-loop.

```
<!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>Looping HTML</title>
</head>
<body>
    <h1>Berlatih Looping</h1>
    <div id="jawaban">

    </div>
</body>

<script>
    var jawaban = document.getElementById("jawaban")
    var jawabanKonsol = "";
    for(var angka = 1; angka <= 10; angka++) {
        jawabanKonsol += 'Iterasi ke-' + angka + "<br>";
    }
    jawaban.innerHTML = jawabanKonsol
</script>
</html>
```

Berlatih Looping

Iterasi ke-1
Iterasi ke-2
Iterasi ke-3
Iterasi ke-4
Iterasi ke-5
Iterasi ke-6
Iterasi ke-7
Iterasi ke-8
Iterasi ke-9
Iterasi ke-10

Array Javascript

Array adalah kumpulan atau tumpukan berbagai data. Cara menuliskan array yaitu dengan kurung siku (`[]`) dan elemen-elemen nya dipisah menggunakan tanda koma (`,`). Setiap elemen dari array memiliki indeks yang dimulai dari 0, 1, 2, dst. Kita dapat memanipulasi array dengan berbagai cara seperti menambahkan dan mengeluarkan elemen dalam array, menggabungkan array, atau bahkan menghapus seluruh elemen Array nya. Kita bisa memasukkan beberapa tipe data yang berbeda ke dalam Array bahkan memasukkan Array ke dalam Array.

Array juga memiliki property `.length` seperti pada string yang berarti panjang dari sebuah array.

Indexed Array

```
var hobbies = ["coding", "cycling", "climbing", "skateboarding"]
```

dari variabel di atas coding masuk ke index ke 0 , cycling masuk ke index 1, climbing masuk ke index ke 2 dan skateboarding masuk ke index 3

Multidimensional Array

Multidimensional array atau array multidimensi adalah array yang berisi array di dalamnya. Jumlah dimensi bergantung seberapa dalam array tersebut memiliki array di dalamnya. Cara akses nilai dari array multidimensi sama seperti array satu dimensi seperti biasa namun jumlah indeksnya terdapat sebanyak dimensi nya.

```
var arrayMulti = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
/* Maka sebagai gambaran, indeks dari array tersebut adalah
[
    [(0,0), (0,1), (0,2)],
    [(1,0), (1,1), (1,2)],
    [(2,0), (2,1), (2,2)]
]
*/
```

di code di atas jika kita ingin mengakses angka 5 didalam variabel kita bisa menjalankan **arrayMulti[1][1]** dan jika kita ingin mengakses angka 7 maka **arrayMulti[2][0]** dan jika ingin mengakses angka 3 maka **arrayMulti[0][2]**

Array Method

.push()

Push adalah metode array untuk menambahkan nilai di belakang elemen terakhir di array. metode push menerima sebuah parameter yaitu nilai yang ingin kita tambahkan ke dalam array.

```
var numbers = [0, 1, 2]
numbers.push(3)
console.log(numbers) // [0, 1, 2, 3]
// Bisa juga memasukkan lebih dari satu nilai menggunakan metode
push
numbers.push(4, 5)
console.log(numbers) // [0, 1, 2, 3, 4, 5]
```

.pop()

Pop adalah kebalikan dari push yaitu menghapus nilai elemen paling terakhir dari sebuah array. metode Pop tidak menerima parameter apapun sehingga metode pop hanya bisa mengeluarkan satu elemen saja yaitu yang paling terakhir dari sebuah array.

```
var numbers = [0, 1, 2, 3, 4, 5]
numbers.pop()
console.log(numbers) // [0, 1, 2, 3, 4]
```

.unshift()

Unshift yaitu menambahkan nilai pada index ke-0 sehingga elemen-elemen sebelumnya bergeser berbeda dengan push yang memasukan nilai di index terakhir

```
var numbers = [0, 1, 2, 3]
numbers.unshift(-1)
console.log(numbers) // [-1, 0, 1, 2, 3]
```

.shift()

Shift kebalikan dari unshift yaitu menghapus nilai pada elemen terdepan index ke-0 dari sebuah array. metode Shift tidak menerima parameter apapun.

```
var numbers = [ 0, 1, 2, 3]
numbers.shift()
console.log(numbers) // [1, 2, 3]
```

.sort()

Sort adalah metode untuk mengurutkan nilai pada array. Secara otomatis, sort akan mengurutkan secara ascending (dari rendah ke tinggi) dan diurutkan berdasarkan unicode dari karakter. Urutan unicode artinya ada karakter yang secara nilai lebih besar dibandingkan dengan karakter yang lainnya. contohnya adalah karakter "b" akan lebih besar daripada "a", karakter "c" lebih besar daripada karakter "b", dst.

```
var animals = ["kera", "gajah", "musang"]
animals.sort()
console.log(animals) // ["gajah", "kera", "musang"]
```

Hal ini akan bermasalah jika kita ingin mengurutkan kata. Secara otomatis metode sort akan membandingkan karakter terdepan dari sebuah kata.

```
var numbers = [12, 1, 3]
numbers.sort()
console.log(numbers) // [1, 12, 3]
```

.slice()

slice adalah metode untuk mengambil irisan dari sebuah array. Metode slice bisa menerima satu atau dua parameter. Parameter pertama adalah nomer index pertama yang akan kita ambil sebagai irisan, sedangkan parameter kedua adalah nomer index terakhir yang ingin kita ambil sebagai irisan.

```
var angka = [0, 1, 2, 3]
var irisan1 = angka.slice(1,3)
console.log(irisan1)//[1, 2, 3]
var irisan2 = angka.slice(0,2)
console.log(irisan2)//[0, 1, 2]
```

Jika parameter kedua tidak diisi maka secara otomatis slice akan mengiris array dari indeks di parameter pertama sampai ke indeks terakhir array tersebut.

```
var angka = [0, 1, 2, 3]
var irisan3 = angka.slice(2)
console.log(irisan3)// [2, 3]
```

Jika diperhatikan, metode slice tidak mengubah Array yang awal dan hasil irisan array dapat kita tampung kembali ke variable yang baru. Oleh karena itu metode Slice juga berfungsi untuk membuat salinan atau "shallow copy" dari sebuah array. Cara membuat shallow copy dengan metode slice yaitu dengan tanpa memberikan parameter apapun pada metode slice.

```
var angka = [0, 1, 2, 3]
var salinAngka = angka.slice()
console.log(salinAngka)// [0, 1, 2, 3]
```

.splice()

Splice yaitu metode untuk menghapus dan/atau menambahkan nilai elemen pada array. Metode splice bisa menerima parameter sebanyak dua atau lebih parameter. Jika ingin menggunakan splice untuk menghapus elemen pada index tertentu maka digunakan 2

paramater. Jika ingin menggunakan splice untuk menambahkan elemen pada index tertentu maka digunakan tiga parameter.

```
array.splice([IndexMulai], [JumlahNilaiYangDihapus],  
[NilaiYang Ditambahkan1], [NilaiYang Ditambahkan2], ...);
```

Berikut contoh penggunaan metode splice()

```
var fruits = [ "banana", "orange", "grape"]  
fruits.splice(1, 0, "watermelon")  
console.log(fruits) // [ "banana", "watermelon", "orange",  
"grape"]
```

Contoh di atas berarti kita mulai mengubah array fruits dari indeks no 1, dan melakukan penghapusan nilai sebanyak 0 (yang berarti tidak menghapus apapun), dan menambahkan nilai "watermelon" pada indeks tersebut.

```
var fruits = [ "banana", "orange", "grape"]  
fruits.splice(0, 2)  
console.log(fruits) // ["grape"]
```

Contoh di atas berarti kita mulai mengubah array fruits dari indeks ke-0 dan melakukan penghapusan data sebanyak 2 elemen.

.split() dan .join()

Metode split yaitu memecah sebuah string sehingga menjadi sebuah array. Split menerima sebuah parameter berupa karakter yang menjadi separator untuk memecah string.

```
var biodata = "name:john,doe"  
var name = biodata.split(":")  
console.log(name) // [ "name", "john,doe"]
```

Metode join yaitu kebalikan dari split yaitu mengubah sebuah array menjadi string dengan cara menggabungkan (join) seluruh elemen array menjadi satu dengan sebuah karakter pemisah yang dikirim melalui parameter. contohnya sebagai berikut:

```
var title = ["my", "first", "experience", "as", "programmer"]
var slug = title.join("-")
console.log(slug) // "my-first-experience-as-programmer"
```

looping array indexed

```
<!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>Looping HTML</title>
</head>
<body>
    <h1>Berlatih Looping Array</h1>
    <ul id="list-item">

    </ul>
</body>

<script>

    var jawaban = document.getElementById("list-item")
    //variabel array
    var indexArray = ["Rezky", "Budy", "Anton", "Aco"]

    //variabel tampung
    var jawabanKonsol = "";
    for(var index = 0; index < indexArray.length; index++) {
```

```

        jawabanKonsol += '<li>Nama Peserta - ' + indexArray[index]
+ "</li>";
    }
    jawaban.innerHTML = jawabanKonsol

</script>
</html>

```

Berlatih Looping Array

- Nama Peserta - Rezky
- Nama Peserta - Budy
- Nama Peserta - Anton
- Nama Peserta - Aco

pada code looping for diatas looping nya mulai dari index ke - 0 `var index = 0;` dan berakhir dengan kondisi panjang dari array nya `indexArray.length;` dan setiap berulang index akan bertambah 1 `index++`

looping array multidimensional

```

<!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>Looping HTML</title>
</head>
<body>
    <h1>Berlatih Looping Array</h1>
        <ul id="list-item">

```

```
</ul>
</body>

<script>

    var jawaban = document.getElementById("list-item")
    //variabel array
    var indexArray = [
        ["Rezky", "27", "Makassar"],
        ["Ardi", "28", "Jakarta"],
        ["Maskur", "27", "Pinrang"],
        ["Asep", "29", "Garut"]
    ]

    //variabel tampung
    var jawabanKonsol = "";
    for(var index = 0; index < indexArray.length; index++) {
        jawabanKonsol += '<li>Nama Peserta - ' +
indexArray[index][0] + "<br> Umur Peserta - " +
indexArray[index][1] + "<br> Kota Peserta - " +
indexArray[index][2] +"</li>';
    }
    jawaban.innerHTML = jawabanKonsol

</script>
</html>
```

Bagian 2 Server Side

BAB 5 PHP

Sejarah PHP



sumber gambar wikipedia (https://en.wikipedia.org/wiki/Rasmus_Lerdorf)

PHP (PHP: Hypertext Preprocessor) adalah bahasa pemrograman server-side yang digunakan untuk mengembangkan aplikasi web dinamis. PHP pertama kali dibuat pada tahun 1994 oleh Rasmus Lerdorf, seorang programmer asal Denmark-Kanada. Awalnya, PHP merupakan singkatan dari "Personal Home Page Tools" yang berfungsi untuk membuat halaman web sederhana.

Apa itu PHP?

Sebuah web yang kita telusuri sehari-hari di internet dapat kita lihat sebagai penggabungan dua sisi yaitu sisi klien dan sisi server. Sisi klien adalah yang terlihat secara langsung oleh pengguna sehingga kaitannya erat dengan tampilan semisal HTML, CSS dan Javascript. Sisi server merupakan bagian website yang tidak tampil ke pengguna, namun bagian ini lah

yang mengolah logika bisnis dan data yang ditampilkan sehingga membuat website tersebut lebih dinamis.

PHP merupakan bahasa pemrograman yang berjalan di sisi server. Saat ini PHP masih populer dalam dunia pengembangan website karena komunitasnya yang sangat banyak di penjuru dunia. Penggunaan PHP dalam pengembangan web juga tergolong mudah karena sudah banyak tersedia hosting yang terjangkau dengan pengaturan yang familiar.

Mempersiapkan PHP

Sebelum mulai untuk menggunakan PHP kita persiapkan terlebih dahulu environment yang dibutuhkan di antaranya: Apache, MySql, PHP, dan Perl.

Apache adalah web server agar seolah-olah PC/laptop kita menjadi server lokal. Mysql akan digunakan sebagai database, sedangkan PHP adalah bahasa pemrograman yang akan dipakai untuk membangun web, terakhir Perl membantu agar server dapat menjalankan PHP.

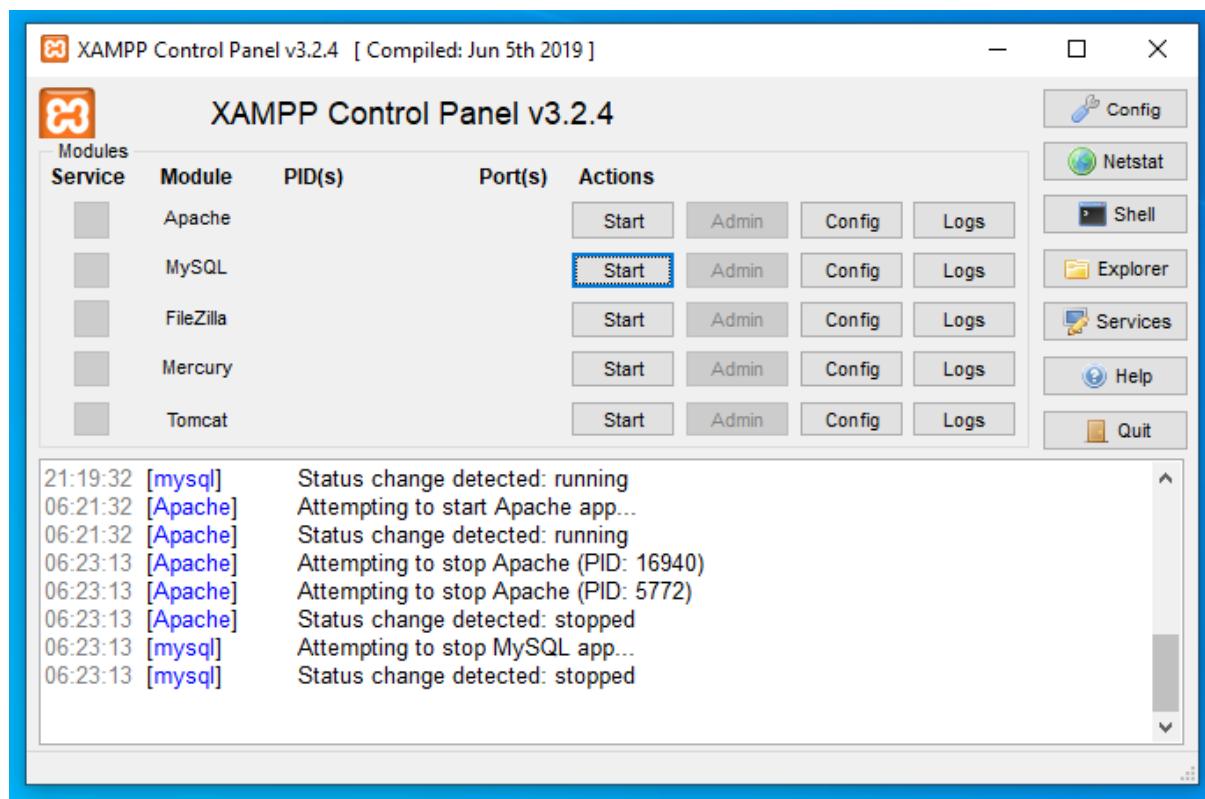
Untuk menginstall semuanya tidak perlu diinstall terpisah satu per satu, kita dapat menginstall aplikasi yang all-in-one seperti XAMPP, WAMPP, LAMPP atau MAMP.

Pilihlah aplikasi yang sesuai dengan Sistem Operasi yang kita pakai:

Windows/MAC : XAMPP (<https://www.apachefriends.org/download.html>)

Menyalakan XAMPP

Ketika sudah selesai menginstall XAMPP, buka aplikasi Xampp lalu akan tampil panel seperti berikut



tampilan XAMPP

nyalakan module **Apache** dan **Mysql** dengan cara menge-klik tombol **start** pada panel. Jika tidak terdapat error, maka kita dapat mengecek ke **web browser** lalu buka alamat url **localhost**. Tampilan localhost pertama kali akan menunjukkan seperti berikut

Welcome to XAMPP for Windows 7.4.6

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQs to learn how to protect your site. Alternatively you can use WAMP, MAMP or LAMP which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

Community

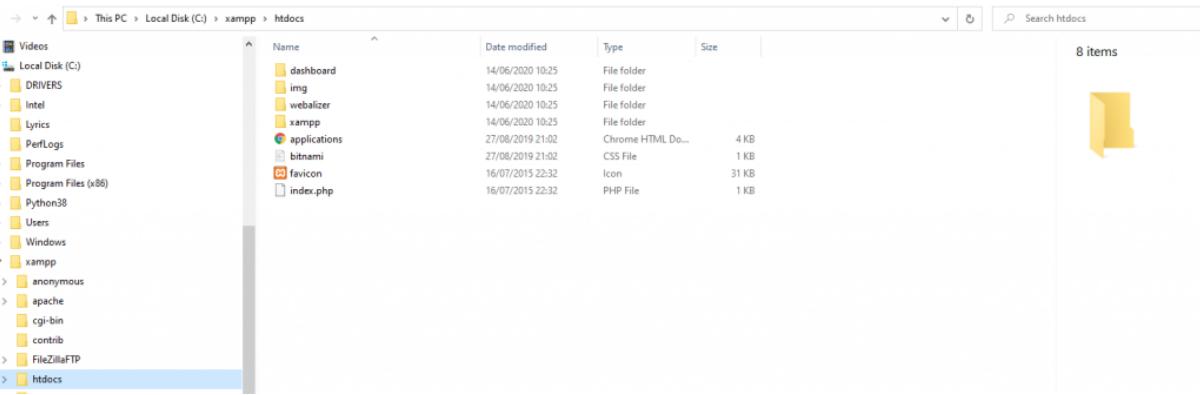
XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.

Contribute to XAMPP translation at translate.apachefriends.org.

Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We

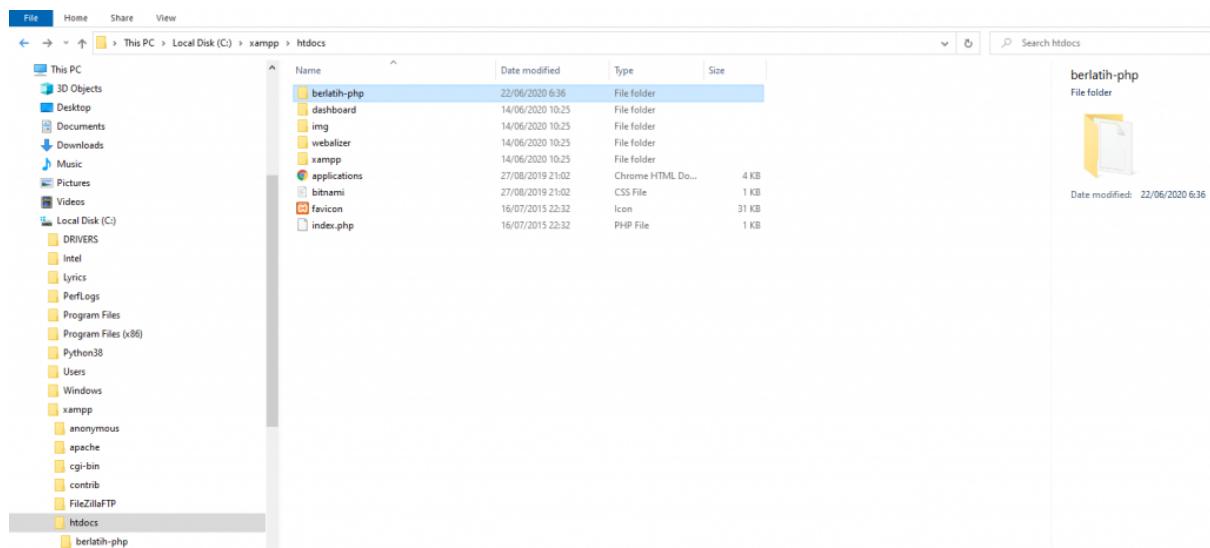
Folder htdocs

Pada folder instalasi XAMPP terdapat folder htdocs. Lokasi nya biasanya dapat kita akses ke C:/xampp/htdocs .



Di dalam folder htdocs ini adalah tempat kita dapat membuat direktori folder yang berisi project-project PHP.

Misalkan kita buat satu folder di dalam htdocs dengan nama **berlatih-php**.



Di dalam folder **berlatih-php**, buatlah satu file **index.php** lalu tulislah code seperti berikut

```
<!DOCTYPE html>
<html>
<head>
<title>Berlatih PHP</title>
</head>
<body>

<?php
$hello = "Hello World";
echo "<h1>" . $hello . "</h1>";
?>

</body>
</html>
```

Hello World

Mulai Menulis PHP

Sintaks

Sintaks PHP ditulis di dalam tag php seperti berikut:

```
<?php
?>
```

<?php sebagai pembuka sintax ?> sebagai penutup sintaxnya

Tag Komentar

Tag Komentar dipakai untuk menandai judul atau bagian dari kode yang sedang ditulis, hal ini membantu penulis kode agar mudah mencari kembali bagian-bagian kode tertentu. Cara menulis komentar yaitu dengan membubuhkan // untuk komentar satu baris, dan /* */ untuk komentar lebih dari satu baris.

```
<!DOCTYPE html>
<html>
<head>
<title>Berlatih PHP</title>
</head>
<body>

<?php
// Komentar satu baris

/*
Komentar lebih dari satu baris
contoh pertama
*/


/***
*komentar lebih dari satu baris
*Contoh kedua
***/
?>

</body>
</html>
```

Variabel dan titik koma ;

Untuk menulis satu variabel di dalam PHP digunakan tanda \$. Aturannya adalah

- tidak boleh menggunakan angka di depan nama variabel tapi boleh menggunakan angka di tempat lain.
- tidak boleh memuat spesial karakter (~!@#\$%. dan lain lain)
- jika variabel terdapat 2 suku kata jangan dipisah dengan spasi ada baiknya menggunakan penulisan camelCase atau snake_case

Setiap baris kode wajib menggunakan semikolon atau titik koma di akhir baris kode.

```
<!DOCTYPE html>
<html>
<head>
<title>Berlatih PHP</title>
</head>
<body>

<?php
// Benar
$name = "Rezky";
$fullName = "Rezky Putra";
$full_name = "Rezky Putra";

//salah
$!name = "Rezky";
$1name = "Rezky";
$full name = "Rezky Putra";
?>

</body>
</html>
```

Output Standar

Kita akan sering menampilkan output dari kode yang kita buat ke halaman browser. Terdapat beberapa pilihan sintaks untuk menampilkan output yaitu echo, print, print_r, dan var_dump.

```
<?php  
$hello = "Hello World!";  
// echo  
echo $hello;  
  
// print  
print $hello;  
  
// print_r  
print_r($hello);  
  
// var_dump  
var_dump($hello);  
  
?>
```

untuk menampilkan output berupa tipe data string biasanya digunakan echo atau print . Sedangkan print_r dapat juga digunakan untuk menampilkan tipe data array. var_dump akan menampilkan output berupa nilai dari yang ingin ditampilkan dilengkapi dengan informasi berupa tipe data dan panjang data.

Data Types PHP

String = tipe data yang berisi karakter-karakter dan biasa ditandai dengan pasangan tanda petik ganda (" ") atau tanda petik tunggal (' ').

Integer = tipe data yang berisi angka

Float (float atau kadang di sebut juga dengan double) = type data yang berisi angka desimal

Boolean = tipe data yang hanya memiliki 2 output yaitu true dan false

Array = type data yang bertujuan untuk mengelompokan data datanya yang di tumpung dalam satu variabel

```
$sapa = "Halo Guys!"; // string bisa menggunakan kutip satu '
$hello = 'Hello World!'; // string bisa menggunakan kutip dua "
$usia = 26; //integer tipe data untuk sebuah angka
$phy = 3.16; //float atau double adalah type data yang menampung
//angka desimal
$isWin = true; // Boolean adalah data type yang hanya memiliki
dua //nilai true/false
$peserta = ["Rezky", 25, true]; //array adalah cara kita untuk
//mengelompokan data
```

String Method

strlen

metode strlen dipakai untuk mengetahui panjang dari sebuah string :

```
<?php
$jargon = "Tetap Santai dan Berkualitas";
// akan menampilkan panjang variabel $jargon yaitu 28 karakter
echo strlen($jargon);

// akan menampilkan panjang string 11
```

```
echo strlen("Halo semua!");  
?>
```

strtoupper

metode strtoupper digunakan untuk membuat sebuah string menjadi ditulis ke dalam karakter kapital.

```
<?php  
$statement = "ini gak marah, cuman caps";  
  
// menampilkanINI GAK MARAH, CUMAN CAPS  
echo strtoupper($statement);  
?>
```

strtolower

metode strtolower kebalikan dari strtoupper yaitu membuat suatu string ditulis menjadi karakter-karakter non-kapital.

```
<?php  
$marah = "INI MARAH";  
  
// menampilkan ini marah  
echo strtolower($marah);  
?>
```

strpos

metode strpos untuk mengatahui karakter tersebut berada di index keberapa.

```
<?php  
$kalimat = "Saya sedang belajar PHP";  
$katayangdicari = "PHP";  
$posisi = strpos($kalimat, $katayangdicari);
```

```
echo $posisi; //20  
?>
```

str_word_count

metode str_word_count berfungsi untuk menghitung jumlah kata pada string

```
<?php  
  
$kalimat = "Selamat Datang Di Sanbercode";  
  
//menampilkan 4 bersarkan jumlah kata pada variabel kalimat  
  
echo str_word_count($kalimat); //4  
?>
```

substr

metode substr untuk mengambil beberapa karakter pada string. menerima 3 parameter (<string yang ingin dipotong>, <indeks mulai potong>, <panjang yang ingin dipotong>).

indeks mulai potong dapat bernilai minus yang berarti mulai indeks dari karakter paling belakang.

```
<?php  
$string = "Saya suka PHP";  
  
// akan menampilkan suka PHP  
echo substr($string, 5, 8);  
  
// akan menampilkan PH  
echo substr($string, -3, 2);  
?>
```

Tipe Data Array

tipe data array adalah tipe data yang mampu menampung lebih dari satu data. Elemen dari array ditandai dengan indeks agar dapat diakses. indeks array selalu diawali dengan indeks 0.

Deklarasi Array/indexed Array

membuat array baru dengan cara seperti berikut:

```
<?php  
$siswa = array("regi", "bobby", "ahmad");  
//Array ( [0] => regi [1] => bobby [2] => ahmad )  
print_r($siswa);  
  
echo "<br>";  
  
$trainer = ["abduh", "agnat", "yoga"];  
//Array ( [0] => abduh [1] => agnat [2] => yoga )  
print_r($trainer);  
  
echo "<br>";  
  
//akses pernilainya di array  
  
echo $trainer[1]; //agnat  
  
echo "<br>";  
  
echo $siswa[2]; //ahmad  
?>
```

ingat! untuk menampilkan array ke output gunakan sintaks print_r.

Mengakses Elemen Array

untuk mengakses elemen pada array kita dapat arahkan dengan menunjuk indeks elemen yang kita inginkan:

```
<?php  
$nilai = [12, 14, 19];  
// akan menampilkan 12 yang merupakan elemen array indeks ke 0  
echo $nilai[0];  
?>
```

Menambahkan elemen baru pada array

terkadang kita ingin menambahkan isi dari array dengan suatu data. caranya adalah dengan
\$array[] = <data yang ingin ditambahkan>

```
<?php  
$siswa = array("regi", "bobby", "ahmad");  
// siswa awal sebelum ditambah  
print_r($siswa);  
$siswa[] = "putri";  
// menampilkan siswa setelah ditambah data baru "putri"  
print_r($siswa);  
?>
```

jika data yang ingin ditambahkan lebih dari satu data, kita bisa gunakan metode
array_push(). berikut ini

```
<?php  
$stack = array("orange", "banana");  
array_push($stack, "apple", "raspberry");
```

```
print_r($stack);  
?>
```

count() untuk menghitung panjang array

metode count digunakan untuk menghitung panjang dari sebuah array.

```
<?php  
$keranjang = ["pisang", "apel", "mangga"];  
// menampilkan panjang array $keranjang  
echo count($keranjang); //3  
?>
```

Array

Array Asosiatif

Array Asosiatif adalah tipe array yang dapat menyimpan banyak data sebagai pasangan key dan value. Berbeda dengan array biasa yang menggunakan angka untuk menandai indeks, array asosiatif menggunakan key sebagai indeks yang dapat kita set sendiri.

penulisan array asosiatif yaitu dengan memasangkan **key** dan **value** dengan pemisah tanda panah => .

Contohnya sebagai berikut:

```
<?php  
  
$siswa1 =  
    [ "nama" => "abduh",  
      "kelas" => "laravel",  
      "nilai" => 70  
    ];
```

```
print_r($siswa1);

// menambahkan key value baru ke array $siswa1
$siswa1["nama"] = "abduh";

print_r($siswa1);
?>
```

Array Multidimensional

Array Multidimensional adalah tipe array yang dapat menyimpan array dalam sebuah array

```
$arrayMulti = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
// Maka sebagai gambaran, indeks dari array tersebut adalah
/*
[
    [(0,0), (0,1), (0,2)],
    [(1,0), (1,1), (1,2)],
    [(2,0), (2,1), (2,2)]
]
*/
```

contohnya sebagai berikut:

```
<?php  
$trainer = array (  
    array("Rezky", "Laravel") ,  
    array("Abduh", "Adonis") ,  
    array("Iqbal", "VueJs") ,  
) ;  
  
echo "<pre>";  
print_r($trainer);  
echo "</pre>";  
?>  
  
//output  
  
Array  
(  
    [0] => Array  
        (  
            [0] => Rezky  
            [1] => Laravel  
        )  
  
    [1] => Array  
        (  
            [0] => Abduh  
            [1] => Adonis  
        )  
  
    [2] => Array  
        (  
            [0] => Iqbal  
            [1] => VueJs  
        )
```

```
)
```

Function PHP

Function dalam PHP adalah sebuah blok kode yang dapat dieksekusi secara terpisah dan dapat dipanggil kapan saja dan di mana saja dalam sebuah program. Function dapat digunakan untuk mengorganisir kode program yang kompleks, meningkatkan efisiensi dalam pengembangan aplikasi, serta memudahkan pengelolaan kode program.

Berikut adalah contoh sintaksis dari sebuah function di PHP:

```
function nama_function($parameter1, $parameter2, ...) {  
    // kode yang akan dijalankan ketika function dipanggil  
}
```

function tanpa parameter

berikut adalah contoh function tanpa parameter

```
<?php  
function baris () {  
    return"===== <br>";  
}  
  
echo baris();  
echo baris();  
echo baris();  
  
//output  
//=====
```

//=====

?>

function dengan parameter

berikut adalah contoh function dengan parameter

```
<?php  
function pengenalanDiri($nama){  
    return "hello perkenalkan nama saya " . $nama . "<br>";  
}  
  
echo pengenalanDiri("rezky");  
echo pengenalanDiri("budi");  
echo pengenalanDiri("ardi");  
  
//output  
//echo hello perkenalkan nama saya rezky  
//echo hello perkenalkan nama saya budi  
//echo hello perkenalkan nama saya ardi  
?>
```

function lebih dari satu parameter

berikut adalah contoh function dengan lebih dari 1 parameter

```
<?php  
function Perkalian($num1, $num2){  
    $kali = $num1 * $num2;  
    return "hasil dari ". $num1 . " dikali " . $num2 . " adalah "  
. $kali . "<br>";  
}  
  
echo Perkalian(2,2);  
echo Perkalian(5,7);  
echo Perkalian(8,9);  
  
//output  
// hasil dari 2 dikali 2 adalah 4
```

```
// hasil dari 5 dikali 7 adalah 35  
// hasil dari 8 dikali 9 adalah 72  
?>
```

Conditional PHP

Conditional atau percabangan pada PHP digunakan untuk mengatur alur program berdasarkan kondisi tertentu atau nama lainnya control flow. Ada beberapa jenis conditional pada PHP, yaitu:

If Statement:

If statement digunakan untuk mengeksekusi kode jika kondisi terpenuhi.

```
<?php  
$usia = 70;  
if ($usia >= 60) {  
    echo "Anda sudah memasuki usia lansia";  
}  
?>  
  
//output  
//Anda sudah memasuki usia lansia
```

Kode di atas akan mengeksekusi echo "Anda sudah memasuki usia lansia" jika usia lebih besar atau sama dengan 60.

If-Else Statement:

If-else statement digunakan untuk mengeksekusi kode jika kondisi terpenuhi, dan kode lain jika kondisi tidak terpenuhi.

```
<?php  
$usia = 50;  
if ($usia >= 60) {  
    echo "Anda sudah memasuki usia lansia";  
} else {  
    echo "Anda belum memasuki usia lansia";  
}
```

```
?>

//output
//Anda belum memasuki usia lamsia
```

Kode di atas akan mengeksekusi echo "Anda sudah memasuki usia lamsia" jika usia lebih besar atau sama dengan 60, dan echo "Anda belum memasuki usia lamsia" jika nilai kurang dari 60.

If-Elseif-Else Statement:

If-elseif-else statement digunakan untuk mengeksekusi kode jika kondisi pertama tidak terpenuhi, kemudian dilanjutkan dengan kondisi kedua jika tidak terpenuhi, dan seterusnya.

```
<?php
$usia = 20;
if ($usia >= 60) {
    echo "Anda sudah memasuki usia lamsia";
} else if($usia >= 26 && $usia < 60){
    echo "Anda sudah memasuki usia Dewasa";
} else if($usia >= 17 && $usia < 26){
    echo "Anda sudah memasuki usia Remaja";
} else {
    echo "Anda belum memasuki usia balita";
}
?>

//output
//Anda sudah memasuki usia Remaja
```

Kode di atas akan mengeksekusi echo "Anda sudah memasuki usia Remaja" karena nilai 20 memenuhi kondisi ketiga $\$usia \geq 17 \text{ && } \$usia < 26$.

Switch Statement:

Switch statement digunakan untuk membandingkan nilai dengan beberapa kondisi, dan mengeksekusi kode yang sesuai dengan kondisi yang cocok.

```
<?php  
$nilai = "A";  
switch ($nilai) {  
    case "A":  
        echo "Anda lulus dengan nilai sangat baik";  
        break;  
    case "B":  
        echo "Anda lulus dengan nilai baik";  
        break;  
    case "C":  
        echo "Anda lulus dengan nilai cukup";  
        break;  
    default:  
        echo "Anda tidak lulus";  
        break;  
}  
//output  
//Anda lulus dengan nilai sangat baik  
?>
```

Kode di atas akan mengeksekusi echo "Anda lulus dengan nilai sangat baik" karena nilai A cocok dengan kondisi pertama. Jika tidak ada kondisi yang cocok, maka kode yang ada di dalam blok default akan dieksekusi "Anda tidak lulus"

Looping PHP

Perulangan atau looping pada PHP dapat dilakukan dengan beberapa cara yang paling umum digunakan, yaitu:

For Loop:

For loop digunakan untuk mengetahui berapa kali Anda perlu mengeksekusi serangkaian instruksi. For loop memiliki tiga bagian: inisialisasi(variabel mulai), kondisi(kondisi berakhirnya perulangan), dan increment/decrement.

```

<?php
for ($i = 1; $i <= 10; $i++) {
    echo "Nilai i adalah: $i <br>";
}
?>

//Output
//Nilai i adalah: 1
//Nilai i adalah: 2
//Nilai i adalah: 3
//Nilai i adalah: 4
//Nilai i adalah: 5
//Nilai i adalah: 6
//Nilai i adalah: 7
//Nilai i adalah: 8
//Nilai i adalah: 9
//Nilai i adalah: 10

```

Loop diatas akan mengeksekusi kode yang di dalamnya sebanyak 10 kali, dimulai dari nilai awal \$i yaitu 1, dan akan diincrement setiap kali loop dijalankan hingga mencapai kondisi \$i tidak lebih kecil lagi dari pada 10.

While Loop:

While loop digunakan ketika kita tidak tahu berapa kali kita perlu mengeksekusi serangkaian instruksi. Loop ini hanya memiliki satu bagian yaitu kondisi.

```

<?php
$i = 1;
while ($i <= 10) {
    echo "Nilai i adalah: $i <br>";
    $i+=2;
}
?>

```

```
//Output  
//Nilai i adalah: 1  
//Nilai i adalah: 3  
//Nilai i adalah: 5  
//Nilai i adalah: 7  
//Nilai i adalah: 9
```

Loop diatas akan mengeksekusi kode yang di dalamnya sebanyak 10 kali, dimulai dari nilai awal \$i yaitu 1, dan akan ditambah 2 setiap kali loop dijalankan hingga mencapai kondisi \$i tidak lebih kecil lagi dari pada 10 maka perulangan di hentikan.

Do-While Loop:

Do-while loop mirip dengan while loop, namun kode di dalamnya akan dieksekusi setidaknya sekali, meskipun kondisinya tidak terpenuhi. Loop ini memiliki satu bagian yaitu kondisi.

```
$i = 0;  
do {  
    echo "Nilai i adalah: $i <br>";  
    $i++;  
} while ($i < 10);  
  
//output  
//Nilai i adalah: 0  
//Nilai i adalah: 1  
//Nilai i adalah: 2  
//Nilai i adalah: 3  
//Nilai i adalah: 4  
//Nilai i adalah: 5  
//Nilai i adalah: 6  
//Nilai i adalah: 7  
//Nilai i adalah: 8  
//Nilai i adalah: 9
```

Loop diatas akan mengeksekusi kode yang di dalamnya setidaknya sekali, meskipun kondisi tidak terpenuhi. \$i akan diincrement setiap kali loop dijalankan hingga mencapai nilai 9.

Foreach Loop:

Foreach loop adalah looping yang dikhususkan untuk looping array

```
<?php
$buah = array("apel", "mangga", "jeruk");
foreach ($buah as $nilai) {
    echo "Saya suka $nilai <br>";
}
?>
//output
//Saya suka apel
//Saya suka mangga
//Saya suka jeruk
```

BAB 6 OOP PHP

Pemrograman berorientasi objek (OOP) adalah paradigma pemrograman yang digunakan dalam pengembangan perangkat lunak modern. OOP memungkinkan pengorganisasian dan struktur kode program berdasarkan objek, yang merupakan instansi dari kelas. PHP adalah bahasa pemrograman skrip server (server side) yang populer dan mendukung OOP.

Di PHP, Anda dapat mendefinisikan kelas menggunakan kata kunci **class**. Kelas adalah cetak biru untuk membuat objek, yang merupakan instansi dari kelas tersebut. Kelas dapat berisi properti, merupakan variabel yang menyimpan data, dan **metode (method)**, merupakan fungsi yang melakukan operasi pada data tersebut.

\$this OOP PHP

\$this dalam OOP PHP merujuk pada objek saat ini yang sedang digunakan. Ini adalah variabel internal yang digunakan untuk mengakses properti dan metode objek yang sedang diproses. Dalam PHP, variabel \$this selalu merujuk pada objek saat ini, dan hanya dapat digunakan di dalam kelas.

Berikut adalah contoh penggunaan \$this dalam OOP PHP:

```
<?php
class Mobil {
    public $merek;
    public $harga;

    public function __construct($merek, $harga) {
        $this->merek = $merek;
        $this->harga = $harga;
    }
}

// membuat objek dari kelas Mobil
```

```

$mobill = new Mobil("Toyota", 200000000);
echo $mobill->merek . "<br>";
echo $mobill->harga . "<br>";

$mobil2 = new Mobil("Honda", 150000000);
echo $mobil2->merek . "<br>";
echo $mobil2->harga . "<br>";

?>

```

Instansiasi Object

Instansiasi objek dalam OOP PHP adalah proses membuat objek dari suatu kelas. Untuk membuat objek dari kelas, kita menggunakan kata kunci new diikuti dengan nama kelas dan argumen yang diperlukan untuk konstruktor kelas.

Berikut adalah contoh penggunaan instansiasi objek dalam OOP PHP:

```

<?php
class Mobil {
    public $merek;
    public $harga;

    public function __construct($merek, $harga) {
        $this->merek = $merek;
        $this->harga = $harga;
    }
}

// membuat objek dari kelas Mobil
$mobill = new Mobil("Toyota", 200000000);
echo $mobill->merek . "<br>";
echo $mobill->harga . "<br>";

```

```
$mobil2 = new Mobil("Honda", 150000000);
echo $mobil2->merek . "<br>";
echo $mobil2->harga . "<br>";

?>
```

Construct

Konstruktor dalam OOP PHP adalah metode khusus yang dipanggil secara otomatis ketika objek dibuat dari suatu kelas. Konstruktor digunakan untuk menginisialisasi nilai properti objek saat objek dibuat.

Dalam PHP, konstruktor didefinisikan dengan metode yang memiliki nama `__construct()`. Ini adalah metode yang dipanggil secara otomatis ketika objek dibuat dari suatu kelas.

Berikut adalah contoh kelas PHP sederhana dengan konstruktor:

```
<?php
class Mobil {
    public $merek;
    public $harga;

    public function __construct($merek, $harga) {
        $this->merek = $merek;
        $this->harga = $harga;
    }
}

// membuat objek dari kelas Mobil
$mobil1 = new Mobil("Toyota", 200000000);
echo $mobil1->merek . "<br>";
echo $mobil1->harga . "<br>";
```

```
$mobil2 = new Mobil("Honda", 150000000);  
echo $mobil2->merek . "<br>";  
echo $mobil2->harga . "<br>";  
  
?>
```

Pada contoh di atas, kita membuat kelas Mobil dengan dua properti yaitu \$merek dan \$harga, konstruktor kelas Mobil menggunakan parameter \$merek dan \$harga untuk menginisialisasi nilai properti \$merek dan \$harga ketika objek dibuat dari kelas.

Setter dan Getter

Setter dan Getter adalah dua metode dalam OOP (Object Oriented Programming) PHP yang digunakan untuk mengatur dan mengambil nilai dari properti dalam kelas.

Setter adalah metode yang digunakan untuk mengatur nilai properti dalam kelas. Dalam setter, kita menggunakan parameter untuk menerima nilai yang akan diatur ke dalam properti. Berikut adalah contoh setter untuk properti nama dalam kelas Karyawan:

```
<?php  
class Karyawan {  
    private $nama;  
    //setter  
    public function setNama($nama) {  
        $this->nama = $nama;  
    }  
}  
?>
```

Pada contoh di atas, kita menggunakan parameter \$nama dalam setter setNama() untuk menerima nilai yang akan diatur ke dalam properti \$nama. Untuk mengatur nilai properti, kita menggunakan sintaks \$this->nama = \$nama;

Getter adalah metode yang digunakan untuk mengambil nilai dari properti dalam kelas. Dalam getter, kita tidak menggunakan parameter, melainkan hanya mengembalikan nilai dari

properti yang diinginkan. Berikut adalah contoh getter untuk properti nama dalam kelas Karyawan:

```
class Karyawan {  
    private $nama;  
  
    //setter  
    public function setNama($nama) {  
        $this->nama = $nama;  
    }  
    //getter  
    public function getNama() {  
        return $this->nama;  
    }  
}
```

Pada contoh di atas, kita menambahkan getter getNama() untuk mengambil nilai dari properti \$nama. Untuk mengambil nilai properti, kita menggunakan sintaks return \$this->nama;.

Dengan menggunakan setter dan getter, kita dapat mengatur dan mengambil nilai dari properti dengan aman dan terkontrol. Setter dan getter juga dapat membantu dalam pengolahan data dan validasi data yang dimasukkan.

Property (properti)

Properti dalam OOP PHP adalah variabel yang didefinisikan di dalam suatu kelas dan digunakan untuk menyimpan nilai yang terkait dengan objek dari kelas tersebut. Properti dapat diakses dan dimodifikasi oleh metode di dalam kelas tersebut.

Anda dapat mendefinisikan properti dengan menggunakan kata kunci public, private, atau protected. Berikut adalah penjelasan singkat tentang ketiga jenis aksesibilitas properti tersebut:

- public: Properti dengan aksesibilitas public dapat diakses dan dimodifikasi dari mana saja, baik dari dalam kelas maupun dari luar kelas.
- private: Properti dengan aksesibilitas private hanya dapat diakses dan dimodifikasi dari dalam kelas itu sendiri. Properti private tidak dapat diakses dari luar kelas.
- protected: Properti dengan aksesibilitas protected hanya dapat diakses dan dimodifikasi dari dalam kelas itu sendiri dan kelas turunan. Properti protected tidak dapat diakses dari luar kelas dan kelas turunan.

Berikut adalah contoh kelas dengan properti public, private, dan protected:

```
<?php

class Mobil {
    public $merek; // properti public
    private $harga; // properti private
    protected $warna; // properti protected

    public function __construct($merek, $harga, $warna) {
        $this->merek = $merek;
        $this->harga = $harga;
        $this->warna = $warna;
    }
}

class MobilSport{
    public function gantiWarna($warna){
        return "wana mobil adalah : " . $this->warna = $warna .
"<br>";
    }
}

$mobil = new Mobil("Toyota", 200000000, "Merah");
echo "Merek mobil: " . $mobil->merek . "<br>";
//echo "Warna mobil: " . $mobil->warna . "<br>"; // error karena
// properti protected
//echo "harga mobil: " . $mobil->harga . "<br>"; // error karena
```

```
properti private

$mobilSport = new MobilSport("Ferrari", 1000000000, "Biru");
echo "Merek mobil: " . $mobilSport->merk . "<br>";
echo $mobilSport->gantiWarna("Merah");
//echo "harga mobil: " . $mobil->harga . "<br>"; // error karena
//properti private

?>
```

Method (metode)

Metode dalam OOP PHP adalah fungsi atau perilaku yang dimiliki oleh sebuah objek. Dalam konsep OOP, objek memiliki data atau properti yang menyimpan informasi dan metode yang melakukan operasi pada data tersebut. Metode sering digunakan untuk mengubah nilai properti atau melakukan tindakan yang berhubungan dengan objek.

Berikut adalah contoh kelas PHP sederhana yang memiliki dua metode:

```
<?php
class Lingkaran {
    public $jari_jari;

    public function hitungLuas() {
        return pi() * pow($this->jari_jari, 2);
    }

    public function hitungKeliling() {
        return 2 * pi() * $this->jari_jari;
    }
}

// membuat objek dari kelas Lingkaran
$lingkaran = new Lingkaran();
```

```
$lingkaran->jari_jari = 5;

// memanggil metode
echo "Luas lingkaran: " . $lingkaran->hitungLuas() . "<br>";
echo "Keliling lingkaran: " . $lingkaran->hitungKeliling() .
"<br>";
?>
```

Pada contoh di atas, kita membuat kelas Lingkaran dengan satu properti yaitu \$jari_jari dan dua metode yaitu hitungLuas() dan hitungKeliling(). Metode hitungLuas() menghitung luas lingkaran berdasarkan nilai \$jari_jari, sedangkan metode hitungKeliling() menghitung keliling lingkaran.

Setelah objek dibuat, kita dapat memanggil metode menggunakan operator ->. Dalam contoh di atas, kita memanggil metode hitungLuas() dan hitungKeliling() pada objek \$lingkaran untuk menghitung luas dan keliling lingkaran.

Harap dicatat bahwa setiap objek yang dibuat dari kelas memiliki salinan metode yang sama, tetapi nilai properti dapat berbeda-beda tergantung pada nilai yang ditetapkan pada saat objek dibuat.

inheritance

Class turunan atau inheritance adalah salah satu konsep penting dalam OOP (Object Oriented Programming) PHP yang memungkinkan kita untuk membuat kelas baru dengan memanfaatkan sifat-sifat (properties) dan perilaku (behavior) dari kelas yang sudah ada (parent class).

Dalam inheritance, kelas yang sudah ada disebut dengan kelas induk atau parent class. Sementara itu, kelas yang baru dibuat dengan mewarisi sifat-sifat dan perilaku dari kelas induk disebut dengan kelas anak atau child class.

Untuk membuat class turunan di PHP, kita dapat menggunakan kata kunci extends diikuti dengan nama kelas induk yang ingin diwarisi. Setelah itu, kita dapat menambahkan properti dan metode baru pada kelas anak.

Berikut adalah contoh penggunaan inheritance dalam OOP PHP:

```
<?php  
class Kendaraan {  
    protected $merek;  
  
    public function __construct($merek) {  
        $this->merek = $merek;  
    }  
  
    public function getMerek() {  
        return $this->merek;  
    }  
}  
  
class Mobil extends Kendaraan {  
    private $harga;  
  
    public function __construct($merek, $harga) {  
        parent::__construct($merek);  
        $this->harga = $harga;  
    }  
  
    public function getInfo() {  
        return "Merek: " . $this->getMerek() . ", Harga: " .  
$this->harga;  
    }  
}  
  
// membuat objek dari kelas Mobil  
$mobil = new Mobil("Toyota", 20000000);
```

```
// memanggil metode  
echo $mobil->getInfo();  
?>
```

Pada contoh di atas, kita memiliki kelas Kendaraan sebagai kelas induk dengan satu properti yaitu \$merek dan satu metode yaitu getMerek(). Selain itu, kita juga memiliki kelas Mobil sebagai kelas anak dengan tambahan satu properti yaitu \$harga dan satu metode yaitu getInfo().

Di dalam kelas Mobil, kita menggunakan kata kunci extends untuk mewarisi properti dan metode dari kelas Kendaraan. Kita juga menggunakan metode parent::__construct() untuk memanggil konstruktor kelas Kendaraan dan menginisialisasi nilai properti \$merek.

Setelah objek dibuat, kita dapat mengakses metode getInfo() dari objek \$mobil. Di dalam metode getInfo(), kita menggunakan metode getMerek() untuk mengembalikan nilai properti \$merek dari objek \$mobil.

BAB 7 SQL Database

database

Pangkalan data atau basis data (bahasa Inggris: database) adalah kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer.

database SQL dan non SQL

SQL (Structured Query Language) dan NoSQL (Not Only SQL) adalah dua jenis basis data yang berbeda. Berikut adalah perbedaan antara SQL dan NoSQL:

- Model Data: SQL database berdasarkan pada model data relasional, sedangkan NoSQL database bisa menggunakan model data yang berbeda seperti dokumen, grafik, atau key-value.

- Skema: SQL database menggunakan skema yang didefinisikan dengan ketat dan harus diikuti, sedangkan NoSQL database lebih fleksibel dalam hal skema.
- Bahasa: SQL database menggunakan bahasa SQL untuk mengakses dan memanipulasi data, sedangkan NoSQL database memiliki bahasa yang berbeda, tergantung pada model data yang digunakan.
- Scalability: SQL database biasanya lebih sulit untuk di-skala horizontal, sedangkan NoSQL database dapat di-skala horizontal dengan mudah.
- Data Integrity: SQL database memiliki dukungan yang kuat untuk menjaga integritas data dan menjaga konsistensi antara tabel yang berbeda, sedangkan NoSQL database tidak memiliki dukungan yang sama.

Pilihan SQL atau NoSQL tergantung pada jenis aplikasi dan kebutuhan bisnis. SQL database cocok untuk aplikasi dengan skema data yang konsisten dan kompleks, sedangkan NoSQL database cocok untuk aplikasi dengan data yang beragam dan skema data yang fleksibel.

Database SQL

disini kita fokus ke database SQL

SQL database atau RDBMS (Relational Database Management System) adalah sistem manajemen basis data yang didasarkan pada model data relasional. SQL sendiri adalah singkatan dari Structured Query Language, yang merupakan bahasa yang digunakan untuk mengakses dan memanipulasi data dalam basis data relasional.

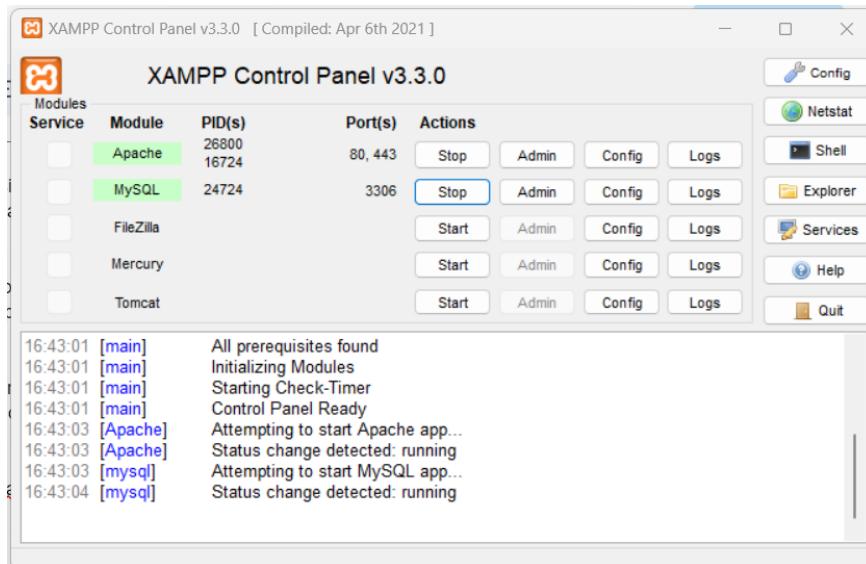
Contoh SQL database yang umum digunakan adalah:

- MySQL: MySQL adalah salah satu sistem basis data open-source yang paling populer di dunia. MySQL mendukung banyak platform dan dapat digunakan dalam aplikasi web, aplikasi desktop, dan banyak lagi.

- Microsoft SQL Server: Microsoft SQL Server adalah sistem basis data relasional yang dikembangkan oleh Microsoft. SQL Server digunakan terutama di lingkungan korporat dan perusahaan.
- Oracle Database: Oracle Database adalah sistem basis data relasional yang dikembangkan oleh Oracle Corporation. Oracle Database memiliki fitur yang sangat kuat dan banyak digunakan dalam lingkungan bisnis.
- PostgreSQL: PostgreSQL adalah sistem basis data open-source yang kuat dan sering digunakan dalam aplikasi web dan lingkungan pengembangan perangkat lunak.
- SQLite: SQLite adalah sistem basis data relasional yang ringan dan mudah digunakan. SQLite banyak digunakan di lingkungan mobile dan aplikasi desktop.

Perangkat yang digunakan

disini kita akan belajar sintax-sintax dasar dari SQL menggunakan Aplikasi XAMPP silahkan jalankan apache dan mysql di XAMPP nya



kemudian buka localhost/phpmyadmin di web browsernya

The screenshot shows the phpMyAdmin interface for a MySQL server at 127.0.0.1. The left sidebar lists databases: Baru, batch35, batch36, batch37, batch38, batch39, batch40, batch41, batch42, blogapi, blogdemo, blogs, contact-app, contohsoal41, contohuud, crowdfunding, crowdfunding43, crowdfunding37, and crowdfunding40. The main panel has three tabs: Pengaturan umum, Pengaturan tampilan, and Server basis data. The Server basis data tab displays server information: Server: 127.0.0.1 via TCP/IP, Jenis server: MariaDB, Koneksi server: SSL sedang tidak digunakan, Versi server: 10.4.24-MariaDB - mariadb.org binary distribution, Versi protokol: 10, Pengguna: root@localhost, and Kumpulan karakter server: UTF-8 Unicode (utf8mb4). The bottom right corner shows the phpMyAdmin logo.

untuk melakukan testing sintax sql nya bisa masuk ke tab SQL

The screenshot shows the phpMyAdmin interface with the SQL tab selected. The top navigation bar includes tabs for Basis data, SQL, Status, Akun pengguna, Eksport, Impor, Pengaturan, Variabel, Set Karakter, Mesin, and Plugin. The main area contains a text input field labeled "Run SQL query/queries on server "127.0.0.1": " followed by a large text area containing the SQL command "1 CREATE DATABASE test123;". Below the text area are buttons for Bersihkan, Format, Get auto-saved query, and Bind parameters. At the bottom, there are checkboxes for Pembatas, Tampilkan ulang perintah SQL, Menahan kotak kueri, Rollback when finished, and Perbolehkan cek forign key, with the "Perbolehkan cek forign key" checkbox checked and circled in red.

untuk menjalankan sintax bisa klik “kirim” atau “sent”

The screenshot shows the SQL query editor with the SQL tab selected. The main area contains the SQL command "1 CREATE DATABASE test123;". At the bottom, there are several buttons: Bersihkan, Format, Get auto-saved query, Bind parameters, and a large "Kirim" button. Below these buttons are checkboxes for Pembatas, Tampilkan ulang perintah SQL, Menahan kotak kueri, Rollback when finished, and Perbolehkan cek forign key. The "Perbolehkan cek forign key" checkbox is checked and circled in red.

Sintax Database

disini kita akan mencoba beberapa sintax seperti membuat Database & menghapus Database

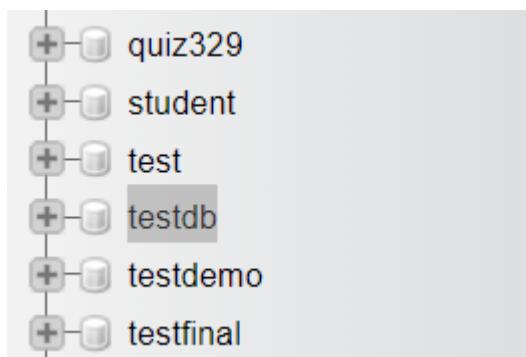
buat Database

```
CREATE DATABASE databasename;
```

contoh:

```
CREATE DATABASE testdb;
```

maka akan terbuat di bagian kiri sidebar phpmyadmin



menghapus database

```
DROP DATABASE databasename;
```

contoh:

```
DROP DATABASE testdb;
```

Table SQL

Dalam database relasional, sebuah tabel adalah struktur data yang digunakan untuk menyimpan dan mengorganisir data dalam kolom dan baris. Setiap kolom dalam tabel mewakili atribut atau sifat data tertentu, sementara setiap baris mewakili sebuah rekaman atau instance data dalam tabel tersebut.

create table

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ...
);
```

disini kita akan membuat 2 table dengan nama buku dan genre
pertama-tama kita harus buat dulu database nya kita sebut saja nama database perpus

```
CREATE DATABASE perpus;
```

jangan lupa klik database yang sudah dibuat di sidebar kiri di phpmyadmin untuk masuk ke databasenya

setelah itu kita akan membuat table genre

```
CREATE TABLE genre(
    id int(8) AUTO_INCREMENT PRIMARY KEY,
    nama varchar(255)
);
```

code diatas kita menggunakan primary key yang fungsinya adalah sebuah kolom dari beberapa kolom dalam tabel yang digunakan untuk mengidentifikasi secara UNIQUE setiap

data dalam tabel tersebut. Kolom primary key memastikan bahwa setiap data dalam tabel memiliki nilai yang UNIQUE atau tidak sama dengan data lainnya.

Selain itu, kita juga dapat menggunakan fitur AUTO_INCREMENT untuk menghasilkan nilai primary key yaitu number UNIQUE yang ter-generate secara automatis untuk setiap data baru yang ditambahkan ke dalam tabel.

code diatas kita juga menggunakan type data varchar dan int berikut adalah link untuk data tipe di SQL

https://www.w3schools.com/sql/sql_datatypes.asp

The screenshot shows the MySQL Workbench interface with the 'Struktur' (Structure) tab selected. Below it, the table 'buku' is displayed with two columns: 'id' (int(11)) and 'nama' (varchar(255)). The 'id' column is defined as AUTO_INCREMENT. Action buttons for 'Ubah' (Edit), 'Hapus' (Delete), and 'Lainnya' (More) are visible for each row.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(255)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

kemudian kita akan membuat table buku

```
CREATE TABLE buku (
    id int(8) AUTO_INCREMENT PRIMARY KEY,
    judul varchar(255),
    ringkasan text,
    genre_id int(8),
    FOREIGN KEY (genre_id) REFERENCES genre(id)
)
```

code diatas kita menggunakan foreign key yang fungsi nya adalah sebuah kolom dalam sebuah tabel yang mengacu pada primary key di tabel lain (table genre kolom id). Dalam hubungan antar-tabel dalam basis data relasional, foreign key digunakan untuk menghubungkan dua atau lebih tabel bersama-sama

Dalam sebuah tabel, foreign key mewakili relasi antara dua tabel, yaitu tabel yang memiliki kolom foreign key (table buku kolom genre_id) dan tabel yang memiliki primary key (table

genre kolom id) yang diacu oleh foreign key tersebut. Kolom foreign key biasanya didefinisikan sebagai **tipe data yang sama dengan primary key** dari tabel yang dihubungkan.

Dalam contoh di atas, tabel **buku** memiliki sebuah kolom foreign key yang mengacu pada kolom **id** di tabel **genre**. Dengan menambahkan FOREIGN KEY pada definisi tabel **buku**, kita menetapkan hubungan antara kedua tabel tersebut, sehingga setiap nilai dalam kolom **genre_id** di tabel **buku** harus memiliki nilai yang sesuai dengan kolom **id** di tabel **genre**.

Dengan adanya foreign key, kita dapat melakukan operasi JOIN antara kedua tabel

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id 🔑	int(11)		Tidak	Tidak ada			AUTO_INCREMENT	Ubah Hapus Lainnya
2	judul	varchar(255)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	ringkasan	text	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
4	genre_id 🔑	int(11)		Tidak	Tidak ada				Ubah Hapus Lainnya

Pilih Semua Dengan pilihan: Jelajahi Ubah Hapus Utama Unik Indeks Spasial Teks penuh

untuk melihat relasinya bisa klik tampilkan hubungan

Actions	Constraint properties	Kolom	Foreign key constraint (INNODB)
Hapus	buku_ibfk_1 ON DELETE RESTRICT ON UPDATE RESTRICT	genre_id	Basis data Tabel Kolom perpus genre id

+ Tambah kolom

Insert Into

insert into adalah perintah untuk memasukan data baru kedalam table

sintax Insert Into

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

disini kita akan memasukan beberapa data dari 2 table yang sudah kita buat yaitu table buku dan genre pertama kita akan memasukan data pada table buku

```
INSERT INTO genre(nama) VALUES ("novel"), ("komik"), ("Manga");
```

code diatas kita hanya memasukan data di kolom nama di table genre karena table id sudah automatis di buat dari fungsi AUTO_INCREMENT dan code diatas kita langsung membuat 3 data yaitu novel, komik dan manga di table genre

The screenshot shows the phpMyAdmin interface with the following details:

- Top navigation bar: Jelajahi, Struktur, SQL, Cari.
- Success message: "Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan)".
- SQL query: "SELECT * FROM `genre`".
- Table results:

	id	nama
<input type="checkbox"/>	1	novel
<input type="checkbox"/>	2	Komik
<input type="checkbox"/>	3	Manga
- Extra options: Tampilkan semua (checkbox), Jumlah baris: 25.
- Bottom buttons: Extra options, Import, Export, Refresh.

setelah itu kita akan melakukan insert data pada table buku

```
INSERT INTO buku(judul, ringkasan, genre_id)
VALUES ("Laskar Pelangi", "Ringkasan novel laskar pelangi", 1),
("One Piece", "Ringkasan Manga One Piece" , 3), ("Naruto
Shipudden", "Ringkasan Manga Naruto", 3);
```

dari code diatas kita hanya bisa memasukan nilai 1, 2 , dan 3 di kolom genre_id karena data yang ada di table genre hanya id 1 untuk novel id 2 untuk komik dan id 3 untuk manga dimana jika kita memasukan data yang belum ada di table genre contoh di kolom genre_id kita masukan 4 maka akan terjadi error di foreign keynya

The screenshot shows a MySQL database interface with the following details:

- Toolbar: Jelajahi, Struktur, SQL, Cari, Tambahkan, Eksport, Import, Hak Akses.
- Status Bar: Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan dalam 0,0005 detik.)
- SQL Query: SELECT * FROM `buku`
- Buttons: Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]
- Table Filter: Tampilkan semua (checkbox), Jumlah baris: 25, Saring baris: Cari di tabel ini, Sort by key: Tidak ada.
- Table Headers: id, judul, ringkasan, genre_id.
- Table Data:

	Ubah	Salin	Hapus	id	judul	ringkasan	genre_id
<input type="checkbox"/>				1	Laskar pelangi	Ringkasan novel laskar pelangi	1
<input type="checkbox"/>				2	One Piece	Ringkasan Manga One Piece	3
<input type="checkbox"/>				3	Naruto Shipudden	Ringkasan Manga Naruto	3

Select

select adalah fungsi SQL untuk memilih data yang akan di tampilkan di databasenya

cara kita untuk semua data yang akan ditampilkan

```
SELECT * FROM table_name;
```

contoh disini kita akan menampilkan/memilih semua data yang akan ditampilkan di table buku

```
SELECT * FROM buku;
```

The screenshot shows a MySQL query results page. At the top, there is a toolbar with various icons: Jelajahi, Struktur, SQL, Cari, Tambahkan, Eksport, Import, and Hak Akses. Below the toolbar, a green status bar displays: "Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan dalam 0,0003 detik.)". The main area contains the SQL query: "SELECT * FROM `buku`". Below the query, there are several buttons: Profil, Edit dikotak, Ubah, Jelaskan SQL, Buat kode PHP, and Segarkan. Further down are filters: "Tampilkan semua", "Jumlah baris: 25", "Saring baris: Cari di tabel ini", "Sort by key: Tidak ada", and "Extra options". The results table has columns: id, judul, ringkasan, genre_id. The data is as follows:

	id	judul	ringkasan	genre_id
<input type="checkbox"/>	1	Laskar pelangi	Ringkasan novel laskar pelangi	1
<input type="checkbox"/>	2	One Piece	Ringkasan Manga One Piece	3
<input type="checkbox"/>	3	Naruto Shipudden	Ringkasan Manga Naruto	3

sintax jika kita hanya ingin menampilkan beberapa kolom saja di table yang kita pilih

```
SELECT column1, column2, ...  
FROM table_name;
```

contoh disini kita akan menampilkan kolom id dan judul saja di table buku

```
SELECT id, judul from buku;
```

✓ Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan dalam 0,0004 detik.)

```
SELECT id, judul from buku;
```

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

Tampilkan semua | Jumlah baris: 25 ▾ Saring baris: Cari di tabel ini

Extra options

	Ubah	Salin	Hapus	id	judul
<input type="checkbox"/>				1	Laskar pelangi
<input type="checkbox"/>				2	One Piece
<input type="checkbox"/>				3	Naruto Shipudden

sintax untuk memiliki data berdasarkan kondisi

```
SELECT *
FROM table_name
WHERE condition;
```

untuk contoh disini kita akan menampilkan semua data di tabel buku dimana genre_id lebih besar dari pada 2

```
SELECT * FROM buku WHERE genre_id > 2;
```

✓ Menampilkan baris 0 - 1 (total 2, Pencarian dilakukan dalam 0,0005 detik.)

```
SELECT * FROM buku WHERE genre_id > 2;
```

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

Tampilkan semua | Jumlah baris: 25 ▾ Saring baris: Cari di tabel ini Sort by key: Tidak ada

Extra options

	Ubah	Salin	Hapus	id	judul	ringkasan	genre_id
<input type="checkbox"/>				2	One Piece	Ringkasan Manga One Piece	3
<input type="checkbox"/>				3	Naruto Shipudden	Ringkasan Manga Naruto	3

dari code diatas kita juga bisa menggunakan kondisi = , < , > , >= , <=

disini kita juga dapat memilih data berdasarkan berdasarkan karakter data pada kolom tertentu menggunakan like

LIKE adalah sebuah operator pada SQL yang digunakan untuk mencari pola atau kata kunci tertentu dalam sebuah kolom teks pada sebuah tabel. Operator LIKE sangat berguna ketika mencari data yang mirip atau sebagian cocok dengan kata kunci tertentu.

berikut adalah sintaxnya

```
SELECT * FROM table_name
WHERE column_name LIKE pattern;
```

beberapa kondisi like

Like Operator	Deskripsi
WHERE column_name LIKE 'a%'	mencari semua data yang memiliki awalan karakter a di datanya
WHERE column_name LIKE '%a'	mencari semua data yang memiliki akhiran karakter a di datanya

WHERE column_name LIKE '%or%'	mencari semua data yang memiliki karakter or di posisi manapun di datanya
WHERE column_name LIKE '_r%'	mencari data yang memiliki karakter r di posisi kedua di datanya

untuk contoh kita akan mencari karakter “Shi” di posisi manapun di data yang berada di kolom genre_id table buku

```
SELECT * FROM buku where judul like "%shi%";
```

✓ Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0,0003 detik.)

```
SELECT * FROM buku where judul like "%shi%";
```

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

Tampilkan semua | Jumlah baris: 25 ▾ Saring baris: Cari di tabel ini

Extra options

← →	id	judul	ringkasan	genre_id
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	3	Naruto Shipudden	Ringkasan Manga Naruto	3

Order By

ORDER BY adalah sebuah pernyataan pada SQL yang digunakan untuk mengurutkan hasil query berdasarkan kolom tertentu dalam tabel. Urutan pengurutan bisa diberikan secara ascending (naik) atau descending (turun).

Berikut adalah sintaks dasar untuk menggunakan ORDER BY pada SQL:

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

untuk contoh order kita akan ngurutkan secara ascending colom judul table buku

```
SELECT * FROM buku order by judul ASC;
```

✓ Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan dalam 0,0003 detik.) [judul: LASKAR PELANGI... - ONE PIECE...]

```
SELECT * FROM buku order by judul ASC;
```

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

Tampilkan semua

Jumlah baris:

25 ▾

Saring baris:

Cari di tabel ini

Sort by key:

Tida

Extra options

	Ubah	Salin	Hapus	id	judul	ringkasan	genre_id
<input type="checkbox"/>				1	Laskar pelangi	Ringkasan novel laskar pelangi	1
<input type="checkbox"/>				3	Naruto Shipudden	Ringkasan Manga Naruto	3
<input type="checkbox"/>				2	One Piece	Ringkasan Manga One Piece	3

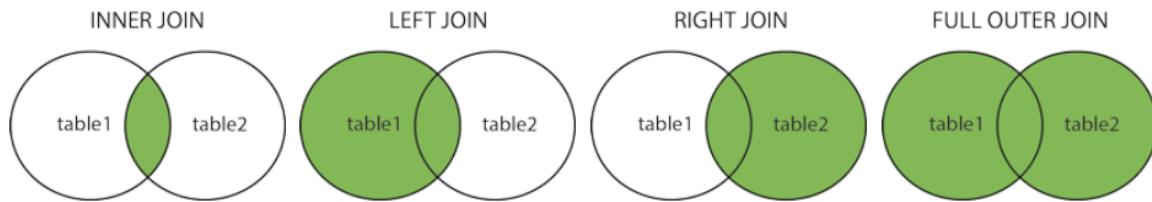
Join

JOIN pada SQL digunakan untuk menggabungkan dua tabel atau lebih dalam satu query berdasarkan kolom-kolom yang memiliki nilai sama. JOIN memungkinkan kita untuk menampilkan data dari tabel yang berbeda secara bersamaan dalam satu query.

Berikut adalah jenis-jenis JOIN pada SQL:

- **INNER JOIN:** menghasilkan baris hanya jika terdapat nilai yang cocok di kedua tabel yang di-join.
- **LEFT JOIN:** menghasilkan semua baris dari tabel kiri (pertama) dan baris yang cocok dari tabel kanan (kedua). Jika tidak ada nilai yang cocok di tabel kanan, maka kolom-kolom tabel kanan akan diisi dengan nilai NULL.
- **RIGHT JOIN:** menghasilkan semua baris dari tabel kanan (kedua) dan baris yang cocok dari tabel kiri (pertama). Jika tidak ada nilai yang cocok di tabel kiri, maka kolom-kolom tabel kiri akan diisi dengan nilai NULL.

- **FULL OUTER JOIN:** menghasilkan semua baris dari kedua tabel yang di-join, kolom-kolom tabel yang tidak memiliki nilai cocok akan diisi dengan nilai NULL.



Berikut adalah sintaks INNER JOIN pada SQL:

```
SELECT column1, column2, ...
FROM table1
JOIN table2
ON table1.column_name = table2.column_name;
```

untuk contohnya kita akan coba menggabungkan table buku dan table genre dan di bandingkan berdasarkan relasi foreign keynya

```
SELECT buku.*, genre.nama as genre FROM buku INNER JOIN genre ON
buku.genre_id = genre.id;
```

dari code diatas kita menampil semua kolom pada tabel buku (*) dan juga kita menampilkan hanya kolom nama di tabel genre kemudian kita gabung dimana table pertama itu buku dan table kedua adalah genre dengan perbandigan berdasarkan foreign dari table buku kolom genre_id sama dengan table genre kolom id

code diatas kita juga menggunakan “as” untuk mengubah nama kolom dari genre yang tadinya nama menjadi genre

Menampilkan baris 0 - 2 (total 3, Pencarian dilakukan dalam 0,0004 detik.)

```
SELECT buku.*, genre.nama as genre FROM buku INNER JOIN genre ON buku.genre_id = genre.id;
```

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

Tampilkan semua | Jumlah baris: 25 ▾ Saring baris: Cari di tabel ini Sort by key: Tidak ada ▾

Extra options

id	judul	ringkasan	genre_id	genre
1	Laskar pelangi	Ringkasan novel laskar pelangi	1	novel
2	One Piece	Ringkasan Manga One Piece	3	Manga
3	Naruto Shipudden	Ringkasan Manga Naruto	3	Manga

Update

Digunakan untuk memperbarui data yang sudah ada dalam tabel.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

untuk contoh sintax update kita akan mengubah data “Naruto Shipudden” menjadi “Jujutsu Kaisen” dari kolom judul table buku

```
UPDATE `buku` SET `judul`='Jujutsu Kaisen' WHERE id = 3;
```

← → ▾

	Ubah	Salin	Hapus	id	judul	ringkasan	genre_id
<input type="checkbox"/>				1	Laskar pelangi	Ringkasan novel laskar pelangi	1
<input type="checkbox"/>				2	One Piece	Ringkasan Manga One Piece	3
<input type="checkbox"/>				3	Jujutsu Kaisen	Ringkasan Manga Naruto	3

Delete

digunakan untuk menghapus data berdasarkan kondisi

```
DELETE FROM table_name WHERE condition;
```

untuk contoh sintax di atas kita akan menghapus data dengan kondisi id = 3 yaitu data jujutsu kaisen

```
DELETE FROM `buku` WHERE id = 3;
```

	Ubah	Salin	Hapus	id	judul	ringkasan	genre_id
<input type="checkbox"/>				1	Laskar pelangi	Ringkasan novel laskar pelangi	1
<input type="checkbox"/>				2	One Piece	Ringkasan Manga One Piece	3

Min, Max, Sum, Avg

Max

Berfungsi nilai Terbesar dari kolom table yang kita ingin menampilkan

berikut sintax nya

```
SELECT MAX(column_name)
FROM table_name
```

table : penjualan

id	name	price
1	Permen	500
2	astor	1000
3	nasi kuning	5000
4	nasi uduk	6000
6	snack	2000
7	minuman	1000

dari table diatas kita dapat melihat mana yang memiliki harga terbesar

```
SELECT MAX(price) AS penjualan_termahal  
FROM penjualan;
```

output

penjualan_termahal
6000

Min

Berfungsi nilai Terkecil dari kolom table yang kita ingin menampilkan

berikut sintaxnya

```
SELECT MIN(column_name)  
FROM table_name
```

table : penjualan

id	name	price
1	Permen	500
2	astor	1000
3	nasi kuning	5000
4	nasi uduk	6000
6	snack	2000
7	minuman	1000

dari table diatas kita dapat melihat mana yang memiliki harga terkecil

```
SELECT MIN(price) AS penjualan_termurah  
FROM penjualan;
```

output

penjualan_termurah
500

SUM

SUM dalam SQL adalah fungsi agregat yang digunakan untuk menghitung jumlah nilai total dari sebuah kolom dalam sebuah tabel.

berikut sintaxnya:

```
SELECT SUM(column_name) FROM table_name;
```

table : penjualan

id	name	price
1	Permen	500
2	astor	1000
3	nasi kuning	5000
4	nasi uduk	6000
6	snack	2000
7	minuman	1000

dari table diatas kita dapat melihat total harga jika semua kolom price dijumlahkan

```
SELECT SUM(price) as total_penjualan FROM penjualan;
```

output

```
total_penjualan
```

```
15500
```

AVG

AVG dalam SQL adalah fungsi agregat yang digunakan untuk menghitung rata-rata nilai dari sebuah kolom dalam sebuah tabel

berikut sintaxnya:

```
SELECT AVG(column_name) FROM table_name;
```

table : penjualan

id	name	price
1	Permen	500
2	astor	1000
3	nasi kuning	5000
4	nasi uduk	6000
6	snack	2000
7	minuman	1000

dari table diatas kita dapat melihat total harga jika semua kolom price dijumlahkan

```
SELECT AVG(price) as rata_rata_penjualan;
```

output

rata_rata_penjualan
2583,333333

BAB 8 Command Line + Git

Command Line

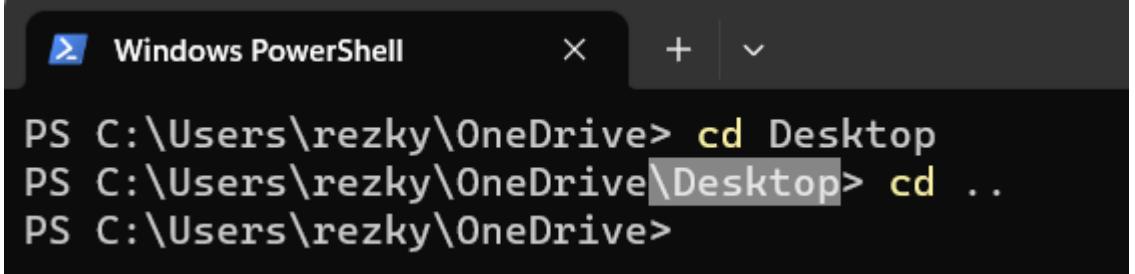
Apa itu command line Interface?

Command line Interface (CLI) atau yang biasa kita kenal cmd atau command prompt atau terminal adalah software atau antar muka untuk memberikan perintah-perintah dasar pada komputer kita. Biasanya kita mengoperasikan komputer seperti browsing media, mencari file, membuat file atau folder, dan lain-lain dengan GUI (Graphical User Interface) yang sudah disediakan pengembang Sistem Operasi seperti Windows, Linux, atau OSX. Namun pada dasarnya kita juga bisa melakukan operasi-operasi tersebut dalam suatu perintah yang kita kirimkan melalui cmd/terminal. Berikut ini beberapa contohnya:

Menuju folder/direktori tertentu

Untuk menuju posisi folder atau direktori tertentu bisa gunakan perintah cd atau kepanjangan dari change directory

```
// Menuju Desktop  
cd Desktop  
// Menuju ke direktori parent (sebelumnya)  
cd ..  
// Mengetahui alamat dari direktori yang sedang dibuka
```



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command history at the bottom of the window shows three commands being run:

```
PS C:\Users\rezky\OneDrive> cd Desktop  
PS C:\Users\rezky\OneDrive\Desktop> cd ..  
PS C:\Users\rezky\OneDrive>
```

The "Desktop" directory is highlighted in the command history.

Terkadang kita butuh tahu sedang dimana posisi terminal kita berjalan. Perintah untuk fungsi ini yaitu PWD atau kependekan dari path of working directory

```
pwd  
/Users/rezky/OneDrive  
// sedang berada di folder rezky/OneDrive  
Mengetahui daftar isi dari sebuah direktori/folder
```

```
PS C:\Users\rezky\OneDrive> pwd  
  
Path  
----  
C:\Users\rezky\OneDrive
```

Melihat daftar file dan folder pada direktori saat ini bisa gunakan perintah ls atau dir . contohnya sebagai berikut

```
$ ls  
Document Desktop Image index.html  
$ dir  
Document Desktop Image index.html
```

Mode	LastWriteTime	Length	Name
-a---l	7/9/2022 1:29 PM	1040	Adobe Animate 2022.lnk
-a---l	7/9/2022 1:29 PM	1084	Adobe Audition 2022.lnk
-a---l	7/9/2022 1:29 PM	1344	Adobe Character Animator 2022.lnk
-a---l	7/9/2022 12:59 PM	1093	Adobe Lightroom Classic.lnk
-a---l	12/24/2022 1:52 PM	1241	Adobe Photoshop 2022.lnk
-a---l	7/9/2022 1:28 PM	1132	Adobe Premiere Pro 2022.lnk
-a---l	9/10/2022 10:10 AM	222	Batman™ Arkham Knight.url
-a---l	7/2/2022 9:52 AM	1086	Internet Download Manager.lnk
-a---l	3/9/2023 10:29 AM	2193	Postman.lnk
-a---l	6/23/2022 3:50 PM	2398	rezky - Chrome.lnk
-a---l	7/19/2022 2:51 PM	1040	Telegram.lnk
-a---l	2/27/2023 12:03 PM	1108	TopTracker.lnk
-a---l	2/9/2023 10:44 AM	1408	Visual Studio Code.lnk
-a---l	1/12/2023 7:22 AM	1939	Zoom.lnk

Perintahnya adalah mkdir atau singkatan dari make directory. contohnya sebagai berikut

```
// membuat folder baru bernama new-folder
$ mkdir new-folder
// mengecek apakah sudah berhasil buat folder baru
$ ls
new-folder . . .
```

```

PS C:\Users\rezky\OneDrive\Desktop> mkdir new-folder

Directory: C:\Users\rezky\OneDrive\Desktop

Mode          LastWriteTime    Length Name
----          -----          ---- 
d---          3/15/2023 11:19 AM      new-folder

PS C:\Users\rezky\OneDrive\Desktop> ls

Directory: C:\Users\rezky\OneDrive\Desktop

Mode          LastWriteTime    Length Name
----          -----          ---- 
d---          3/15/2023 11:19 AM      new-folder
-a--l          7/9/2022  1:29 PM     1040 Adobe Animate 2022.lnk
-a--l          7/9/2022  1:29 PM     1084 Adobe Audition 2022.lnk
-a--l          7/9/2022  1:29 PM     1344 Adobe Character Animator 2022.lnk
-a--l          7/9/2022  12:59 PM    1093 Adobe Lightroom Classic.lnk
-a--l          12/24/2022 1:52 PM     1241 Adobe Photoshop 2022.lnk
-a--l          7/9/2022  1:28 PM     1132 Adobe Premiere Pro 2022.lnk
-a--l          9/10/2022 10:10 AM    222 Batman™ Arkham Knight.url
-a--l          7/2/2022   9:52 AM     1086 Internet Download Manager.lnk
-a--l          3/9/2023  10:29 AM     2193 Postman.lnk
-a--l          6/23/2022 3:50 PM     2398 rezky - Chrome.lnk
-a--l          7/19/2022 2:51 PM     1040 Telegram.lnk
-a--l          2/27/2023 12:03 PM    1108 TopTracker.lnk
-a--l          2/9/2023  10:44 AM    1408 Visual Studio Code.lnk
-a--l          1/12/2023 7:22 AM     1939 Zoom.lnk

```

Software terminal / command line yang direkomendasikan

Untuk OS Windows, terminal bawaannya adalah command prompt atau cmd yang relatif kurang nyaman untuk dipakai dalam tahap development. Oleh karena itu direkomendasikan untuk install terminal di bawah ini agar lebih optimal dalam mengerjakan project.

Git bash, Git bash biasanya ter-install jika kita meng-install git di pc/laptop kita.

Apa itu Git ?

Git adalah sebuah Version Control System yaitu sistem yang mengelola perubahan dari sebuah dokumen, program komputer, website dan kumpulan informasi lain.

Selain itu, Git memungkinkan para pengembang perangkat lunak dari berbagai belahan dunia mengerjakan banyak projek bersama-sama tanpa mengharuskan bertatap muka.

Git sendiri disimpan dalam sebuah folder atau directory project yang biasa disebut repository

Sejarah Git



Git pertama kali dibuat pada tahun 2005 oleh Linus Torvalds, yang juga merupakan pencipta kernel Linux. Tujuan dari pembuatan Git adalah untuk menggantikan sistem kontrol versi yang sudah ada pada saat itu, yaitu BitKeeper, yang digunakan oleh kernel Linux.

sedangkan apa itu gitlab

GitLab adalah sebuah platform manajemen kode sumber yang berbasis web dan open source. Dibuat pertama kali pada tahun 2011 oleh Dmitriy Zaporozhets dan Valery Sizov, GitLab menyediakan fitur-fitur seperti version control, issue tracking, continuous integration, dan continuous deployment. Platform ini sangat populer di kalangan pengembang perangkat lunak dan organisasi yang membutuhkan manajemen kode sumber yang terpusat dan mudah digunakan.

Perintah Dasar Git

git init

git init adalah perintah untuk inisialisasi atau memasangkan git pada sebuah repository yang sedang terbuka di command prompt atau terminal.

pada contoh di atas, setelah memberikan perintah git init akan muncul keterangan “Initialized empty Git repository in . . . ” yang berarti folder git-demo yang sedang dibuka di cmd sudah dapat menggunakan Git di dalamnya.

git add

git add yaitu perintah git untuk menambahkan perubahan-perubahan pada direktori project sebelum menyimpan perubahan tersebut. Misalkan di repository project kita terdapat file index.html yang ingin kita simpan perubahannya.

```
git add index.html
```

git commit

git commit yaitu perintah dari git untuk menyimpan perubahan yang ada di repository dan perintah ini biasanya disertakan dengan -m “[keterangan_perubahan_yang_dilakukan]” untuk menuliskan perubahan yang dilakukan

```
git commit -m "menambahkan index.html"
```

git status

git status adalah perintah dalam git untuk memeriksa status repository, apakah ada file baru, perubahan pada file atau file sudah di add, atau apakah perubahan sudah di commit atau belum.

```
C:\Users\abdul\git-demo>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html

nothing added to commit but untracked files present (use "git add" to track)

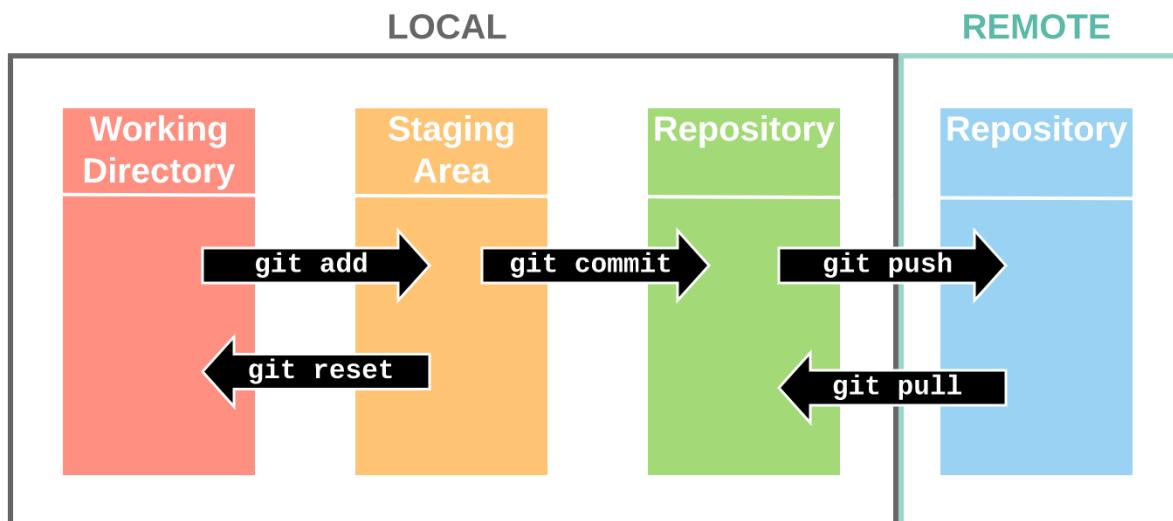
C:\Users\abdul\git-demo>■
```

git push

git push adalah perintah dalam git untuk mengirimkan perubahan perubahan file pada repository

git pull

git pull adalah perintah dalam git untuk mengambil perubahan terakhir dari repository di git



Latihan

untuk latihan kali ini kita akan mencoba untuk memasukan project yang kita buat masuk ke repositori gitlab

1. Petunjuk Pengerajan

Ikutilah langkah-langkah pengerajan di bawah ini:

pertama-tama buat akun di gitlab dulu https://gitlab.com/users/sign_up



GitLab.com

First name Last name

Username

Email
We recommend a work email address.

Password
Minimum length is 8 characters.

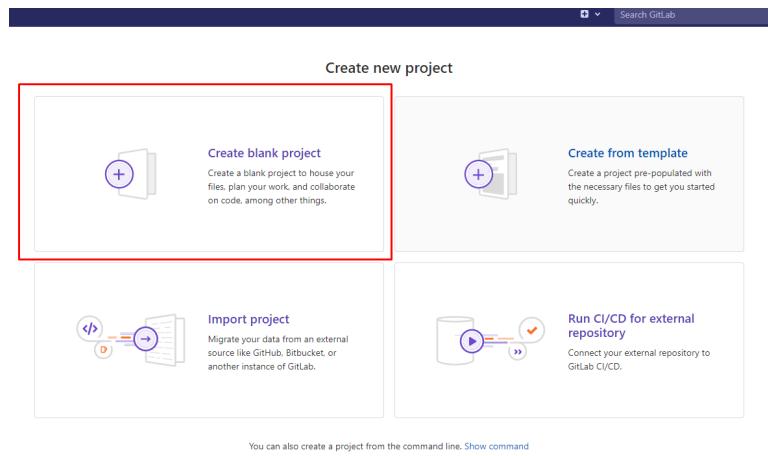
Saya bukan robot 
reCAPTCHA
Privasi • Persyaratan

Register

By clicking Register or registering through a third party you accept the GitLab Terms of Use and acknowledge the Privacy Policy and Cookie Policy

setelah berhasil membuat akun gitlab silahkan buat project di gitlabnya masuk kembali ke halaman <https://gitlab.com/> dan buat project baru





Buatlah satu repository project baru dengan nama project “**IM Ebook portal berita**”, pastikan visibility level di set “**private**”, dan centang pada bagian “**Initialize repository with a README**”.

Your work / Projects / New project / Create blank project

Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name
 Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL
 / **Project slug**
 Want to organize several dependent projects under the same namespace? [Create a group](#).

Project deployment target (optional)

Visibility Level [?](#)
 Private Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.
 Public The project can be accessed without any authentication.

Project Configuration
 Initialize repository with a README Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.
 Enable Static Application Security Testing (SAST) Analyze your source code for known security vulnerabilities. [Learn more](#).

Create project **Cancel**

tambahkan user penilai pada akun sebagai **maintance** di akun gitlab

IM Ebook portal berita

Project information Repository Issues Merge requests CI/CD Security and Compliance Deployments Packages and registries Infrastructure Monitor Analytics Wiki Snippets Settings

Activity Labels Members

IM Ebook portal berita

Project ID: 44326068

6 Commits 1 Branch 0 Tags 2 MB Project Storage

selesai rezky putra authored 1 minute ago

main / im-ebook-portal-berita / +

Find file Web IDE Clone

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Enable Auto DevOps Add Kubernetes cluster Set up CI/CD Add Wiki

Configure Integrations

Name	Last commit	Last update
portal-berita	selesai	1 minute ago

IM Ebook portal berita

Project information Activity Labels Members Repository Issues Merge requests CI/CD Security and Compliance Deployments Packages and registries Infrastructure

rezky putra > IM Ebook portal berita > Members

Project members

You can invite a new member to **IM Ebook portal berita** or invite another group.

Members 1
Filter members

Import from a project Invite a group Invite members

Account	Source	Max role	Expiration	Activity
rezky putra @rezkyputra	Direct member by rezky putra	Owner	Expiration date	User created: Jun 13, 2018 Access granted: Mar 15, 2023 Last activity: Apr 06, 2023

masukan username penilai

@rezkyputra dan buat rolenya menjadi Maintaner

rezky putra > IM Ebook portal berita > Members

Project members

You can invite a new member to **IM Ebook portal berita** or invite another group.

Members 1

Import from a project Invite a group Invite members

Account

rezky putra @rezkyputra

Invite members

You're inviting members to the **IM Ebook portal berita** project.

Username or email address

rezky putra

Select members or type email addresses

Select a role

Maintainer

Read more about role permissions

Access expiration date (optional)

YYYY-MM-DD

Cancel Invite

Expiration Activity

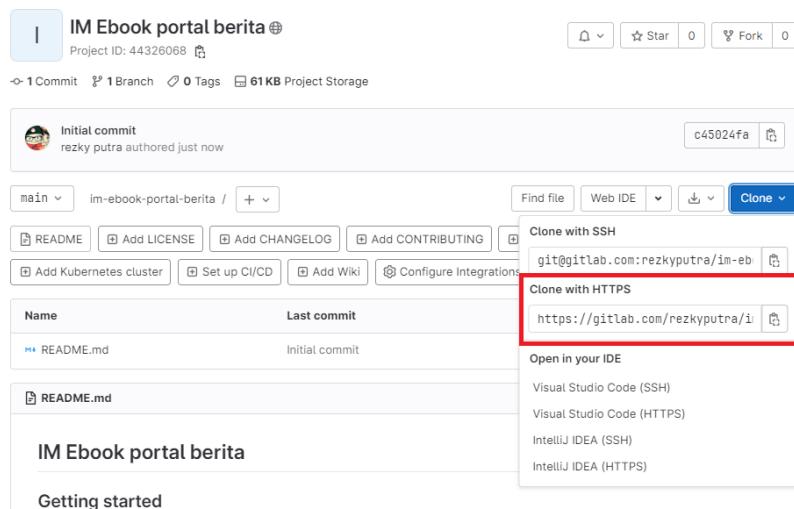
Expiration date User created: Jun 13, 2018
Access granted: Mar 15, 2023
Last activity: Apr 06, 2023

2. Clone repository

Langkah selanjutnya yaitu melakukan clone terhadap repository project yang sudah dibuat di Gitlab. Berikut langkah-langkah untuk melakukan clone:

1. Buka halaman repositori project yang ingin kita clone:

link "Clone with HTTPS", untuk copy dapat dilakukan dengan mengeklik tombol berlambang clipboard .



2. copy text yang ada di dalam Clone With HTTPS
3. Buka terminal/CMD kalian, pastikan posisi terminal dibuka di folder atau direktori yang ingin kalian gunakan untuk menyimpan repositori project yang akan diclone. Misal di C:/Users/John/Documents atau di drive D: atau dimanapun yang kalian inginkan.
4. jalankan perintah berikut di terminal, sesuaikan **link_https** dengan link yang kalian copy dari langkah ke 2.

```
git clone link_https
```

5. Pindah ke folder hasil clone dengan cara

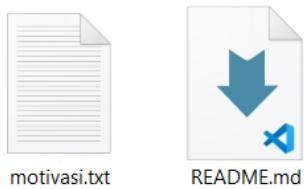
```
cd nama_folder_hasil_clone
```

3. Commit Pertama

Buat file "motivasi.txt". Kurang lebih struktur foldernya seperti ini

```
im-ebook-portal-berita/  
    motivasi.txt
```

> This PC > New Volume (D:) > ebook > im-ebook-portal-berita



Isi file "motivasi.txt" tersebut dengan motivasi kalian ingin menjadi seorang developer setelah mengisi "motivasi.txt", lakukanlah git add untuk menambahkan perubahan yang sudah kita buat ke dalam satu "commit".

```
git add .
```

lalu berikan commit disertai dengan message seperti berikut

```
git commit -m "push pertama saya"
```

nb: jika terdapat perintah seperti dibawah

```
Author identity unknown  
*** Please tell me who you are.  
Run  
  git config --global user.email "you@example.com"  
  git config --global user.name "Your Name"  
to set your account's default identity.  
Omit --global to set the identity only in this repository.
```

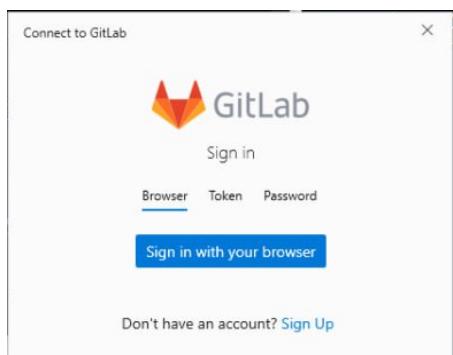
silahkan jalankan setup gitlabnya

```
git config --global user.name "username gitlab anda"  
git config --global user.email "email gitlab anda"
```

terakhir lakukan push

```
git push origin main
```

nb: saat pertama kali melakukan push akan muncul pop up seperti dibawah



silahkan klik "sign in with browser" maka akan muncul halaman baru di web browsernya
untuk silahkan klik "authorize"

tampilan setelah berhasil push

Name	Last commit	Last update
README.md	Initial commit	5 days ago
motivasi.txt	push pertama saya	5 days ago

Bagian 3 Membuat aplikasi portal berita dengan Laravel

BAB 9 pengenalan Laravel dan Laravel MVC

Apa itu Laravel?

Laravel adalah sebuah framework PHP yang bersifat open source, dirancang untuk memudahkan proses pengembangan aplikasi web dengan menyediakan berbagai fitur

Sejarah Laravel

Framework ini dikembangkan pertama kali oleh Taylor Otwell pada tahun 2011. Laravel awalnya dibuat sebagai alternatif untuk framework PHP lainnya yang saat itu sudah ada seperti CodeIgniter dan Yii.

Laravel dirancang dengan konsep yang modern dan modular, serta memiliki fitur-fitur yang sangat membantu dalam pembuatan aplikasi web seperti routing yang mudah, database migration, template engine, dan banyak lagi. Laravel juga menyediakan fitur keamanan yang cukup kuat seperti sistem autentikasi dan autorisasi.

Saat ini, Laravel telah menjadi salah satu framework PHP yang paling populer dan banyak digunakan di seluruh dunia. Framework ini juga aktif dikembangkan oleh komunitas pengembang yang besar, sehingga menjadikan Laravel sebagai salah satu pilihan yang sangat baik dalam pembuatan aplikasi web modern.

Laravel MVC

Laravel adalah sebuah framework yang menggunakan konsep arsitektur MVC (Model-View-Controller). Konsep ini membagi aplikasi menjadi tiga komponen utama, yaitu:

- Model: Merupakan bagian aplikasi yang bertanggung jawab untuk mengatur data dan bisnis logic. Model menyediakan akses ke database dan berinteraksi dengan tabel-tabel dalam database.
- View: Merupakan bagian aplikasi yang bertanggung jawab untuk menampilkan data ke pengguna. View mengambil data dari Model dan menampilkannya ke pengguna.
- Controller: Merupakan bagian aplikasi yang bertanggung jawab untuk mengontrol alur aplikasi. Controller menerima input dari pengguna melalui View, memproses data menggunakan Model, dan kemudian menampilkan hasilnya kembali ke pengguna melalui View.

Dengan memisahkan tiga komponen utama ini, konsep arsitektur MVC memungkinkan pengembang untuk membangun aplikasi yang lebih terstruktur, mudah dikelola, dan mudah diuji. Dalam Laravel, pengembang dapat dengan mudah membuat Model, View, dan Controller menggunakan perintah artisan, sebuah perintah baris yang disediakan oleh Laravel.

Laravel juga memiliki fitur yang memudahkan pengembang untuk mengatur rute (routing) URL dan menghubungkan controller dengan view, sehingga pengembang dapat memisahkan logika aplikasi dari tampilan dengan mudah.

Dengan menggunakan konsep arsitektur MVC, pengembang dapat membuat aplikasi yang lebih mudah dikembangkan, dipelihara, dan ditingkatkan. Selain itu, konsep MVC juga memungkinkan pengembang untuk mengembangkan aplikasi yang lebih mudah diuji dan memiliki performa yang lebih baik.

requirement Laravel 9

untuk laravel 9 sendiri membutuhkan minimal php versi 8.0 atau lebih

Version	PHP (*)	Release	Bug Fixes Until	Security Fixes Until
8	7.3 - 8.1	September 8th, 2020	July 26th, 2022	January 24th, 2023
9	8.0 - 8.2	February 8th, 2022	August 8th, 2023	February 6th, 2024
10	8.1 - 8.2	February 14th, 2023	August 6th, 2024	February 4th, 2025
11	8.2	Q1 2024	August 5th, 2025	February 3rd, 2026

instalasi Laravel

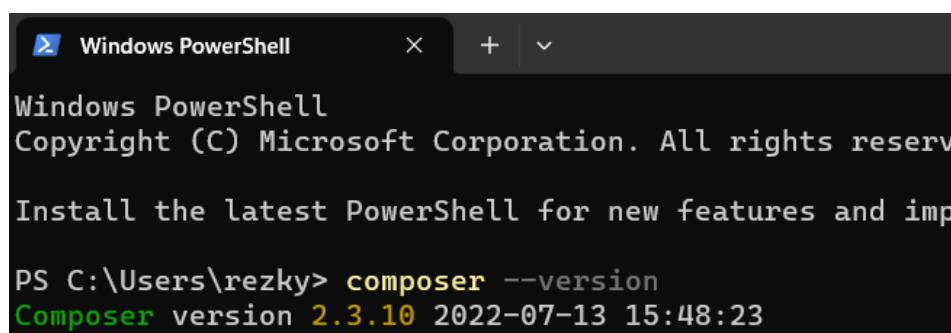
Sebelum kita mulai mencoba Laravel kita harus menginstall terlebih dahulu **composer**

Install Composer

Composer adalah *dependency manager PHP* atau software untuk mengatur project-project yang menggunakan PHP. Untuk menginstall composer di Windows, download terlebih dahulu dari halaman [download composer](#). Selanjutnya install dengan mengikuti petunjuk install dari composer-setup.exe.

Mengecek Composer

Untuk memastikan jika composer sudah terpasang atau belum silakan buka terminal/command prompt lalu ketikkan perintah composer -v jika tidak ada error maka terminal akan memberikan jawaban seperti berikut



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements.

PS C:\Users\rezky> composer --version
Composer version 2.3.10 2022-07-13 15:48:23
```

Membuat Project Laravel

Petunjuk pembuatan project web Laravel dapat dibaca selengkapnya di halaman dokumentasi resmi Laravel (Laravel Versi 9 yang akan kita gunakan)

<https://laravel.com/docs/9.x>

nb:disini kita menggunakan laravel 9 karena di laravel 10 masih terdapat error pada laravel/ui untuk auth nya

Setelah menginstall composer, kita dapat langsung membuat project Laravel baru di dalam direktori manapun di komputer kita. Untuk membuat project Laravel dengan nama project “portal-berita” kita gunakan perintah composer seperti berikut:

```
composer create-project laravel/laravel:^9.0 portal-berita
```



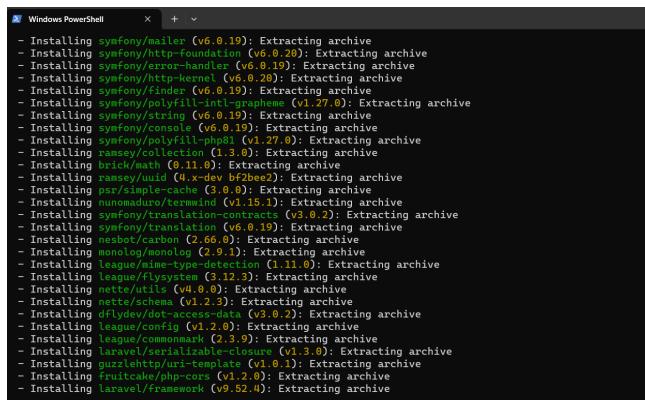
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\ebook\im-ebook-portal-berita> composer create-project laravel/laravel:^9.0 portal-berita
```

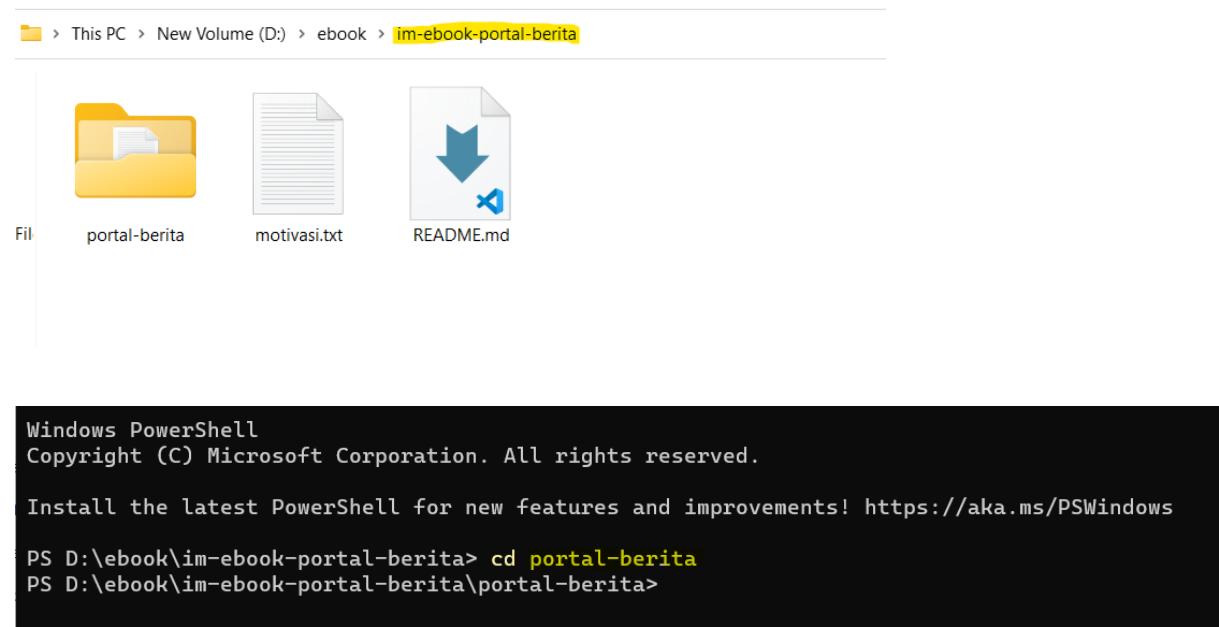
disini saya meninstall laravel 9 di **folder yang sudah dihubungkan git di materi git sebelumnya**

Composer akan mendownload semua file dan folder yang dibutuhkan untuk project Laravel proses download dan instalasi file kadang lama memakan waktu paling lambat 30 menit tergantung kecepatan internetnya



```
- Installing symfony/mailer (v6.0.19): Extracting archive
- Installing symfony/http-foundation (v6.0.20): Extracting archive
- Installing symfony/error-handler (v6.0.19): Extracting archive
- Installing symfony/http-kernel (v6.0.20): Extracting archive
- Installing symfony/finder (v6.0.19): Extracting archive
- Installing symfony/polyfill-intl-grapheme (v1.27.0): Extracting archive
- Installing symfony/string (v6.0.19): Extracting archive
- Installing symfony/console (v6.0.19): Extracting archive
- Installing symfony/polyfill-intl-icu (v1.27.0): Extracting archive
- Installing symfony/polyfill-mbstring (v1.3.0): Extracting archive
- Installing brick/math (v0.11.0): Extracting archive
- Installing ramsey/uuid (4.1.x-dev bf2be62): Extracting archive
- Installing psr/simple-cache (3.0.0): Extracting archive
- Installing numenaduro/termwind (v1.15.1): Extracting archive
- Installing symfony/translation-contracts (v3.0.2): Extracting archive
- Installing symfony/translation (v6.0.19): Extracting archive
- Installing nesbot/carbon (2.66.0): Extracting archive
- Installing league/colorable (2.2.0): Extracting archive
- Installing league/datetimeimmutable (1.11.0): Extracting archive
- Installing league/flysystem (3.12.3): Extracting archive
- Installing nette/utils (v4.0.0): Extracting archive
- Installing nette/schema (v1.2.3): Extracting archive
- Installing dflydev/dot-access-data (v3.0.2): Extracting archive
- Installing league/config (v1.2.0): Extracting archive
- Installing league/commonmark (2.3.9): Extracting archive
- Installing laravel/serializable渡鸦 (v1.0.0): Extracting archive
- Installing laravel/framework-classmap (v1.0.0): Extracting archive
- Installing fruitcake/laravel-cors (v1.2.0): Extracting archive
- Installing laravel/framework (v9.52.0): Extracting archive
```

setelah selesai mendownload maka sudah terdapat folder baru dengan nama “portal-berita” atau nama yang sesuai dengan yang kita berikan di perintah composer create-project .



Jika kita masuk ke dalam folder “portal-berita” tersebut dapat kita lihat bahwa di dalam foldernya sudah terdapat banyak file dan folder.

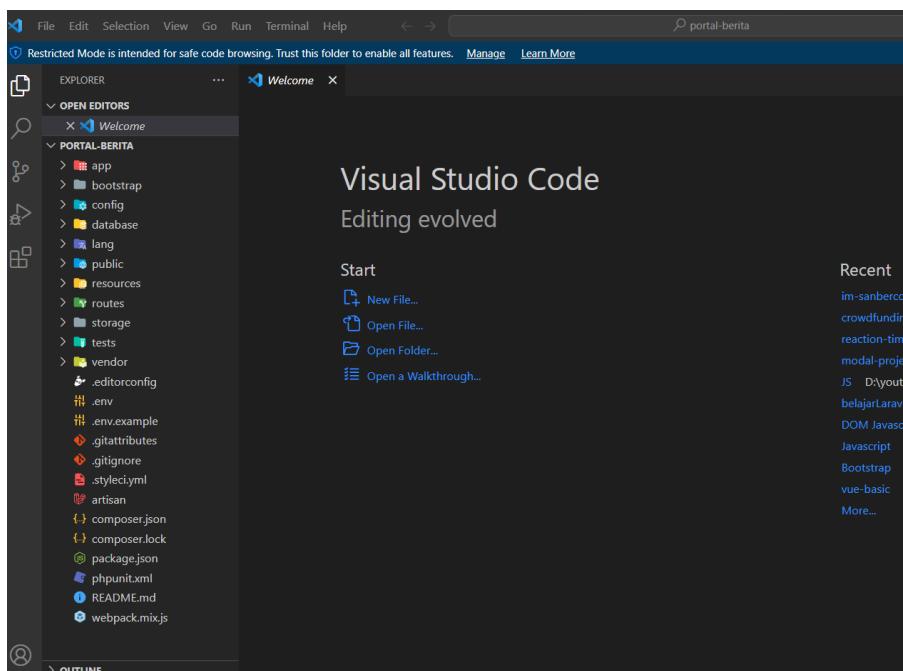
```
PS D:\ebook\im-ebook-portal-berita\portal-berita> ls

Directory: D:\ebook\im-ebook-portal-berita\portal-berita

Mode                LastWriteTime       Length Name
----                -----        ---- 
d----
```

disini kita bisa munulis “code .” untuk membuka folder laravel di text editor vs codenya

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> code .
```



fungsi file dan folder di laravel

Laravel adalah sebuah framework PHP yang sangat populer untuk membangun aplikasi web. Di bawah ini adalah penjelasan mengenai fungsi setiap folder dan file di Laravel:

- Folder "app": Folder ini berisi kode aplikasi utama. Di dalamnya, terdapat folder "Http" yang berisi file-file yang berkaitan dengan request dan response HTTP, seperti controller, middleware, dan request validation.
- Folder "bootstrap": Folder ini berisi file-file yang diperlukan untuk memuat framework Laravel. File yang paling penting di sini adalah "app.php", yang memuat instansi utama dari aplikasi Laravel.
- Folder "config": Folder ini berisi file-file konfigurasi aplikasi, seperti file database dan file-file konfigurasi lainnya.
- Folder "database": Folder ini berisi file-file yang berkaitan dengan database, seperti migration dan file seed.
- Folder "public": Folder ini berisi file-file publik, seperti file CSS, JavaScript, dan gambar. File "index.php" di sini adalah file utama yang dipanggil oleh web server untuk memulai aplikasi Laravel.
- Folder "resources": Folder ini berisi file-file yang berkaitan dengan sumber daya aplikasi, seperti file blade template, file CSS, JavaScript, dan gambar.
- Folder "routes": Folder ini berisi file-file yang berisi routing aplikasi, yaitu menentukan aksi apa yang dijalankan ketika request HTTP tertentu diterima.

- Folder "storage": Folder ini berisi file-file yang berkaitan dengan penyimpanan aplikasi, seperti file cache, file session, dan file log.
- Folder "tests": Folder ini berisi file-file yang berkaitan dengan pengujian aplikasi.
- File "artisan": File ini adalah command-line interface untuk Laravel, yang memungkinkan pengguna untuk melakukan banyak tugas, seperti menjalankan migration, menjalankan pengujian, dan menghasilkan kode.
- File ".env": File ini berisi variabel lingkungan untuk aplikasi, seperti pengaturan database, pengaturan cache, dan pengaturan email.
- File "composer.json": File ini berisi daftar dependensi aplikasi dan konfigurasi lainnya.
- File "package.json": File ini berisi daftar dependensi JavaScript aplikasi dan konfigurasi lainnya.
- File "webpack.mix.js": File ini adalah file konfigurasi Webpack, yang digunakan untuk mengompilasi aset JavaScript dan CSS.

Semua folder dan file di atas memainkan peran penting dalam pengembangan aplikasi Laravel dan harus dikelola dengan hati-hati untuk memastikan keamanan dan performa aplikasi yang optimal.

Menjalankan Laravel

Pada folder “portal-berita”, kita dapat jalankan perintah berikut pada terminal :

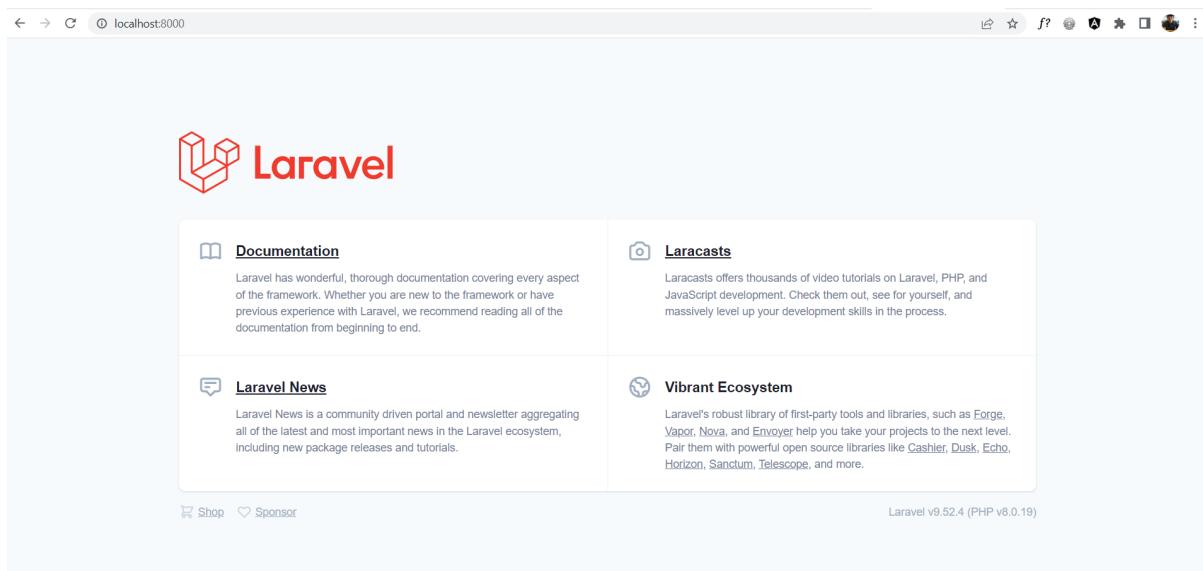
```
php artisan serve
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan serve
```

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

```
Press Ctrl+C to stop the server
```

kemudian jalan kan localhost:8000 di web browsernya



tampilan di atas adalah tampilan default dari laravelnya

Routing Laravel

Di Laravel, routing adalah proses mendefinisikan URL (Uniform Resource Locators) untuk aplikasi Anda dan mengaitkannya dengan tindakan controller yang tepat. Sistem routing Laravel memungkinkan Anda untuk mendefinisikan rute untuk semua permintaan HTTP aplikasi Anda, termasuk GET, POST, PUT/PATCH, DELETE, dan lain-lain.

```
Route::get($uri, $callback);
Route::post($uri, $callback);
Route::put($uri, $callback);
Route::patch($uri, $callback);
Route::delete($uri, $callback);
Route::options($uri, $callback);
```

Untuk mendefinisikan rute di Laravel, Anda dapat menggunakan facade Route, yang menyediakan berbagai metode untuk mendefinisikan rute. Sebagai contoh, untuk mendefinisikan rute untuk permintaan GET, Anda dapat menggunakan metode get seperti ini:

silahkan buka file web.php di folder routes pada project laravelnya

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
|
| Web Routes
|
| Here is where you can register web routes for your application.
These
| routes are loaded by the RouteServiceProvider within a group
which
| contains the "web" middleware group. Now create something
great!
|
*/
Route::get('/', function () {
    return view('welcome');
});
```

code diatas terdapat use Illuminate\Support\Facades\Route; berfungsi untuk memanggil fungsi routing di laravel

code diatas kita menggunakan method get kemudian '/' halaman awal pada saat kita membuka localhost:8000 kemudian ada callback yang menjalankan function untuk return file welcome.blade.php yang berada di folder resources/views

disini kita akan mencoba untuk mengubah tampilan dari resource view welcome.blade.php

resources/views/welcome.blade.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>Document</title>
</head>
<body>
    <h1>Selamat Datang di sanbercode</h1>
</body>
</html>
```

maka tampilan di web browsernya akan seperti di bawah



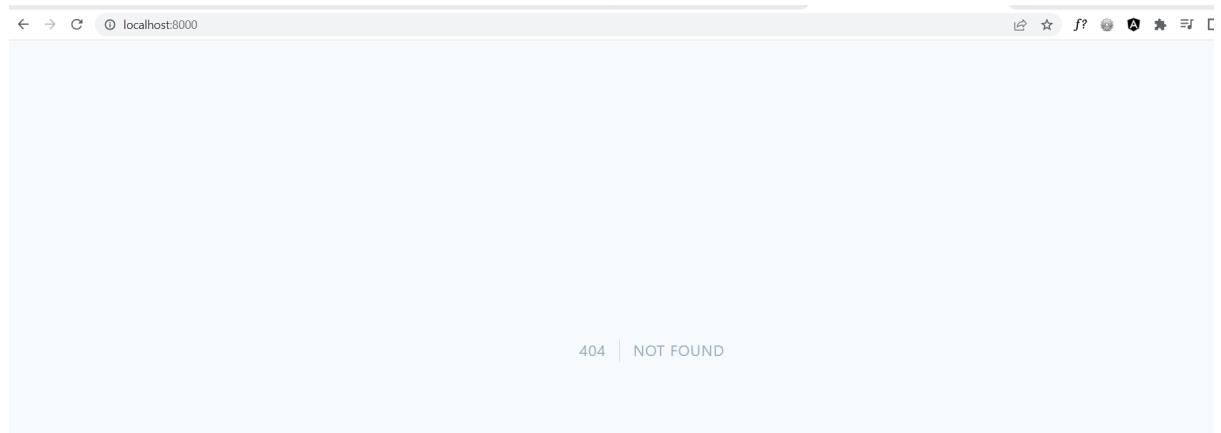
tapi jika kita menganti url di web.php dari '/' menjadi '/halo'

routes/web.php

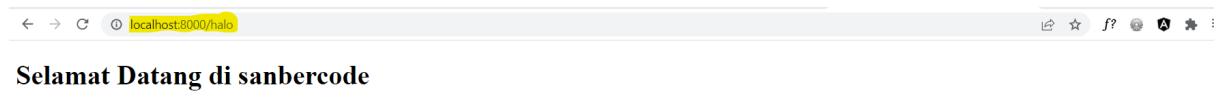
```
<?php
use Illuminate\Support\Facades\Route;
```

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

maka tampilan di web browser akan not found



kecuali kita menganti url nya menjadi localhost:8000/halo maka tampilan nya akan seperti dibawah



disini kita akan ubah kembali url di web nya menjadi '/'

Named Routes

named routes memungkin kan kita mengenerate nama sebuah route
disini kita akan membuat satu lagi route

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
})->name('home');

Route::get('/pendataan', function () {
    return view('pendataan');
})->name('pendataan');
```

buatlah file **pendataan.blade.php** dan edit file welcome.blade.php di folder resources/views

resources/view/welcome.blade.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>Document</title>
</head>
<body>
    <h1>Selamat Datang di sanbercode</h1>
    <a href="/pendataan">pendataan</a>
```

```
</body>  
</html>
```

code diatas kita menggunakan element anchor untuk berpindah kehalaman /pendataan berdasarkan url yang dibuat di web.php

resources/views/pendataan.blade.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>Document</title>  
</head>  
<body>  
    <h2>Ini halaman Pendataan</h2>  
    <a href="{{ route('home') }}">Home</a>  
</body>  
</html>
```

code diatas kita menggunakan element anchor untuk berpindah halaman ke halaman Selamat datang berdasarkan name dari route yang kita buat di web.php



← → C ⓘ localhost:8000

Selamat Datang di sanbercode

[pendataan](#)

pada saat kita mengklik link pendataan maka akan di arahkan ke halaman pendataan

Ini halaman Pendataan

[Home](#)

begitu pun jika kita mengklik link home maka akan diarahkan kembali kehalaman Home

Controller

Controller di Laravel adalah class PHP yang bertanggung jawab untuk menangani permintaan HTTP dan melakukan logika bisnis aplikasi. Controller dapat menerima input dari permintaan HTTP, memproses data, dan mengembalikan tampilan

Untuk membuat controller baru di Laravel, kita dapat menggunakan perintah **make:controller** di terminal. Misalnya, perintah berikut akan membuat controller **TestController**:

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan make:controller TestController
[INFO] Controller [D:\ebook\im-ebook-portal-berita\portal-berita\app\Http\Controllers\TestController.php] created successfully.
```

file **TestController.php** akan terbuat ke dalam folder App/Http/Controllers/

disini kita akan membuat route kita menggunakan controller

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\TestController;

Route::get('/', [TestController::class, 'home'])->name('home');

Route::get('/pendataan', [TestController::class,
'data'])->name('pendataan');
```

rute dari code diatas untuk URL / akan memanggil function home pada controller TestController ketika rute diakses melalui permintaan GET. dan rute kedua dari code diatas untuk URL /pendataan akan memanggil function pendataan pada controller TestController ketika rute diakses melalui permintaan GET dan pada bagian use kita mamanggil TestControllernya

App\Http\Controllers\TestController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class TestController extends Controller
{
    public function home()
    {
        return view('welcome');
    }

    public function data()
    {
        return view('pendataan');
    }
}
```

code diatas kita membuat 2 function pada class Test Controllernya yaitu home dan data yang mereturn ke view masing masing

Route Parameter

Route Parameter di Laravel memungkinkan kita untuk menangani URL dinamis dengan nilai yang berbeda tanpa harus menuliskan rute untuk setiap nilai. Kita dapat menambahkan parameter pada rute dengan menempatkan tanda kurung kurawal {} di dalam definisi rute.

disini kita akan membuat satu route lagi untuk contoh Route Parameter

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\TestController;

Route::get('/', [TestController::class, 'home'])->name('home');

Route::get('/pendataan', [TestController::class,
'data'])->name('pendataan');

Route::get('/data/{nama}', [TestController::class,
'datanama'])->name('data-nama');

```

Pada contoh di atas, parameter {nama} ditambahkan pada rute /data. Parameter ini akan menerima nilai yang dikirimkan melalui URL dan akan diproses pada fungsi closure yang terdapat di dalam route.

App\Http\Controllers\TestControllers.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class TestController extends Controller
{
    public function home()
    {
        return view('welcome');
    }

    public function data()
    {

```

```

        return view('pendataan');

    }

    public function datanama($nama)
    {
        return view('datanama', ["nama" => $nama]);
    }
}

```

code diatas kita memasukan \$nama kedalam parameter function datanama yang berfungsi untuk menerima parameter route {nama} pada web.php, dan juga disini kita melakukan passing data `["nama" => $nama]` yang berfungsi melempar parameter nama pada controller nya ke view datanama.blade.php

kemudian kita akan membuat folder baru di resource view bernama **datanama.blade.php** untuk menerima \$nama yang sudah di passing di controllernya

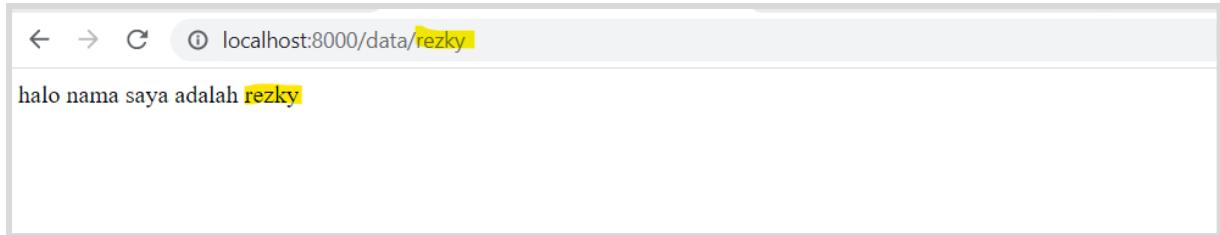
resources/views/datanama.blade.php

```

<!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>Document</title>
</head>
<body>
    halo nama saya adalah {{ $nama }}
</body>
</html>

```

code diatas kita menampilkan \$nama yang sudah di passing di controller nya `{{$nama}}`



Passing Data

Di Laravel, kita dapat melakukan passing data antara komponen seperti view, controller, dan rute menggunakan beberapa metode yang tersedia.

disini kita akan mencoba passing data

1. pertama kita atur dulu rute di web phnya

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\TestController;

Route::get('/', [TestController::class, 'home'])->name('home');

Route::get('/pendataan', [TestController::class,
'data'])->name('pendataan');

Route::get('/data/{nama}', [TestController::class,
'datanama'])->name('data-nama');

Route::get('/daftar', [TestController::class,
```

```
'daftar'])->name('daftar');
Route::post('/kirim', [TestController::class,
'kirim'])->name('kirim');
```

pada code diatas kita membuat 2 route, route yang pertama bertugas mengarahkan kita ke form view daftar menggunakan metode get kemudian route yang kedua berfungsi untuk aksi agar kita dapat mengirim data inputan ke halaman yang view yang lain menggunakan method post

2. kemudian kita buat form view nya

disini kita akan membuat file baru bernama daftar.blade.php

resources/views/daftar.blade.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>Document</title>
</head>
<body>
    <form action="{{route('kirim')}}" method="POST">
        @csrf
        <label>Nama</label> <br>
        <input type="text" name="nama"> <br> <br>
        <label>Alamat</label> <br>
        <textarea name="alamat" cols="30" rows="10"></textarea>
    <br> <br>
```

```
<input type="submit" value="kirim">
</form>
</body>
</html>
```

dari code diatas kita membuat sebuah form di mana terdapat action dengan method post yang akan mentriger route dengan name kirim pada web.php. kemudian kita memberi attribute name dengan value "nama" dan "alamat" sebagai request dari inputan di controllernya nanti

3. terakhir kita akan mengatur bagian logic nya di controller

app\Http\Controllers\TestController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class TestController extends Controller
{
    public function home()
    {
        return view('welcome');
    }

    public function data()
    {
        return view('pendataan');
    }

    public function datanama($nama)
    {
        return view('datanama', ["nama" => $nama]);
    }
}
```

```

public function daftar()
{
    return view('daftar');
}

public function kirim(Request $request)
{
    $name = $request->input('nama');
    $address = $request->input('alamat');

    return view('dashboard', ['name'=> $name,
'address'=>$address]);
}

```

pada kode diatas kita membuat 2 function yaitu function daftar dan function kirim, dimana function daftar adalah fungsi yang mengarahkan kita ke halaman view daftar.blade.php kemudian function kirim adalah function aksi login dimana kita memasukan fungsi laravel Request untuk mengambil name inputan pada view daftar, di function kirim terdapat 2 variabel yaitu \$name dan \$address yang menampung isi inputan dari name="address" dan name="nama" pada inputan view daftar.blade.php, disini kita melempar 'name' => \$name name berarti key dan \$name valuenya dan kita juga melempar 'address' => \$address ke sebuah file baru yaitu dashboard.blade.php

resources/views/dashboard.blade.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>Document</title>
</head>
<body>
    <h1>Halo nama saya {{ $name }}</h1>
    <p>saya beralamat di {{ $address }}</p>
</body>
</html>
```

code diatas kita menampil name dan address menggunakan `{{ $variabel }}` yaitu fungsi di engine blade yang fungsi nya mirip echo di php `<?php echo $variabel; ?>`

\$name dan \$addres didapatkan dari key passing data dari function kirim di controllernya

A screenshot of a web browser window titled "localhost:8000/daftar". The page contains two input fields: "Nama" with the value "rezky" and "Alamat" with the value "Makassar sulawesi Selatan". Below the inputs is a "kirim" button.



Halo nama saya rezky

saya beralamat di Makassar sulawesi Selatan

BAB 10 Laravel Templating

sebelum kita masuk ke templating kita pendalaman dulu tentang blade engine di laravel yang mengatur bagian front-end

Blade Engine Template

Blade adalah engine template yang disediakan oleh Laravel untuk mempermudah pembuatan tampilan pada aplikasi web. Blade memungkinkan kita untuk menggunakan fitur-fitur template seperti ekstensi, control flow , dll.

Berikut adalah beberapa fitur Blade pada Laravel:

Inklusi(include)

kita dapat menggunakan sintaks @include('namafile') untuk menyisipkan bagian-bagian tampilan ke dalam tampilan utama. Dengan begitu, kita tidak perlu menulis ulang kode yang sering digunakan pada tampilan yang berbeda-beda.

Data forwarding

kita dapat menampilkan data dari controller ke tampilan menggunakan sintaks {{ \$nama_variabel }}. seperti passing data di materi sebelumnya Dengan begitu, Anda dapat menampilkan data dari variabel pada tampilan.

Komentar

kita dapat menambahkan komentar pada tampilan menggunakan sintaks {{-- komentar --}}. Komentar ini tidak akan ditampilkan pada tampilan di web browser, sehingga dapat digunakan untuk dokumentasi atau debugging.

Looping

kita dapat menggunakan sintaks @foreach, @for, @while, @forelse untuk melakukan looping pada sebuah array. Dengan begitu, Anda dapat menampilkan data pada tampilan secara berulang-ulang dengan cara yang lebih mudah.

beberapa sintax looping

- foreach

```
@foreach ($users as $user)
    {{-- kode--} }
@endforeach
```

- for

```
@for ($i = 0; $i < 10; $i++)
    {{-- kode--} }
@endfor
```

- while

```
@while (true)
    {{-- kode--} }
@endwhile
```

- forelse

```
@forelse ($variable as $value)
    {{-- jika tidak data kosong--} }
@empty
    {{-- jika data kosong--} }
@endforelse
```

Control Flow

kita dapat menggunakan sintaks @if, @elseif, @else, dan @endif untuk melakukan control flow atau kondisi pada tampilan. Dengan begitu, kita dapat menampilkan tampilan yang berbeda-beda berdasarkan kondisi yang diberikan.

```
@if (count($records) === 1)
    I have one record!
```

```
@elseif (count($records) > 1)
    I have multiple records!
@else
    I don't have any records!
@endif
```

Pemanggilan fungsi

kita dapat memanggil fungsi pada tampilan menggunakan sintaks {{ nama_fungsi() }}.

Dengan begitu, kita dapat melakukan operasi pada data yang akan ditampilkan pada tampilan.

nilai boolean

kita dapat menggunakan sintaks @if(\$var), @unless(\$var), dan @isset(\$var) untuk melakukan pengecekan nilai boolean pada tampilan.

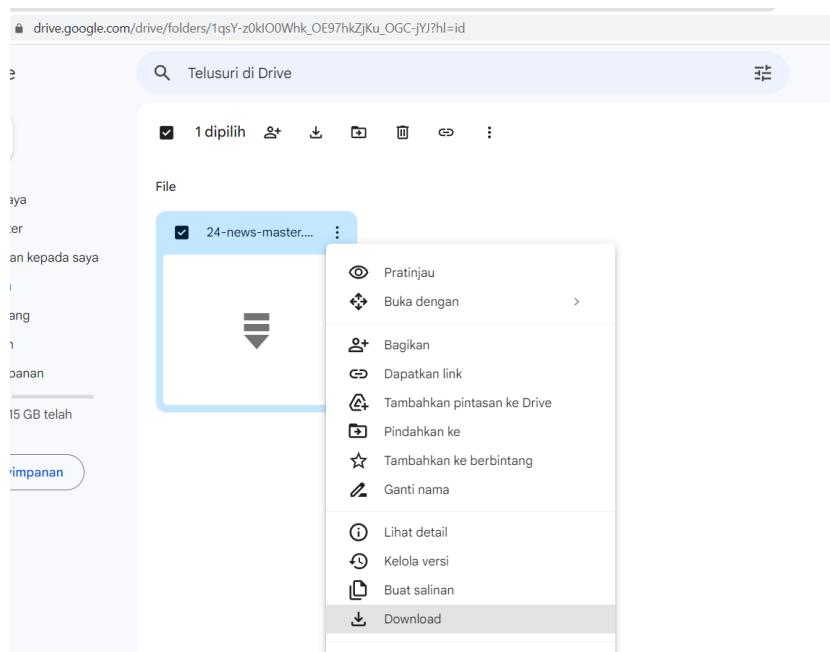
Templating Portal Berita

disini kita akan mengatur templating website kita menggunakan template gratis yang saya dapat pada di themawagon :

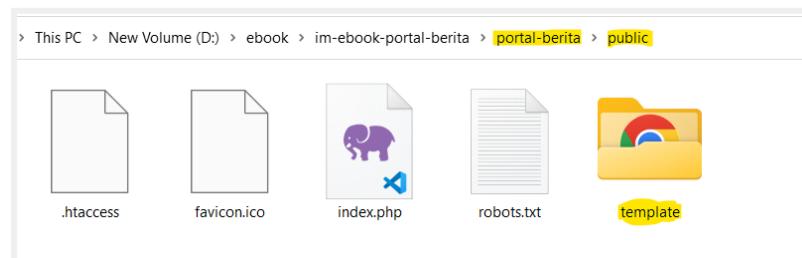
berikut linknya:

https://drive.google.com/drive/folders/1qsY-z0kIO0Whk_OE97hkZjKu_OGC-jYJ?hl=id

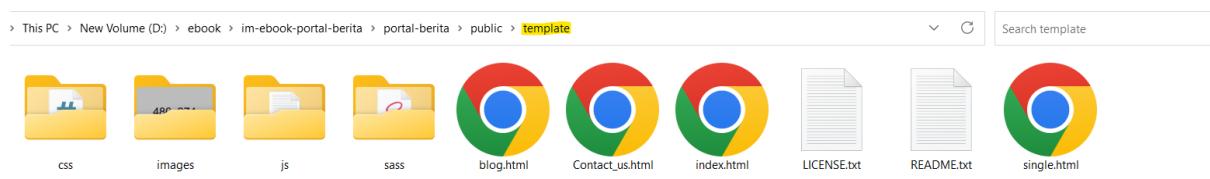
silahkan download templatanya



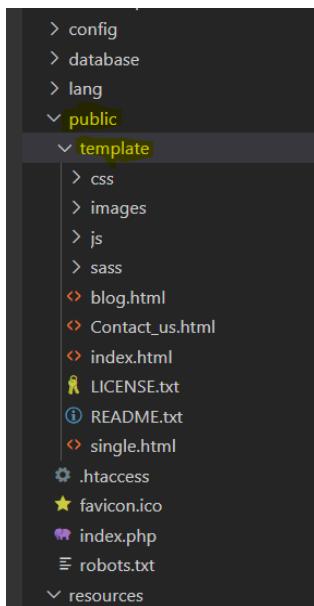
kemudian disilahkan di ekstrak file rar nya dan pindahkan ke folder public pada project laravel kita ‘portal-berita’ ubah nama file nya dari ‘24-news-master’ ke ‘template’



didalam folder template



tampilan di vs code



disini kita akan menggunakan fungsi @extends, @section, dan @yield untuk mengatur templatingnya

@extends, @section, dan @yield adalah tiga fitur penting pada Blade, engine template pada Laravel, yang digunakan untuk mempermudah pembuatan tampilan pada aplikasi web.

Berikut adalah penjelasan dari ketiga fitur tersebut:

@extends('namafile')

Fitur @extends digunakan untuk mewarisi tampilan pada tampilan yang lain. Dengan begitu, Anda dapat membuat tampilan induk yang akan digunakan pada beberapa tampilan turunan.

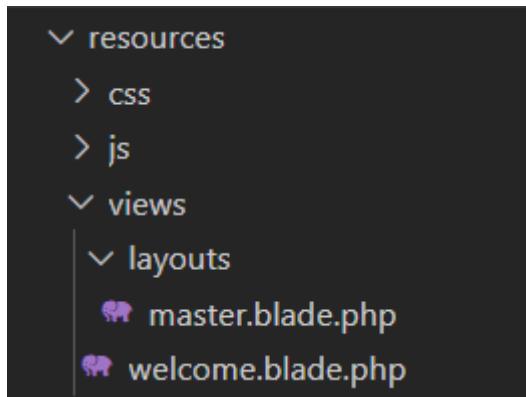
@section('nama') ... @endsection

Fitur @section digunakan untuk menentukan bagian tampilan yang akan diisi pada file turunan.

@yield('nama')

Fitur @yield digunakan untuk menampilkan isi dari @section pada file turunan pada bagian yang sesuai.

pertama-tama kita buat file view untuk mengatur tampilan induk nya, yang kita simpan di subfolder layout di folder views dengan nama file **master.blade.php** dan hapus semua file di views kecuali file welcome.blade.php yang sudah dibuat di materi sebelumnya



resources/views/layouts/**master.blade.php**

```
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
    <title>Portal News</title>
    <link href="{{asset('template/css/media_query.css')}}" rel="stylesheet" type="text/css"/>
    <link href="{{asset('template/css/bootstrap.css')}}" rel="stylesheet" type="text/css"/>
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet"
integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGofQNHSoD2xbE+QkPxCAF1NEevOEH3S10sibVcOQVnN" crossorigin="anonymous">
    <link href="{{asset('template/css/animate.css')}}" rel="stylesheet" type="text/css"/>
    <link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet">
```

```

<link href="{{asset('template/css/owl.carousel.css')}}" rel="stylesheet" type="text/css"/>
<link href="{{asset('template/css/owl.theme.default.css')}}" rel="stylesheet" type="text/css"/>
<!-- Bootstrap CSS -->
<link href="{{asset('template/css/style_1.css')}}" rel="stylesheet" type="text/css"/>
<!-- Modernizr JS -->
<script
src="{{asset('template/js/modernizr-3.5.0.min.js')}}"></script>
</head>
<body>

{{-- Header --}}
<div class="container-fluid bg-faded fh5co_padd_mediya padding_786">
<div class="container padding_786">
<nav class="navbar navbar-toggleable-md navbar-light ">
<button class="navbar-toggler navbar-toggler-right mt-3" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"><span class="fa fa-bars"></span></button>
<a class="navbar-brand" href="#"></a>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav mr-auto">
<li class="nav-item ">
<a class="nav-link" href="/">Home <span class="sr-only">(current)</span></a>
</li>
<li class="nav-item ">
<a class="nav-link" href="/about">About <span class="sr-only">(current)</span></a>

```

```

        </li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle"
href="#" id="dropdownMenuButton2" data-toggle="dropdown"
aria-haspopup="true"
aria-expanded="false">Kategori <span
class="sr-only">(current)</span></a>
            <div class="dropdown-menu"
aria-labelledby="dropdownMenuLink_1">
                <a class="dropdown-item"
href="#">Tampil Kategori</a>
                <a class="dropdown-item"
href="#">Tambah Kategori</a>
            </div>
        </li>
    </ul>
</div>
</nav>
</div>
</div>
{{-- End Header --}}


{{-- Content --}}
<div class="container-fluid pb-4 pt-4 paddding">
    @yield('content')
</div>
{{-- End Content --}}


{{-- Footer --}}
<div class="container-fluid fh5co_footer_right_reserved">
    <div class="container">
        <div class="row ">
            <div class="col-12 col-md-6 py-4 Reserved"> ©
Copyright 2018, All rights reserved. Design by <a
href="https://freehtml5.co" title="Free HTML5 Bootstrap
templates">FreeHTML5.co</a>. </div>

```

```

        </div>
    </div>
</div>
{{-- End Footer --} }

<div class="gototop js-top">
    <a href="#" class="js-gotop"><i class="fa fa-arrow-up"></i></a>
</div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script
src="{{asset('template/js/owl.carousel.min.js')}}"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js"
integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIxUYHKkDx6Qin1DkWx51bBrb"
crossorigin="anonymous"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js"
integrity="sha384-vBWWzlZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVkJ+jo0ieGizqPLForn"
crossorigin="anonymous"></script>
<script
src="{{asset('template/js/jquery.waypoints.min.js')}}"></script>
<script src="{{asset('template/js/main.js')}}"></script>

</body>
</html>

```

`{{asset()}}` adalah fungsi bawaan pada Laravel yang digunakan untuk asset yang ada di folder public (file gambar, file JavaScript, file CSS, dll) pada aplikasi web contohnya kita memanggil file asset yang main.js yang terdapat di folder public=>template=>js=>main.js `{{asset('template/js/main.js')}}`

disini kita menggunakan fungsi template blade yaitu `@yield` yang akan di hubungkan dengan `@section` tampilan file turunannya

sebelum kita lanjut untuk membuat tampilan turunnya pertama tama kita atur dulu web.php dan controller nya disini kita akan membuat 1 controller lagi yaitu DashboardController

```
php artisan make:controller DashboardController
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan make:controller DashboardController
[INFO] Controller [D:\ebook\im-ebook-portal-berita\portal-berita\app\Http\Controllers\DashboardController.php] created successfully.
PS D:\ebook\im-ebook-portal-berita\portal-berita>
```

kita atur route di web.php

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\TestController;
use App\Http\Controllers\DashboardController;

Route::get('/', [DashboardController::class,
'home'])->name('home');
Route::get('/about', [DashboardController::class,
'about'])->name('about');
```

code diatas kita membuat 2 rute dengan url yaitu / dan /about yang mengarah ke function about dan home di DasboardController disini kita mengganti semua route yang sudah kita buat di materi sebelumnya

app\Http\Controllers\DashboardController.php

```
<?php

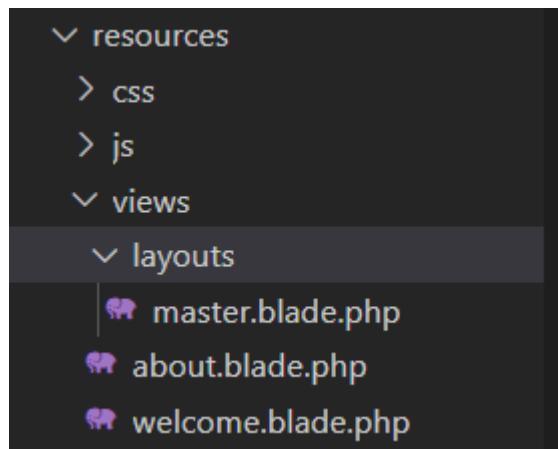
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class DashboardController extends Controller
{
    public function home()
    {
        return view('welcome');
    }

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

disini kita buat satu file lagi di folder views yaitu about.blade.php



resources/views/welcome.blade.php

```
@extends('layouts.master')

@section('content')
<div class="container padding">
    <div class="row mx-0">
        <div class="col-md-8 animate-box"
data-animate-effect="fadeInLeft">
            <div>
                <div class="fh5co_heading
fh5co_heading_border_bottom py-2 mb-4">News</div>
            </div>
            <div class="row pb-4">
                <div class="col-md-5">
                    <div class="fh5co_hover_news_img">
                        <div class="fh5co_news_img">
                        alt="" /></div>
                    <div></div>
                </div>
                <div class="col-md-7 animate-box">
                    <a href="single.html" class="fh5co_magna
py-2"> Magna aliqua ut enim ad minim veniam quis
nostrud quis exercitation ullamco. </a>
                    Thomson Smith - April 18, 2016
                    <div class="fh5co_consectetur"> Amet
consectetur adipisicing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim
ad minim veniam, quis nostrud exercitation.
                </div>
            </div>
        </div>
        <div class="col-md-3 animate-box">
```

```

data-animate-effect="fadeInRight">
    <div>
        <div class="fh5co_heading fh5co_heading_border_bottom py-2 mb-4">Kategory</div>
    </div>
    <div class="clearfix"></div>
    <div class="fh5co_tags_all">
        <a href="#" class="fh5co_tagg">Business</a>
        <a href="#" class="fh5co_tagg">Technology</a>
        <a href="#" class="fh5co_tagg">Sport</a>
        <a href="#" class="fh5co_tagg">Art</a>
        <a href="#" class="fh5co_tagg">Lifestyle</a>
    </div>
    </div>
</div>
<div class="row mx-0 animate-box"
data-animate-effect="fadeInUp">
    <div class="col-12 text-center pb-4 pt-4">
        <a href="#" class="btn_mange_pagging"><i class="fa fa-long-arrow-left"></i>&ampnbsp&ampnbsp Previous</a>
        <a href="#" class="btn_pagging">1</a>
        <a href="#" class="btn_pagging">2</a>
        <a href="#" class="btn_pagging">3</a>
        <a href="#" class="btn_pagging">...</a>
        <a href="#" class="btn_mange_pagging">Next <i
class="fa fa-long-arrow-right"></i>&ampnbsp&ampnbsp </a>
    </div>
</div>
</div>
@endsection

```

resources/views/about.blade.php

```
@extends('layouts.master')

@section('content')


![Cinque Terre](https://dummyimage.com/480x320/000/fff)



# Nama Saya



### Motivasi Saya adalah



Pellentesque condimentum augue et odio lacinia tempus. Donec quam ligula, congue vel interdum nec, vehicula scelerisque neque. Ut hendrerit, erat ac molestie sodales, diam lectus pretium est, et rhoncus odio libero a tellus. Maecenas convallis nisi at ex imperdiet, vitae dictum enim facilisis. Suspendisse potenti. Integer fringilla aliquet fringilla. Vivamus ut nisl metus. Praesent posuere nunc placerat, interdum mi ac, vehicula tellus. Maecenas augue nibh, porttitor a odio sollicitudin, iaculis tristique neque. Donec sit amet massa nibh. In justo quam, viverra accumsan nunc quis, ultrices tempus nisl. Nullam non sagittis justo. Etiam ornare nibh a nunc pharetra sollicitudin. Integer commodo mattis aliquet.


```

Curabitur egestas quam ut neque condimentum, eu bibendum magna gravida. Mauris molestie pretium ex, ultricies sagittis nisi maximus sed. Duis et dignissim felis, eu ultrices augue. Morbi id posuere risus, ac lobortis justo. Etiam ac faucibus dolor. Morbi at scelerisque nunc. Duis vel massa feugiat, pulvinar felis quis, vehicula ipsum. Donec sem quam, lacinia ullamcorper bibendum in, rutrum ut magna. Duis vestibulum vitae massa rhoncus cursus. Maecenas molestie, dolor a aliquet

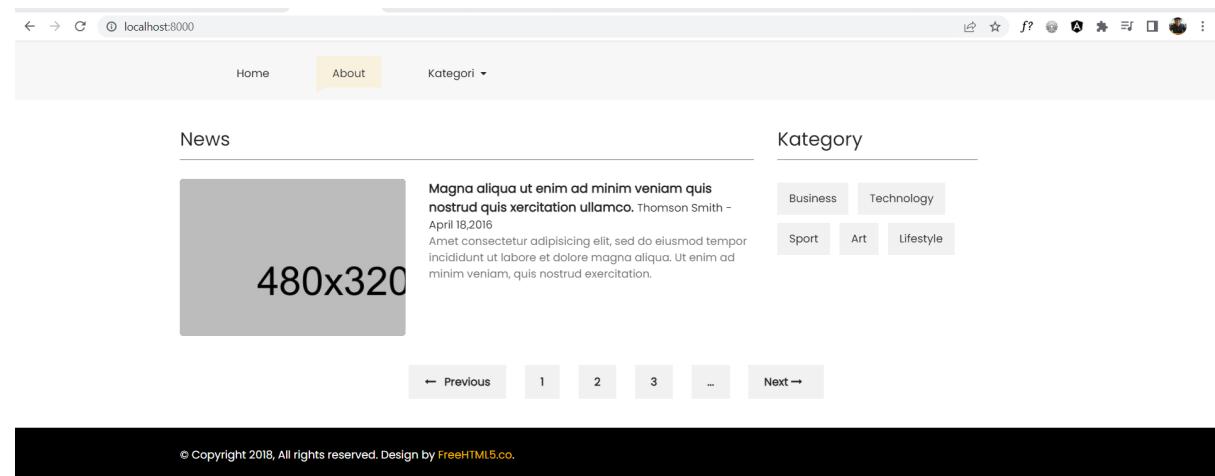
```
viverra, lacus lectus ultricies est, vel malesuada justo nunc sed  
purus. Donec malesuada, magna non vehicula commodo, urna metus  
dictum risus, non porta tellus est eget felis.
```

```
        Morbi id diam vitae ipsum imperdiet aliquet et eu  
odio. Morbi lectus felis, consectetur nec pulvinar vitae, gravida  
et nunc. Curabitur ex massa, bibendum eget lectus sit amet,  
premium lacinia tellus. Fusce sit amet tortor ac mauris volutpat  
vestibulum. Nulla arcu arcu, vulputate quis auctor venenatis,  
ultrices ut nisl. Praesent accumsan porta vestibulum. Nunc in  
magna tortor. In hac habitasse platea dictumst.
```

```
</p>  
</div>  
</div>  
</div>  
@endsection
```

di 2 code diatas kita menggunakan **@extends** untuk memanggil file tampilan induk **master.blade.php** dan juga kita menggunakan **@section** yang terhubung dengan **@yield** di tampilan induk

Tampilan **welcome.blade.php** di web browser localhost:8000



tampilan **about.blade.php** di web browser localhost:8000/about

The screenshot shows a web page with a header containing navigation links: Home, About (which is highlighted), and Kategori. Below the header is a large black circle placeholder image with the text "480 x 320" in white. To the right of the image is a section titled "Nama Saya" with the subtitle "Motivasi Saya adalah". A large amount of placeholder text follows, starting with "Pellentesque condimentum augue et odio lacinia tempus. Donec quam ligula, congue vel interdum nec, vehicula scelerisque neque. Ut hendrerit, erat ac molestie sodales, diam lectus pretium est, et rhoncus odio libero a tellus. Maecenas convallis nisi at ex imperdiet, vitae dictum enim facilisis. Suspendisse potenti. Integer fringilla aliquet fringilla. Vivamus ut nisl metus. Praesent posuere nunc placerat, interdum mi ac, vehicula tellus. Maecenas augue nibh, porttitor a odio sollicitudin, iaculis tristique neque. Donec sit amet massa nibh. In justo quam, viverra accumsan nunc quis, ultrices tempus nisl. Nullam non sagittis justo. Etiam ornare nibh a nunc pharetra sollicitudin. Integer commodo mattis aliquet. Curabitur egestas quam ut neque condimentum, eu bibendum magna gravida. Mauris molestie pretium ex, ultricies sagittis nisi maximus sed. Duis et dignissim felis, eu ultrices augue. Morbi id posuere risus, ac lobortis justo. Etiam ac faucibus dolor. Morbi at scelerisque nunc. Duis vel massa feugiat, pulvinar felis quis, vehicula ipsum. Donec sem quam, lacinia ullamcorper bibendum in, rutrum ut magna. Duis vestibulum vitae massa rhoncus cursus. Maecenas molestie, dolor a aliquet viverra, locus lectus ultricies est, vel malesuada justo nunc sed purus. Donec malesuada, magna non vehicula commodo, urna metus dictum risus, non porta tellus est eget felis. Morbi id diam vitae ipsum imperdiet aliquet et eu odio. Morbi lectus felis, consectetur nec pulvinar vitae, gravida et nunc. Curabitur ex massa, bibendum eget lectus sit amet, pretium lacinia tellus. Fusco sit amet tortor ac mauris volutpat vestibulum. Nulla arcu arcu, vulputate quis auctor venenatis, ultrices ut nisl. Praesent accumsan porta vestibulum. Nunc in magna tortor. In hac habitasse platea dictumst."

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

di code di atas kita masih menggunakan tampilan dummy di codenya kita akan mengatur kolom dan table di database nya menggunakan laravel migration di materi selanjutnya

BAB 11 Laravel Migration

Apa itu Laravel Migration

Laravel Migration yaitu pengatur perubahan-perubahan pada database di project Laravel kita.

Kenapa perubahan pada Database perlu diatur ? Kan database tinggal bikin aja secara manual di mysql atau di phpmyadmin ?

Jawabannya adalah karena ke depannya project Laravel kita bisa jadi dikerjakan oleh banyak developer. Karena development dikerjakan bersama sehingga database untuk masing-masing developer pun harus sama. Database pada tahap development atau pengembangan biasanya menggunakan database di localhost. sehingga untuk setiap developer memiliki database di localhost nya masing-masing. Isi atau records dari database masing-masing bisa jadi berbeda, tapi untuk struktur database seperti nama tabel, nama kolom, hubungan antar tabel, foreign key, dan lain-lain harus sama.

Oleh karena itu, Laravel menyediakan Migration agar kita dapat mengatur database kita secara rapi bahkan ketika dikerjakan oleh banyak developer secara bersamaan.

Berikut ini adalah langkah-langkah dasar untuk membuat migration di Laravel:

1. Buka terminal Anda dan navigasi ke direktori proyek Laravel Anda.
2. Jalankan perintah berikut untuk membuat file migration baru:

```
php artisan make:migration create_table_name
```

Ini akan membuat file migration baru di direktori database/migrations dengan nama seperti YYYY_MM_DD_HHMMSS_create_table_name.php.

3. Kemudian, buka file migration yang telah dibuat dan tentukan skema database yang ingin Anda buat atau perbarui. Anda dapat menggunakan metode Laravel Schema Builder untuk menentukan tabel, kolom, indeks, dan relasi antar tabel.
4. Setelah Anda menentukan skema, jalankan perintah berikut untuk menjalankan migration:

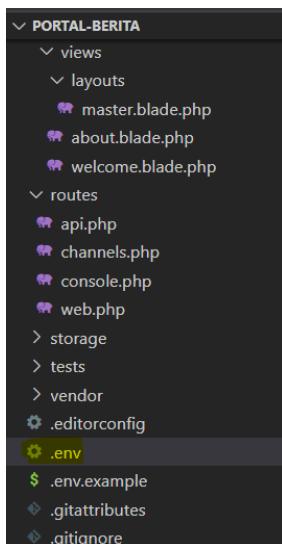
```
php artisan migrate
```

untuk materi sekarang kita akan coba membuat migration untuk project portal berita yang sudah kita buat

file .env

File .env pada Laravel adalah sebuah file konfigurasi yang berisi nilai-nilai konfigurasi spesifik lingkungan seperti detail koneksi database, detail server email, kunci API, dan pengaturan lainnya yang dapat berbeda tergantung pada lingkungan di mana aplikasi dijalankan (misalnya, pengembangan, staging, atau produksi).

File .env biasanya terletak di direktori utama dari proyek Laravel dan dibaca oleh library dotenv. Library ini memuat nilai-nilai konfigurasi dari file .env ke dalam variabel lingkungan server, yang kemudian dapat diakses oleh aplikasi Laravel.



Berikut ini adalah contoh file .env:

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
APP_DEBUG=true
APP_URL=http://localhost

DB_CONNECTION=mysql
```

```
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=namadatabase  
DB_USERNAME=namauusername  
DB_PASSWORD=isijikaada
```

Data Tipe Migration

Dalam Laravel Migration, terdapat beberapa tipe data untuk kolom di databasenya yang bisa digunakan. banyak tipe data yang bisa digunakan

<https://laravel.com/docs/9.x/migrations#available-column-types>

- BigIncrements: Tipe data ini digunakan untuk menentukan kolom id yang otomatis bertambah nilainya dengan tipe data big integer dan extra auto_increment dan unsigned.
- BigInteger: Tipe data ini digunakan untuk menentukan kolom dengan tipe data big integer.
- String: Tipe data ini digunakan untuk menentukan kolom dengan tipe data varchar di kolom.
- text: Tipe data ini digunakan untuk menentukan kolom dengan tipe data teks.
- Timestamp: Tipe data ini digunakan untuk menentukan kolom dengan tipe data timestamp.
- id: tipe data ini digunakan untuk membuat kolom dengan tipe data big integer unsigned yang sudah ter primary key dan memiliki extra yaitu auto increment
- unsignedBigInteger: tipe data ini digunakan untuk membuat kolom dengan tipe big integer dengan extra unsigned (data tidak boleh nilai angka minus)

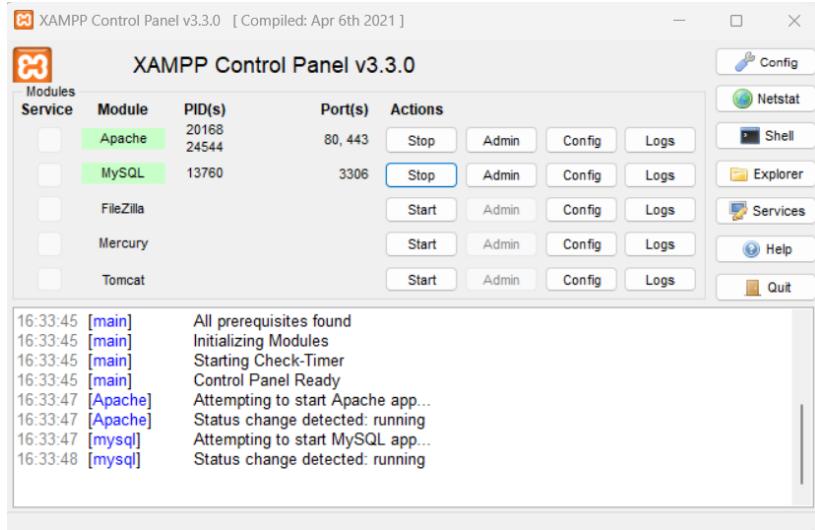
Jenis Jenis indeks di Laravel Migration

Berikut ini adalah jenis-jenis indeks yang tersedia pada Laravel Migration:

command	description
\$table->uuid('id')->primary();	primary() berfungsi untuk mengubah kolom jadi primary key
\$table->string('email')->unique();	unique() berfungsi membuat kolom menjadi unique
\$table->unsignedBigInteger('user_id');	foreign() berfungsi membuat foreign key yaitu kolom user_id yang terhubung dengan table users kolom id
\$table->foreign('user_id')->references('id')->on('users');	
\$table->text('descripsion')->nullable();	nullable berfungsi membuat kolomnya dapat bisa dikosongkan/tidak wajib diisi

Latihan

pertama-tama kita koneksi dulu project laravel kita dengan database lokal kita start “apache” dan “mysql” di xamppnya



setelah itu setting database di file .env di project laravelnya dan buat database dengan nama yang sama dengan DB_DATABASE di .env

.env

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE= portalberita
DB_USERNAME=root
DB_PASSWORD=
```

code diatas kita mengatur username, database, dan password databasenya (defaultnya jika kita menginstall xampp secara default username nya akan root dan passwordnya kosong) kemudian kita buat database di phpmyadmin dengan nama “portalberita” sesuai dengan nama database yang sudah di setting di .env project laravel nya

klik "baru" atau "new" di localhost/phpmyadmin

The screenshot shows the phpMyAdmin interface with the following details:

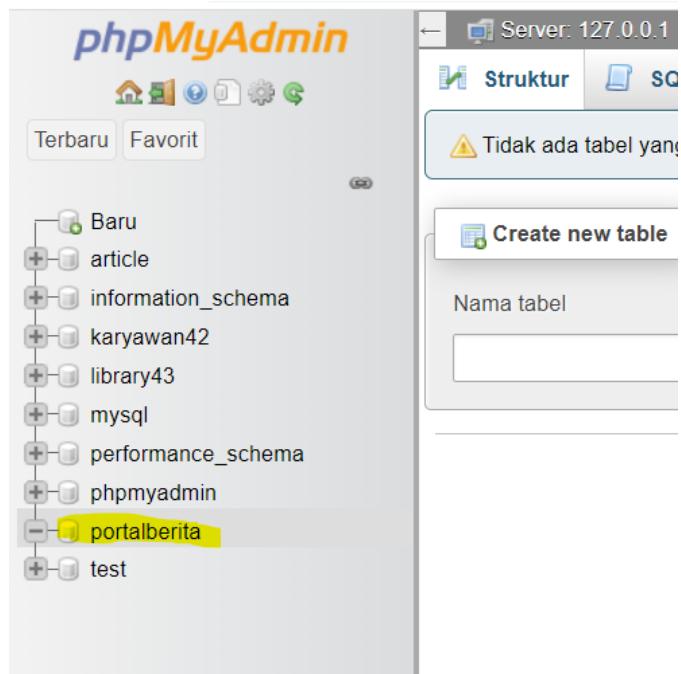
- Header:** Server: 127.0.0.1, Basis data, SQL, Status, Akun pengguna.
- Left Sidebar:** Terbaru, Favorit, Baru, article, information_schema, karyawan42, library43, mysql, performance_schema, phpmyadmin, test.
- General Settings (Pengaturan umum):** Server connection collation: utf8mb4_unicc, Pengaturan lain.
- Display Settings (Pengaturan tampilan):** Bahasa (Language): Bahasa Indonesia - In.

isi input field nya dengan "portalberita" kemudian klik "buat" atau "create"

The screenshot shows the phpMyAdmin interface with the following details:

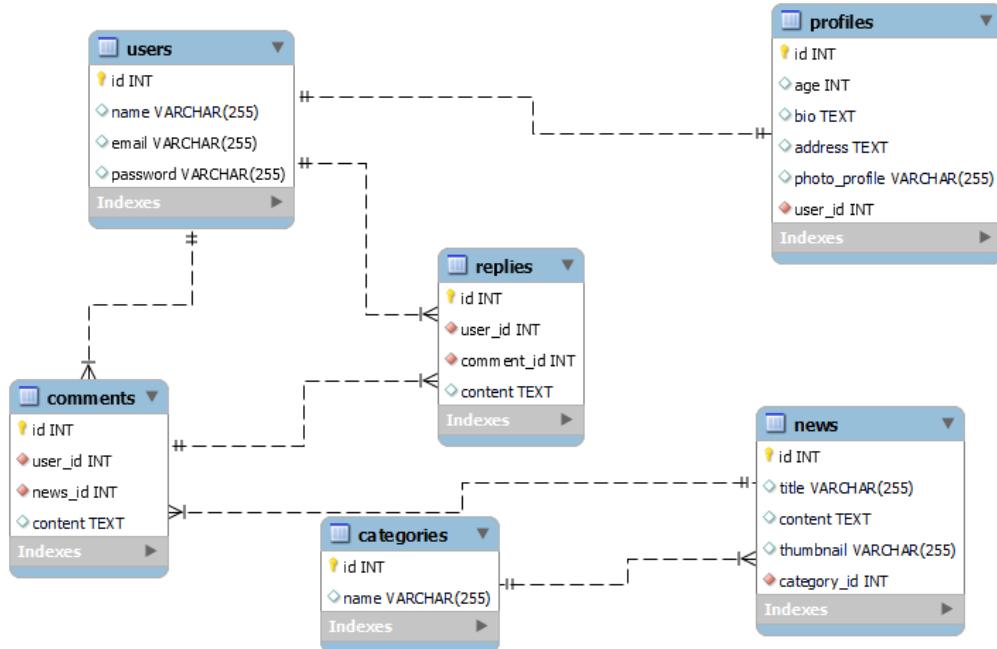
- Header:** Server: 127.0.0.1, Basis data, SQL, Status, Akun pengguna, Eksport, Impor, P.
- Left Sidebar:** Terbaru, Favorit, Baru, article, information_schema, karyawan42, library43, mysql, performance_schema, phpmyadmin.
- Basis data Section:** Buat basis data, portalberita, utf8mb4_general_ci, Buat.
- Buttons at the bottom:** Pilih Semua, Hapus.

maka database kita akan terbuat di sebelah sidebar kiri di list database phpmyadmin



disini kita akan membuat beberapa table menggunakan menggunakan fitur migration di laravel berikut adalah tabel-tabel yang akan kita buat beserta relasi tablenya

deskripsi table



tabel users

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
name	string	
email	string	unique
password	string	
timestamps		

table categories

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
name	string	
timestamps		

tabel profiles

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
age	integer	
bio	text	
address	text	
photo_profile	string	nullable
user_id	big integer	foreign key references table users column id unsigned

timestamps		
------------	--	--

tabel news

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
title	integer	
content	text	
thumbnail	text	
category_id	big integer	foreign key references table categories column id unsigned
timestamps		

tabel comments

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
news_id	big integer	foreign key references table news column id unsigned
user_id	big integer	foreign key references table users column id unsigned
content	text	
timestamps		

tabel replies

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned

comment_id	big integer	foreign key references table comments column id unsigned
user_id	big integer	foreign key references table users column id unsigned
content	text	
timestamps		

Buat Migration

disini kita akan membuat migration dari disini untuk table users kita pake saja file bawaan dari laravelnya yang berada di folder database/migrations

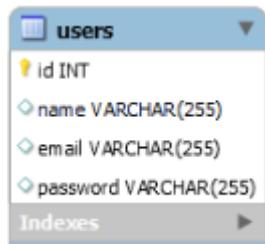
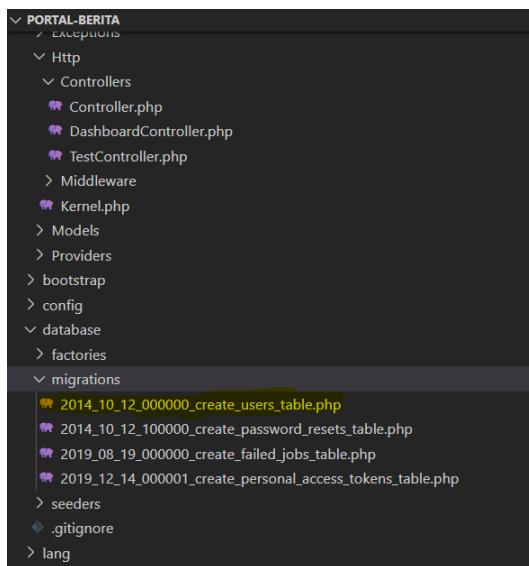


table users

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
name	string	
email	string	unique
password	string	
timestamps		

Migration Users

```
<?php

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

return new class extends Migration
{
    public function up()
    {
        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();
        });
    }
}
```

```
public function down()
{
    Schema::dropIfExists('users');
}

};
```

setelah itu jalankan migrationnya

```
php artisan migrate
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate
INFO  Preparing database.
Creating migration table ..... 136ms DONE
INFO  Running migrations.
2014_10_12_000000_create_users_table ..... 83ms DONE
2014_10_12_100000_create_password_resets_table ..... 30ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 27ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 43ms DONE
```

disini kita bisa mengecek status dari migration nya apakah sudah jalan atau tidak

```
php artisan migrate:status
```

```
PDO\Pi\construct(mysql:host=127.0.0.1;port=3306;dbname=portalberita , root , , [])
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate:status
Migration name ..... Batch / Status
2014_10_12_000000_create_users_table ..... [1] Ran
2014_10_12_100000_create_password_resets_table ..... [1] Ran
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran
```

di gambar diatas status dari semua filenya sudah run dan masuk ke batch 1 yaitu yang pertama kali kita migrate

migration yang sudah run akan masuk kedalam databasenya

di table databasenya kita juga mendapat satu table yaitu migration yaitu table yang mencatat file apa saja di laravel kita yang sudah jalan fungsi seputin php artisan migrate:status

setelah kita berhasil membuat melakukan migrate pada table users kemudian kita akan lanjut ke table yang tidak memiliki kolom foreign key dulu yaitu table categories

table categories

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
name	string	
timestamps		

pertama-tama kita buat dulu migrationnya

```
php artisan make:migration create_categories_table
```

kemudian kita coba edit file didalam migration categories yang sudah dibuat

Migration Categories

```
<?php

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

return new class extends Migration
{
    public function up()
    {
        Schema::create('categories', function (Blueprint $table)
        {
            $table->id();
            $table->string('name');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('categories');
    }
};
```

terakhir kita tinggal menjalankan migration nya dan check statusnya

```
php artisan migrate
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate
[INFO] Running migrations.
2023_03_23_061052_create_categories_table ..... 13ms DONE
```

php artisan migrate:status

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate:status
Migration name ..... Batch / Status
2014_10_12_000000_create_users_table ..... [1] Ran
2014_10_12_100000_create_password_resets_table ..... [1] Ran
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran
2023_03_23_061052_create_categories_table ..... [2] Ran
```

pada gambar diatas migration categories berhasil run dan masuk ke batch 2 ter migrate ke antrian ke 2

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	<code>id</code>	bigint(20)		UNSIGNED	Tidak	<i>Tidak ada</i>		AUTO_INCREMENT	Ubah Hapus Lainnya
2	<code>name</code>	varchar(255)	utf8mb4_unicode_ci		Tidak	<i>Tidak ada</i>			Ubah Hapus Lainnya
3	<code>created_at</code>	timestamp			Ya	<i>NULL</i>			Ubah Hapus Lainnya
4	<code>updated_at</code>	timestamp			Ya	<i>NULL</i>			Ubah Hapus Lainnya

setelah kita membuat table categories dan users selanjutnya kita akan membuat table yang terforeign key atau references ke table yang sudah kita buat yaitu news dan profiles

news	profiles
<code>id</code> INT	<code>id</code> INT
<code>title</code> VARCHAR(255)	<code>age</code> INT
<code>content</code> TEXT	<code>bio</code> TEXT
<code>thumbnail</code> VARCHAR(255)	<code>address</code> TEXT
<code>category_id</code> INT	<code>photo_profile</code> VARCHAR(255)
<code>user_id</code> INT	<code>user_id</code> INT

tabel profiles

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
age	integer	
bio	text	
address	text	
photo_profile	string	nullable
user_id	big integer	foreign key references table users column id unsigned
timestamps		

tabel news

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
title	integer	
content	text	
thumbnail	text	
category_id	big integer	foreign key references table categories column id unsigned
timestamps		

seperti sebelumnya kita buat migrationnya dulu

```
php artisan make:migration create_profiles_table
```

```
php artisan make:migration create_news_table
```

setelah itu kita edit code di migration profile dan news

migration profiles

```
<?php

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

return new class extends Migration
{
    public function up()
    {
        Schema::create('profiles', function (Blueprint $table) {
            $table->id();
            $table->integer('age');
            $table->text('bio');
            $table->text('address');
            $table->string('photo_profile')->nullable();
            $table->unsignedBigInteger('user_id');

            $table->foreign('user_id')->references('id')->on('users');
            $table->timestamps();
        });
    }

    public function down()
    {
```

```
    Schema::dropIfExists('profiles');

}

};
```

Migration news

```
<?php

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

return new class extends Migration
{
    public function up()
    {
        Schema::create('news', function (Blueprint $table) {
            $table->id();
            $table->string('title');
            $table->text('content');
            $table->string('thumbnail');
            $table->unsignedBigInteger('category_id');
            $table->foreign('category_id')->references('id')->on('categories');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('news');
    }
};
```

kemudian kita jalankan migrate dan check status migratennya

```
php artisan migrate
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate
[INFO] Running migrations.
2023_03_23_062502_create_profiles_table ..... 52ms DONE
2023_03_23_062514_create_news_table ..... 43ms DONE
```

```
php artisan migate:status
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate:status
Migration name ..... Batch / Status
2014_10_12_000001_create_users_table ..... [1] Ran
2014_10_12_100001_create_password_resets_table ..... [1] Ran
2019_08_19_000001_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran
2023_03_23_061052_create_categories_table ..... [2] Ran
2023_03_23_062502_create_profiles_table ..... [3] Ran
2023_03_23_062514_create_news_table ..... [3] Ran
```

pada gambar diatas migration profile dan news berhasil run dan masuk ke batch 3 atau termigrate ke antrian ke 3

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1
- Database:** portaberita
- Table:** profiles
- Structure:** Shows the columns of the 'profiles' table.
- Columns:**
 - id:** bigint(20), UNSIGNED, Tidak, Tidak ada, AUTO_INCREMENT, Tindakan: Ubah, Hapus, Lainnya
 - age:** int(11), Tidak, Tidak ada, Tindakan: Ubah, Hapus, Lainnya
 - bio:** text, utf8mb4_unicode_ci, Tidak, Tidak ada, Tindakan: Ubah, Hapus, Lainnya
 - address:** text, utf8mb4_unicode_ci, Tidak, Tidak ada, Tindakan: Ubah, Hapus, Lainnya
 - photo_profile:** varchar(255), utf8mb4_unicode_ci, Ya, NULL, Tindakan: Ubah, Hapus, Lainnya
 - user_id:** bigint(20), UNSIGNED, Tidak, Tidak ada, Tindakan: Ubah, Hapus, Lainnya
 - created_at:** timestamp, Ya, NULL, Tindakan: Ubah, Hapus, Lainnya
 - updated_at:** timestamp, Ya, NULL, Tindakan: Ubah, Hapus, Lainnya

The screenshot shows the 'Struktur tabel' (Table Structure) page in phpMyAdmin. The database is 'portalberita' and the table is 'news'. The table structure is as follows:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ter nilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	bigint(20)		unsigned	Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	title	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	content	text	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	thumbnail	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	category_id	bigint(20)		unsigned	Tidak	Tidak ada			Ubah Hapus Lainnya
6	created_at	timestamp			Ya	NULL			Ubah Hapus Lainnya
7	updated_at	timestamp			Ya	NULL			Ubah Hapus Lainnya

setelah kita membuat table news dan profiles selanjutnya kita akan membuat table yang terforeign key atau references ke table news dan users yaitu table comments

The screenshot shows the 'Struktur tabel' (Table Structure) page in phpMyAdmin. The database is 'portalberita' and the table is 'comments'. The table structure is as follows:

	comments			
	id INT	user_id INT	news_id INT	content TEXT
Indexes				

tabel comments

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
news_id	big integer	foreign key references table news column id unsigned
user_id	big integer	foreign key references table users column id unsigned
content	text	
timestamps		

pertama-tama kita buat dulu migrationnya

```
php artisan make:migration create_comments_table
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan make:migration create_comments_table
[INFO] Migration [D:\ebook\im-ebook-portal-berita\portal-berita\database\migrations\2023_03_23_064802_create_comments_table.php] created successfully.
```

setelah itu kita edit code di migration comments

Migration Comments

```
<?php

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

return new class extends Migration
{
    public function up()
    {
        Schema::create('comments', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('news_id');

            $table->foreign('news_id')->references('id')->on('news');
            $table->unsignedBigInteger('user_id');

            $table->foreign('user_id')->references('id')->on('users');
            $table->text('content');
            $table->timestamps();
        });
    }

    public function down()
    {
```

```

Schema::dropIfExists('comments');

}

} ;

```

kemudian kita jalankan migrate dan check status migratanya

```
php artisan migrate
```

```

PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate
INFO  Running migrations.
2023_03_23_064802_create_comments_table ..... 84ms DONE

```

```
php artisan migrate:status
```

```

PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate:status
Migration name ..... Batch / Status
2014_10_12_000000_create_users_table ..... [1] Ran
2014_10_12_100000_create_password_resets_table ..... [1] Ran
2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran
2023_03_23_061052_create_categories_table ..... [2] Ran
2023_03_23_062502_create_profiles_table ..... [3] Ran
2023_03_23_062514_create_news_table ..... [3] Ran
2023_03_23_064802_create_comments_table ..... [4] Ran

```

pada gambar diatas migration comments berhasil run dan masuk ke batch 4 atau ter migrate ke antrian ke 4

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Eksra	Tindakan
1	<code>id</code>	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	<code>news_id</code>	bigint(20)		UNSIGNED	Tidak	Tidak ada			Ubah Hapus Lainnya
3	<code>user_id</code>	bigint(20)		UNSIGNED	Tidak	Tidak ada			Ubah Hapus Lainnya
4	<code>content</code>	text		utf8mb4_unicode_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
5	<code>created_at</code>	timestamp			Ya	NULL			Ubah Hapus Lainnya
6	<code>updated_at</code>	timestamp			Ya	NULL			Ubah Hapus Lainnya

setelah kita membuat table comments selanjutnya kita akan membuat table yang terforeign key atau references ke table comments dan users yaitu table replies

replies	
!	id INT
◆	user_id INT
◆	comment_id INT
◆	content TEXT
Indexes	

tabel replies

kolom	type data	extra
id	big integer	Primary key auto_increment unsigned
comment_id	big integer	foreign key references table comments column id unsigned
user_id	big integer	foreign key references table users column id unsigned
content	text	
timestamps		

pertama-tama seperti sebelum sebelumnya kita buat dulu migrationnya

```
php artisan make:migration create_replies_table
```

```
ps D:\ebook\im-ebook-portal-berita\portal-berita> php artisan make:migration create_replies_table
[INFO] Migration [D:\ebook\im-ebook-portal-berita\portal-berita\database\migrations\2023_03_23_070038_create_replies_table.php] created successfully.
```

setelah itu kita edit code di migration replies

Migration Replies

```
<?php

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

return new class extends Migration
{
    public function up()
    {
        Schema::create('replies', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('comment_id');

            $table->foreign('comment_id')->references('id')->on('comments');
            $table->unsignedBigInteger('user_id');

            $table->foreign('user_id')->references('id')->on('users');
            $table->text('content');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('replies');
    }
};
```

kemudian kita jalankan migrate dan check status migratennya

```
php artisan migrate
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate
[INFO] Running migrations.
  2023_03_23_070038_create_replies_table ..... 74ms DONE
```

```
php artisan migrate:status
```

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan migrate:status
Migration name ..... Batch / Status
2014_10_12_000001_create_users_table ..... [1] Ran
2014_10_12_100001_create_password_resets_table ..... [1] Ran
2019_08_19_000001_create_failed_jobs_table ..... [1] Ran
2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran
2023_03_23_061052_create_categories_table ..... [2] Ran
2023_03_23_062502_create_profiles_table ..... [3] Ran
2023_03_23_062514_create_news_table ..... [3] Ran
2023_03_23_064802_create_comments_table ..... [4] Ran
2023_03_23_070038_create_replies_table ..... [5] Ran
```

pada gambar diatas migration replies berhasil run dan masuk ke batch 5 atau ter migrate ke antrian ke 5

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1
- Database:** portalberita
- Table:** replies
- Structure Tab:** Selected
- Columns:**

#	Nama	Jenis	Penyortiran	Atribut	Tak Ter nilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	comment_id	bigint(20)		UNSIGNED	Tidak	Tidak ada			Ubah Hapus Lainnya
3	user_id	bigint(20)		UNSIGNED	Tidak	Tidak ada			Ubah Hapus Lainnya
4	content	text	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	created_at	timestamp			Ya	NULL			Ubah Hapus Lainnya
6	updated_at	timestamp			Ya	NULL			Ubah Hapus Lainnya
- Buttons:** Pilih Semua, Dengan pilihan, Jelajahi, Ubah, Hapus, Utama, Unik, Indeks, Spasial, Teks penuh, Add to central columns, Cetak, Usulkan struktur tabel, Lacak tabel, Move columns, Normalisasi.
- Form:** Tambahan 1 kolom setelah updated_at

setelah kita sudah membuat semua migration table kita akan membahas tentang CRUD menggunakan Query Builder di materi berikutnya

BAB 12 Laravel CRUD Query Builder

CRUD

CRUD (Create, Read, Update, Delete) adalah serangkaian operasi yang sering dilakukan dalam pengembangan web untuk mengelola data dalam basis data. Laravel menyediakan fitur-fitur yang memudahkan pengembang untuk membuat aplikasi CRUD dengan cepat dan mudah.

ada 2 macam CRUD di laravel yaitu dengan menggunakan Query Builder dan eloquent ORM (dengan model)

di materi kali ini kita akan belajar laravel CRUD menggunakan Query Builder, Laravel menyediakan Query Builder yang memudahkan kita untuk membuat Query database tanpa harus menulis SQL secara langsung. Berikut adalah langkah-langkah untuk membuat operasi CRUD dengan menggunakan Query Builder di Laravel:

1. Create Data (membuat data baru)

Untuk membuat data baru, gunakan fungsi **insert()** pada Query Builder dengan parameter array yang berisi data yang ingin dibuat, seperti contoh berikut:

```
DB::table('nama_tabel')->insert([
    'nama_kolom_1' => 'nilai_1',
    'nama_kolom_2' => 'nilai_2',
    // dan seterusnya
]);
```

2. Read Data (mengambil data)

Untuk mengambil data, gunakan fungsi **get()** atau **first()** pada Query Builder, seperti contoh berikut:

```
$data = DB::table('nama_tabel')->get(); // mengambil semua data

$data = DB::table('nama_tabel')->where('nama_kolom',
```

```
'nilai')->get();  
// mengambil data dengan kondisi tertentu  
  
$data = DB::table('nama_tabel')->orderBy('nama_kolom',  
'asc')->get();  
// mengambil data dengan urutan tertentu  
  
$data = DB::table('nama_tabel')->select('nama_kolom_1',  
'nama_kolom_2')->get(); // mengambil data dengan kolom  
tertentu  
  
$data = DB::table('nama_tabel')->where('nama_kolom',  
'nilai')->first();  
// mengambil data pertama dengan kondisi tertentu
```

3. Update data (mengubah data)

Untuk mengubah data, gunakan fungsi **update()** pada Query Builder dengan parameter array yang berisi data yang ingin diubah dan kondisi untuk data yang ingin diubah, seperti contoh berikut:

```
DB::table('nama_tabel')->where('nama_kolom',  
'nilai')->update([  
    'nama_kolom_1' => 'nilai_baru_1',  
    'nama_kolom_2' => 'nilai_baru_2',  
    // dan seterusnya  
]);
```

4. Delete data (menghapus data)

Untuk menghapus data, gunakan fungsi **delete()** pada Query Builder dengan kondisi untuk data yang ingin dihapus, seperti contoh berikut:

```
DB::table('nama_tabel')->where('nama_kolom',  
'nilai')->delete();
```

Dengan Query Builder di Laravel, Anda dapat membuat operasi CRUD dengan mudah dan efisien tanpa harus menulis kueri SQL secara manual.

Laravel Validation

Laravel menyediakan fitur validasi yang memudahkan pengembang untuk memeriksa data masukan dari pengguna sebelum disimpan ke dalam basis data. Validasi ini dapat dilakukan dengan cepat dan mudah menggunakan fitur validasi bawaan Laravel.

beberapa rules yang sering digunakan di validation

- required: kolom wajib diisi
- string: kolom harus berupa string
- numeric: kolom harus berupa angka
- email: kolom harus berupa alamat email yang valid
- unique:nama_tabel,nama_kolom: kolom harus unik di dalam tabel
- min:value: panjang string atau nilai numerik harus setidaknya value

Latihan

sebelum kita mangatur CRUD categories menambahkan sedikit kode untuk halaman master.blade.php (tambahkan code yang di warnai dengan warna biru)

resources/views/layouts/**master.blade.php**

```
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
    <title>Portal News</title>
    <link href="{{asset('template/css/media_query.css')}}" rel="stylesheet" type="text/css"/>
```

```

<link href="{{asset('template/css/bootstrap.css')}}" rel="stylesheet" type="text/css"/>
<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet" integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAF1NEevOEH3S10sibVcOQVnN" crossorigin="anonymous">
<link href="{{asset('template/css/animate.css')}}" rel="stylesheet" type="text/css"/>
<link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet">
<link href="{{asset('template/css/owl.carousel.css')}}" rel="stylesheet" type="text/css"/>
<link href="{{asset('template/css/owl.theme.default.css')}}" rel="stylesheet" type="text/css"/>
<!-- Bootstrap CSS -->
<link href="{{asset('template/css/style_1.css')}}" rel="stylesheet" type="text/css"/>
<!-- Modernizr JS -->
<script src="{{asset('template/js/modernizr-3.5.0.min.js')}}"></script>
</head>
<body>

{{-- Header --}}
<div class="container-fluid bg-faded fh5co_padd_mediya padding_786">
<div class="container padding_786">
<nav class="navbar navbar-toggleable-md navbar-light ">
<button class="navbar-toggler navbar-toggler-right mt-3" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"><span class="fa fa-bars"></span></button>

```

```

        <a class="navbar-brand" href="#">![img](images/logo.png)></a>
        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <ul class="navbar-nav mr-auto">
                <li class="nav-item ">
                    <a class="nav-link" href="/">Home <span
class="sr-only">(current)</span></a>
                </li>
                <li class="nav-item ">
                    <a class="nav-link" href="/about">About
<span class="sr-only">(current)</span></a>
                </li>
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle"
href="#" id="dropdownMenuButton2" data-toggle="dropdown"
aria-haspopup="true"
aria-expanded="false">Kategori <span
class="sr-only">(current)</span></a>
                    <div class="dropdown-menu"
aria-labelledby="dropdownMenuLink_1">
                        <a class="dropdown-item"
href="/categories">Tampil Kategori</a>
                        <a class="dropdown-item"
href="/categories/categories">Tambah Kategori</a>
                    </div>
                </li>
            </ul>
        </div>
    </div>
{{-- End Header --}}
{{-- Content --}}
<div class="container-fluid pb-4 pt-4 paddding">
```

```

@yield('content')
</div>
{{-- End Content --} }

{{-- Footer --}}
<div class="container-fluid fh5co_footer_right_reserved">
<div class="container">
<div class="row ">
<div class="col-12 col-md-6 py-4 Reserved"> ©
Copyright 2018, All rights reserved. Design by <a
href="https://freehtml5.co" title="Free HTML5 Bootstrap
templates">FreeHTML5.co</a>. </div>
</div>
</div>
</div>
{{-- End Footer --} }

<div class="gototop js-top">
<a href="#" class="js-gotop"><i class="fa
fa-arrow-up"></i></a>
</div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.m
in.js"></script>
<script
src="{{asset('template/js/owl.carousel.min.js')}}"></script>
<!--<script
src="https://code.jquery.com/jquery-3.1.1.slim.min.js"
integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k90mg/a/EheAdg
tzNs3hpfrag6Ed950n" crossorigin="anonymous"></script>-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tethe
r.min.js"

integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8ffjk5JAIXUYHKk

```

```

Dx6Qin1DkWx51bBrb"
    crossorigin="anonymous">></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js"

integrity="sha384-vBWWz1ZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVkJ+jo0ieGizqPLForn"
    crossorigin="anonymous">></script>
<!-- Waypoints -->
<script
src="{{asset('template/js/jquery.waypoints.min.js')}}"></script>
<!-- Main -->
<script src="{{asset('template/js/main.js')}}"></script>

</body>
</html>

```

Create Data

disini kita akan melakukan CRUD pada table categories yang ada di databasenya, pertama tama kita akan dengan huruf C dari CRUD yaitu Create Data dari table categories

pertama tama kita buat dulu route dan juga controlernya

```
php artisan make:controller CategoriesController
```

```

PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan make:controller CategoriesController
[INFO] Controller [D:\ebook\im-ebook-portal-berita\portal-berita\app\Http\Controllers\CategoriesController.php] created successfully.

```

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');

//Create Data Category
Route::get('/categories/categories', [CategoriesController::class,
'create'])->name('category.create');
Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');
```

code diatas kita memanggil file controller categories menggunakan use, use

App\Http\Controllers\CategoriesController; kemudian terdapat 2 route,

1. route yang pertama berfungsi mengarahkan kita ke CategoriesController fungsi create dengan method get,
2. fungsi route kedua mengarahkan kita ke CategoriesController fungsi store dengan method post

app\Http\Controllers\CategoriesController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CategoriesController extends Controller
{
    public function create()
    {
        return view('category.tambah');
    }

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

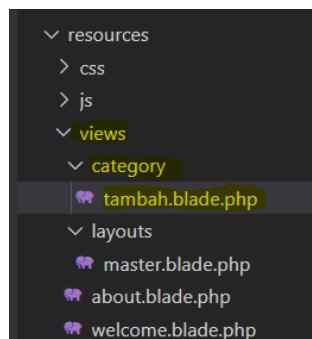
        DB::table('categories')->insert([
            'name' => $request->input('name'),
        ]);

        return redirect('/categories')->with('success', 'Data berhasil ditambahkan');
    }
}
```

pada code di atas kita memanggil fungsi dari database Buildernya use Illuminate\Support\Facades\DB; kemudian kita membuat 2 fungsi

1. fungsi pertama create untuk mengarahkan kita ke file view sub folder category file tambah.blade.php .

- fungsi kedua untuk melakukan validasi dari inputan name dengan rules required dan max:255 yaitu maksimal karakter yang dimasukan tidak boleh lebih dari 255 karakter kemudian melakukan insert data pada table categories berdasarkan request input name pada tampilan form viewnya setelah berhasil melakukan insert data ke database maka akan di arahkan ke halaman /categories dengan pesan singkat menggunakan flash message



resources/views/category/**tambah.blade.php**

```

@extends('layouts.master')

@section('content')



# Add Categories


<form action="/categories" method="post">
  @csrf
  <div class="form-group">
    <label for="name">Category Name</label>
    <input type="text" class="form-control" id="name" name="name">
  </div>
  @error('name')
    <div class="alert alert-danger">{{ $message }}</div>
  @enderror
  <button type="submit" class="btn btn-primary">Tambah</button>


```

```

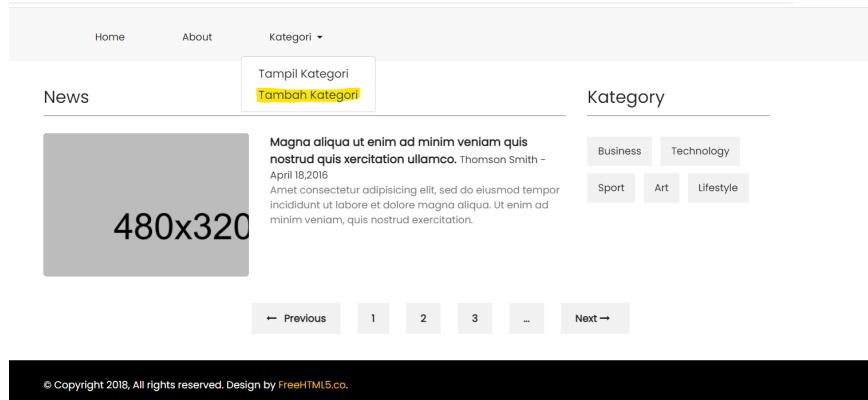
</form>

</div>

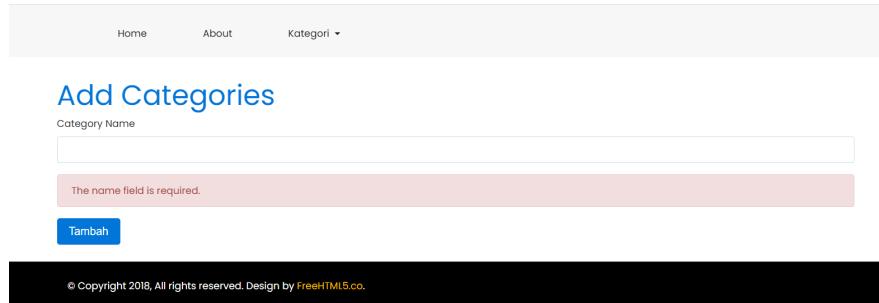
@endsection

```

di form kita memasukan action /kategori dengan method post sesuai dengan route yang sudah di daftarkan di route web.php, @csrf digunakan untuk proteksi csrf , dan juga @error('name') yang berfungsi untuk menampilkan error validasinya kemudian juga kita membuat attribute name dengan value name berfungsi untuk menerima request inputan dari controllernya



tampilan validasi dengan error validasi jika data tidak diisi kemudian diklik submit



tampilan pada saat berhasil submit

```
Stack Context Share DOCS
Symfony\Component\HttpKernel\Exception\MethodNotAllowedHttpException
The GET method is not supported for route categories. Supported methods: POST.

Collapse vendor frames
 Illuminate\Routing\AbstractRouteCollection::122 requestMethodNotAllowed()
 Illuminate\Routing\AbstractRouteCollection::107 getRouteForMethods()
 Illuminate\Routing\AbstractRouteCollection::41 handleMatchedRoute()
 Illuminate\Routing\RouteCollection::162 match()
 Illuminate\Routing\Router::753

D:\ebook\im-ebook-portal-berita\portal-berita\vendor\laravel\framework\src\Illuminate\Routing\AbstractRouteCollection.php
122     $this->requestMethodNotAllowed($request, $methods, $request->method());
107     /**
111      * Throw a method not allowed HTTP exception.
112      *
113      * @param \Illuminate\Http\Request $request
114      * @param array $others
115      * @param string $method
116      * @return void
117      */
118      * @throws \Symfony\Component\HttpKernel\Exception\MethodNotAllowedHttpException
PHP 8.1.12 9.52.4
```

pada saat kita submit kita akan mendapatkan error seperti diatas karena kita belum membuat route untuk /categories dengan method get, tapi data kita berhasil masuk ke database

Server: 127.0.0.1 > Database: portalberita > Tabel: categories

Jelajahi Struktur SQL Cari Tambahan Eksport Impor Hak Akses O

Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0,0005 detik.)

SELECT * FROM `categories`

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini

Extra options

	Ubah	Salin	Hapus	1	Traveler	NULL	NULL
Pilih Semua	Dengan pilihan:	Ubah	Salin	Hapus	Eksport		

Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini

Operasi hasil kueri

Cetak Salin ke clipboard Eksport Tampilkan bagan Buat tampilan

disini kita akan lanjut untuk membuat halaman Read data untuk mengatasi error diatas

Read Data

selelah kita berhasil Create data sekarang kita masuk ke dalam R di CRUD yaitu Read data disini kita akan menampilkan data di kolom database kedalam viewnya pertama - tama kita atur dulu routenya

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');

//Create Data Category
Route::get('/categories/categories',
[CategoriesController::class,
'create'])->name('category.create');

Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');

//Read Data Category
Route::get('/categories', [CategoriesController::class,
'index'])->name('category.index');
```

di code kita menambahkan satu route yang berfungsi mengarahkan kita ke CategoriesController fungsi index dengan method get,

kemudian kita atur Controllernya

app\Http\Controllers\CategoriesController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CategoriesController extends Controller
{
    public function create()
    {
        return view('category.tambah');
    }

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

        DB::table('categories')->insert([
            'name' => $request->input('name'),
        ]);

        return redirect('/categories');
    }

    public function index()
    {
        $categories = DB::table('categories')->get();

        return view('category.tampil', ['categories' =>
$categories]);
    }
}
```

```
}
```

pada code diatas kita membahakan code untuk mengambil semua data di table categories yang di tampung kedalam \$categories yang akan kita passing ke halaman view **tampil.blade.php** yang berada di sub folder category

terakhir kita akan menampilkan didalam viewnya

resources/views/category/tampil.blade.php

```
@extends('layouts.master')

@section('content')

<div class="container padding">
    <h1 class="text-primary my-3">Categories</h1>

    @if(session('success'))
        <div class="alert alert-success">
            {{ session('success') }}
        </div>
    @endif

    <table class="table table-bordered">
        <thead class="thead-dark">
            <tr>
                <th scope="col">#</th>
                <th scope="col">Name</th>
                <th scope="col">Action</th>
            </tr>
        </thead>
        <tbody>
            @forelse ($categories as $key => $item)
                <tr>
                    <th scope="row">{{ $key + 1 }}</th>
                    <td>{{ $item->name }}</td>
```

```

<td></td>
</tr>
@empty
<tr>
    <td colspan="3">Tidak ada Data</td>
</tr>
@endforelse

</tbody>
</table>
</div>

@endsection

```

code diatas kita melakukan looping forelse pada data yang sudah kita passing \$categories di controllernya \$key untuk menampung key dari data nya yang biasanya dimulai dari index 0 dan \$item adalah satuan dari datanya jika datanya ada dia akan menjalan code di dalam @forelse di kolom pertama kita menampilkan \$key + 1 untuk nomor urut, di kolom kedua kita manampilkan nama \$item->nama dan di kolom ketiga yang masih kosong akan kita isi nantinya dengan tombol aksi edit dan delete tapi jika datanya kosong dia akan menjalankan @empty

di code di atas kita juga membuat kondisi @if(session('success')) untuk menampilkan pesan alert dari flash message function store di CategoriesController

tampilan tampil data <http://localhost:8000/categories>

The screenshot shows a web page titled "Categories". At the top, there is a navigation bar with links for "Home", "About", and "Kategori ▾". Below the title, there is a table with three columns: "#", "Name", and "Action". A single row is present in the table, showing "#1" and "Name: Traveler". The "Action" column contains a small icon. At the bottom of the page, there is a footer bar with the text "© Copyright 2018, All rights reserved. Design by FreeHTML5.co."

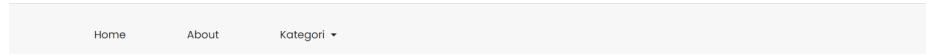
#	Name	Action
1	Traveler	

disini kita akan coba menambah satu data lagi di database nya

The screenshot shows a web page titled "Categories". At the top, there is a navigation bar with links for "Home", "About", and "Kategori ▾". Below the title, there is a table with three columns: "#", "Name", and "Action". A single row is present in the table, showing "#1" and "Name: Traveler". To the right of the table, there is a yellow box highlighting a button labeled "Tambah Kategori". At the bottom of the page, there is a footer bar with the text "© Copyright 2018, All rights reserved. Design by FreeHTML5.co."

#	Name	Action
1	Traveler	

The screenshot shows a web page titled "Add Categories". At the top, there is a navigation bar with links for "Home", "About", and "Kategori ▾". Below the title, there is a form with a label "Category Name" and a text input field containing "Sport". Below the input field is a blue "Tambah" button. At the bottom of the page, there is a footer bar with the text "© Copyright 2018, All rights reserved. Design by FreeHTML5.co."



Categories

Data berhasil ditambahkan

#	Name	Action
1	Traveler	
2	Sport	

pada saat kita berhasil submit maka error sebelumnya akan hilang dan flash message akan tampil di bagian tampil datanya dan juga data nya berhasil tersimpan di database

berikutnya kita akan membuat untuk Update data

Update Data

selelah kita berhasil Create data dan Read Data sekarang kita masuk ke dalam U di CRUD yaitu Update data disini kita akan membuat dua Route 2 route untuk mengarah ke form edit dengan membawa data berdasarkan id dan route yang kedua untuk update data berdasarkan id nya

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');

//Create Data Category
```

```

Route::get('/categories/categories',
[CategoriesController::class,
'create'])->name('category.create');

Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');

//Read Data Category
Route::get('/categories', [CategoriesController::class,
'index'])->name('category.index');

//Update Data Category
Route::get('/categories/{id}/edit', [CategoriesController::class,
'edit'])->name('category.edit');

Route::put('/categories/{id}', [CategoriesController::class,
'update'])->name('category.update');

```

code diatas terdapat 2 route,

1. route yang pertama berfungsi mengarahkan kita ke CategoriesController fungsi edit dengan method get dengan membawa parameter id,
2. fungsi route kedua mengarahkan kita ke CategoriesController fungsi update dengan method put dengan membawa parameter id

app\Http\Controllers\CategoriesController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CategoriesController extends Controller
{
    public function create()

```

```

    {

        return view('category.tambah');

    }

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

    DB::table('categories')->insert([
        'name' => $request->input('name'),
    ]);

    return redirect('/categories')->with('success', 'Data berhasil ditambahkan');
}

public function index()
{
    $categories = DB::table('categories')->get();

    return view('category.tampil', ['categories' => $categories]);
}

public function edit($id)
{
    $category = DB::table('categories')->where('id', $id)->first();

    return view('category.edit', ['category' => $category]);
}

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

```

```

    $request->validate([
        'name' => 'required|max:255',
    ]);

    DB::table('categories')
        ->where('id', $id)
        ->update(
            [
                'name' => $request->input('name'),
            ]
        );
    }

    return redirect('/categories')->with('success', 'Data
berhasil Diedit');
}

}

```

pada code di atas kita memanggil fungsi dari database Buildernya use Illuminate\Support\Facades\DB; kemudian kita membuat 2 fungsi

1. fungsi pertama edit untuk mengarahkan kita ke file view sub folder category file edit.blade.php dengan melempar data \$category yang dihasilkan dari mengambil data dari table categories dimana memiliki kondisi id nya sama dengan parameter \$id yang ada di route parameter
2. fungsi kedua untuk melakukan validasi dari inputan name dengan rules required dan max:255 yaitu maksimal karakter yang dimasukan tidak boleh lebih dari 255 karakter kemudian melakukan update data pada table categories berdasarkan request input name dengan kondisi dimana paramater id sama dengan \$id pada tampilan form viewnya setelah berhasil melakukan insert data ke database maka akan di arahkan ke halaman /categories dengan pesan singkat menggunakan flash message

resources/views/category/**tampil.blade.php**

```
@extends('layouts.master')

@section('content')

<div class="container padding">
    <h1 class="text-primary my-3">Categories</h1>

    @if(session('success'))
        <div class="alert alert-success">
            {{ session('success') }}
        </div>
    @endif

    <table class="table table-bordered">
        <thead class="thead-dark">
            <tr>
                <th scope="col">#</th>
                <th scope="col">Name</th>
                <th scope="col">Action</th>
            </tr>
        </thead>
        <tbody>
            @forelse ($categories as $key => $item)
                <tr>
                    <th scope="row">{{ $key + 1 }}</th>
                    <td>{{ $item->name }}</td>
                    <td><a href="/categories/{{ $item->id }}/edit"
class="btn btn-info btn-sm">Edit</a></td>
                </tr>
            @empty
                <tr>
                    <td colspan="3">Tidak ada Data</td>
                </tr>
            @endfor
        </tbody>
    </table>
</div>
```

```

        @endforelse

    </tbody>
</table>
</div>

@endsection

```

di halaman view tampil.blade.php kita membuat satu tombol di aksi ya sesuai route yang sudah kita daftarkan di web.php `/categories/{{ $item->id}}/edit` untuk mengarah ke halaman edit berdasarkan `{{$item->id}}`

#	Name	Action
1	Traveler	<button>Edit</button>
2	Sport	<button>Edit</button>

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

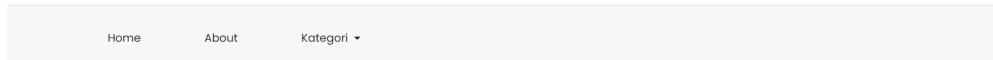
resources/views/category/**edit.blade.php**

```
@extends('layouts.master')

@section('content')
<div class="container padding">
    <h1 class="text-primary my-2">Edit Categories</h1>
    <form action="/categories/{{ $category->id }}" method="post">
        @csrf
        @method('put')
        <div class="form-group">
            <label for="name">Category Name</label>
            <input type="text" value="{{ $category->name }}"
class="form-control" id="name" name="name">
        </div>
        @error('name')
            <div class="alert alert-danger">{{ $message }}</div>
        @enderror
        <button type="submit" class="btn
btn-primary">Update</button>
    </form>
</div>
@endsection
```

pada code diatas kita membuat action dari form nya sesuai dengan url yang sudah didaftarkan di web.php `/categories/{{ $category->id }}` dengan `@method('put')` dan juga kita melakukan protect `@csrf` dan di inputan kita juga menambahkan `value="{{ $category->id }}"` untuk mengisi inputan yang sudah kita passing berdasarkan kondisi di controllernya

tampilan pada saat tombol edit pada kolom traveler diklik



Edit Categories

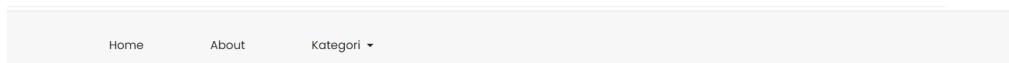
Category Name

Traveler

Update

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

disini kita akan update data Traveler menjadi Culinary



Edit Categories

Category Name

Culinary

Update

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

pada saat tombol update di klik

Data berhasil Diedit

#	Name	Action
1	Culinary	Edit
2	Sport	Edit

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

kategori traveller dan berubah menjadi Culinary

Delete Data

disini kita masuk ke fitur terakhir dari CRUD yaitu Delete data

pertama-tama kita atur dulu routenya

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');
```

```

//Create Data Category
Route::get('/categories/categories',
[CategoriesController::class,
'create'])->name('category.create');

Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');

//Read Data Category
Route::get('/categories', [CategoriesController::class,
'index'])->name('category.index');

//Update Data Category
Route::get('/categories/{id}/edit', [CategoriesController::class,
'edit'])->name('category.edit');

Route::put('/categories/{id}', [CategoriesController::class,
'update'])->name('category.update');

//Delete Data Category
Route::delete('/categories/{id}', [CategoriesController::class,
'destroy'])->name('category.destroy');

```

code diatas kita membuat route yang berfungsi mengarahkan kita ke CategoriesController fungsi destroy dengan method delete dengan membawa parameter id,

app\Http\Controllers\CategoriesController.php

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CategoriesController extends Controller

```

```

{
    public function create()
    {
        return view('category.tambah');
    }

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

        DB::table('categories')->insert([
            'name' => $request->input('name'),
        ]);

        return redirect('/categories')->with('success', 'Data berhasil ditambahkan');
    }

    public function index()
    {
        $categories = DB::table('categories')->get();

        return view('category.tampil', ['categories' => $categories]);
    }

    public function edit($id)
    {
        $category = DB::table('categories')->where('id', $id)->first();

        return view('category.edit', ['category' => $category]);
    }
}

```

```

public function update(Request $request, $id)
{
    $request->validate([
        'name' => 'required|max:255',
    ]);

    DB::table('categories')
        ->where('id', $id)
        ->update(
            [
                'name' => $request->input('name'),
            ]
        );
}

return redirect('/categories')->with('success', 'Data berhasil Diedit');
}

public function destroy($id)
{
    DB::table('categories')->where('id', $id)->delete();

    return redirect('/categories')->with('success', 'Data berhasil Didelete');
}

```

code diatas kita menambahkan fungsi destroy dengan parameter id sesuai dimana di dalam function kita membuat fungsi untuk menghapus data berdasarkan parameter \$id kemudian redirect ke halaman /categories dengan membawa flash message

resources/views/category/tampil.blade.php

```
@extends('layouts.master')

@section('content')
<div class="container padding">
    <h1 class="text-primary my-3">Categories</h1>

    @if(session('success'))
        <div class="alert alert-success">
            {{ session('success') }}
        </div>
    @endif

    <table class="table table-bordered">
        <thead class="thead-dark">
            <tr>
                <th scope="col">#</th>
                <th scope="col">Name</th>
                <th scope="col">Action</th>
            </tr>
        </thead>
        <tbody>
            @forelse ($categories as $key => $item)
                <tr>
                    <th scope="row">{{ $key + 1 }}</th>
                    <td>{{ $item->name }}</td>
                    <td>
                        <form action="/categories/{{ $item->id }}" method="post">
                            <a href="/categories/{{ $item->id }}/edit" class="btn btn-info btn-sm">Edit</a>
                            @method('delete')
                            @csrf
                            <input type="submit" class="btn btn-danger btn-sm" value="Delete">
                        </form>
                    </td>
                </tr>
            @empty
                <p>No categories found.</p>
            @endforelse
        </tbody>
    </table>
</div>
```

```

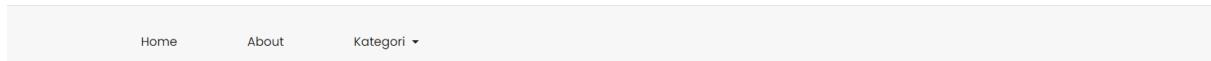
        </form>
    </td>
</tr>
@empty
<tr>
    <td colspan="3">Tidak ada Data</td>
</tr>
@endforelse

</tbody>
</table>
</div>

@endsection

```

pada code diatas kita membuat form submit bukan a href karena method yang kita daftarkan di web.php adalah method delete maka disini kita menggunakan form action dan submit untuk mengubah method di view nya menjadi delete @method('delete') dengan url yang kita simpan di action **action="/categories/{{ \$item->id}}"** yang akan jalan pada saat kita menekan tombol submit



Categories

#	Name	Action
1	Culinary	Edit Delete
2	Sport	Edit Delete

disini kita akan menghapus category culinary
tampilan pada saat tombol delete pada category culinary di klik

Categories

Data berhasil Didelete

#	Name	Action
1	Sport	Edit Delete

pada tampilan diatas category culinary berhasil dihapus

di materi selanjutkan kita masih akan belajar tentang CRUD dengan metode ORM Eloquent yang ada di laravel menggunakan model

BAB 13 Laravel CRUD ORM + File Upload

Apa itu ORM

ORM (Object Relational Mapping) adalah teknik pemetaan objek ke dalam tabel relasional di basis data. Dalam konteks Laravel, ORM digunakan untuk memudahkan pengembang dalam mengakses dan memanipulasi data pada basis data tanpa harus menulis query SQL secara langsung.

Laravel menyediakan fitur ORM bawaan yang kuat dan mudah digunakan, yaitu Eloquent ORM. Dengan Eloquent ORM, Anda dapat dengan mudah memetakan model objek ke dalam tabel relasional di basis data dan melakukan operasi CRUD (create, read, update, delete) pada model tersebut.

Fitur-Fitur Eloquent ORM

1. Membuat model

Membuat model pada Laravel sangat mudah. perintah membuat model

```
php artisan make:model namaModel
```

maka model akan terbuat pada folder app/Models

```
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    protected $table = "products";
    protected $fillable = ['name', 'price'];
}
```

```
protected $table berfungsi untuk mendefinisikan target tablenya  
protected $fillable = [] berfungsi untuk mendaftarkan kolom-kolom apa saja yang  
akan kita manipulasi
```

2. Query Builder

Query Builder adalah cara untuk membuat kueri SQL dalam Laravel dengan sintaks yang lebih mudah dan ekspresif. Query Builder dapat digunakan untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada basis data berapa fungsi yang sering di gunakan seperti save(), find(), all(), get() dll.

3. Relasi

Eloquent ORM memungkinkan Anda untuk mendefinisikan relasi antara model. Ada beberapa jenis relasi, seperti One-to-One, One-to-Many, dan Many-to-Many. Relasi dapat didefinisikan dengan menggunakan metode pada kelas model, seperti hasOne(), hasMany(), dan belongsToMany().

relasi sendiri akan di jelaskan di bab ke #15 Laravel Eloquent relationship

Dengan menggunakan fitur ORM Laravel, Kita dapat dengan mudah melakukan operasi basis data dengan objek dan membuat relasi antara model. Hal ini dapat membantu Kita untuk meningkatkan produktivitas dan konsistensi dalam pengembangan aplikasi web dengan Laravel.

Latihan

di latihan kali ini kita akan membuat CRUD untuk table news menggunakan laravel ORM Eloquent dan melakukan file upload untuk gambarnya

Persiapan

pertama-tama kita edit dulu master.blade.php untuk membuat link halaman yang menuju ke halaman tampil berita dan tambah berita

resources/views/layouts/**master.blade.php**

```
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
    <title>Portal News</title>
    <link href="{{asset('template/css/media_query.css')}}" rel="stylesheet" type="text/css"/>
    <link href="{{asset('template/css/bootstrap.css')}}" rel="stylesheet" type="text/css"/>
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet"
integrity="sha384-wvfXpqpZZVQGK6TAh5PVLGOfQNHSoD2xbE+QkPxCAF1NEevOEH3Sl0sibVcOQVnN" crossorigin="anonymous">
    <link href="{{asset('template/css/animate.css')}}" rel="stylesheet" type="text/css"/>
    <link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet">
    <link href="{{asset('template/css/owl.carousel.css')}}" rel="stylesheet">
```

```

rel="stylesheet" type="text/css"/>
<link href="{{asset('template/css/owl.theme.default.css')}}" rel="stylesheet" type="text/css"/>

<link href="{{asset('template/css/style_1.css')}}" rel="stylesheet" type="text/css"/>

<script src="{{asset('template/js/modernizr-3.5.0.min.js')}}"></script>
</head>
<body>

{{-- Header --}}
<div class="container-fluid bg-faded fh5co_padd_mediya padding_786">
    <div class="container padding_786">
        <nav class="navbar navbar-toggleable-md navbar-light ">
            <button class="navbar-toggler navbar-toggler-right mt-3" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"><span class="fa fa-bars"></span></button>
            <a class="navbar-brand" href="#"></a>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item ">
                        <a class="nav-link" href="/">Home <span class="sr-only">(current)</span></a>
                    </li>
                    <li class="nav-item ">
                        <a class="nav-link" href="/about">About <span class="sr-only">(current)</span></a>
                    </li>
                
            
```

```

        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle"
href="#" id="dropdownMenuButton2" data-toggle="dropdown"
                aria-haspopup="true"
            aria-expanded="false">Kategori <span
            class="sr-only">(current)</span></a>
                <div class="dropdown-menu"
aria-labelledby="dropdownMenuLink_1">
                    <a class="dropdown-item"
href="/categories">Tampil Kategori</a>
                    <a class="dropdown-item"
href="/categories/categories">Tambah Kategori</a>
                </div>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle"
href="#" id="dropdownMenuButton2" data-toggle="dropdown"
                    aria-haspopup="true"
            aria-expanded="false">Berita <span
            class="sr-only">(current)</span></a>
                <div class="dropdown-menu"
aria-labelledby="dropdownMenuLink_1">
                    <a class="dropdown-item"
href="/news">Tampil Berita</a>
                    <a class="dropdown-item"
href="/news/create">Tambah Berita</a>
                </div>
            </li>
        </ul>
    </div>
</nav>
</div>
{{-- End Header --}}
{{-- Content --}}

```

```

<div class="container-fluid pb-4 pt-4 paddding">
    @yield('content')
</div>
{{-- End Content --} }

{{-- Footer --}}
<div class="container-fluid fh5co_footer_right_reserved" >
    <div class="container">
        <div class="row ">
            <div class="col-12 col-md-6 py-4 Reserved" > ©
                Copyright 2018, All rights reserved. Design by <a href="https://freehtml5.co" title="Free HTML5 Bootstrap templates">FreeHTML5.co</a>.
            </div>
        </div>
    </div>
{{-- End Footer --} }

<div class="gototop js-top">
    <a href="#" class="js-gotop"><i class="fa fa-arrow-up"></i></a>
</div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script
src="{{asset('template/js/owl.carousel.min.js')}}"></script>
<!--<script
src="https://code.jquery.com/jquery-3.1.1.slim.min.js"
integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k90mg/a/EheAdgtzNs3hpaf6Ed950n" crossorigin="anonymous"></script>-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js"
integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8ffjk5JAIXUYHKk

```

```

Dx6Qin1DkWx51bBrb" crossorigin="anonymous">></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/b
ootstrap.min.js"
integrity="sha384-vBWWz1ZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyEvk
E+jo0ieGizqPLForn" crossorigin="anonymous">></script>
<!-- Waypoints -->
<script
src="{{asset('template/js/jquery.waypoints.min.js')}}"></script>
<!-- Main -->
<script src="{{asset('template/js/main.js')}}"></script>

</body>
</html>

```

The screenshot shows a web application interface. At the top, there is a navigation bar with links for Home, About, Kategori (which has a dropdown arrow), and Berita (which also has a dropdown arrow). Under the Berita link, there are two buttons: 'Tampil Berita' and 'Tambah Berita'. Below the navigation bar is a section titled 'Categories' containing a table. The table has three columns: '#', 'Name', and 'Action'. There is one row in the table with the value '1' in the '#' column, 'Sport' in the 'Name' column, and two buttons in the 'Action' column: 'Edit' (blue) and 'Delete' (red). At the very bottom of the page, there is a dark footer bar with the text '© Copyright 2018, All rights reserved. Design by FreeHTML5.co.'

#	Name	Action
1	Sport	Edit Delete

kemudian kita atur controller, model dan routenya

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;
use App\Http\Controllers\NewsController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');

//Create Data Category
Route::get('/categories/categories',
[CategoriesController::class,
'create'])->name('category.create');

Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');

//Read Data Category
Route::get('/categories', [CategoriesController::class,
'index'])->name('category.index');

//Update Data Category
Route::get('/categories/{id}/edit', [CategoriesController::class,
'edit'])->name('category.edit');

Route::put('/categories/{id}', [CategoriesController::class,
'update'])->name('category.update');

//Delete Data Category
Route::delete('/categories/{id}', [CategoriesController::class,
```

```
'destroy'])->name('category.destroy');

//CRUD News
Route::resource('news', NewsController::class);
```

pada code diatas kita menggunakan route resource yaitu route khusus untuk membuat CRUD nya untuk melihat rute nya bisa gunakan

`php artisan route:list`

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan route:list
GET|HEAD / ..... home > DashboardController@home
POST _ignition/execute-solution ..... ignition.executeSolution > Spatie\LaravelIgnition\ExecuteSolutionController
GET|HEAD _ignition/health-check ..... ignition.healthCheck > Spatie\LaravelIgnition\HealthCheckController
POST _ignition/update-config ..... ignition.updateConfig > Spatie\LaravelIgnition\UpdateConfigController
GET|HEAD about ..... about > DashboardController@about
GET|HEAD api/user ..... 
POST categories ..... category.store > CategoriesController@store
GET|HEAD categories ..... category.index > CategoriesController@index
GET|HEAD categories/categories ..... category.create > CategoriesController@create
PUT categories/{id} ..... category.update > CategoriesController@update
DELETE categories/{id} ..... category.destroy > CategoriesController@destroy
GET|HEAD categories/{id}/edit ..... category.edit > CategoriesController@edit
GET|HEAD news ..... news.index > NewsController@index
POST news ..... news.store > NewsController@store
GET|HEAD news/create ..... news.create > NewsController@create
PUT|PATCH news/{news} ..... news.show > NewsController@show
DELETE news/{news} ..... news.update > NewsController@update
GET|HEAD news/{news}/edit ..... news.destroy > NewsController@destroy
GET|HEAD sanctum/csrf-cookie ..... news.edit > NewsController@edit
..... Laravel\Sanctum > CsrfCookieController@show
```

pada saat kita menjalankan `route:list` maka akan tampil route CRUD dari resource route yang sudah kita buat

Kemudian kita akan membuat modelnya News dan Categories

`php artisan make:model News`

`php artisan make:model Categories`

`App\Models\Categories.php`

```
<?php

namespace App\Models;

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

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

class Categories extends Model
{
    use HasFactory;
    protected $table = 'categories';
    protected $fillable = ['name'];
}
```

app\Models\News.php

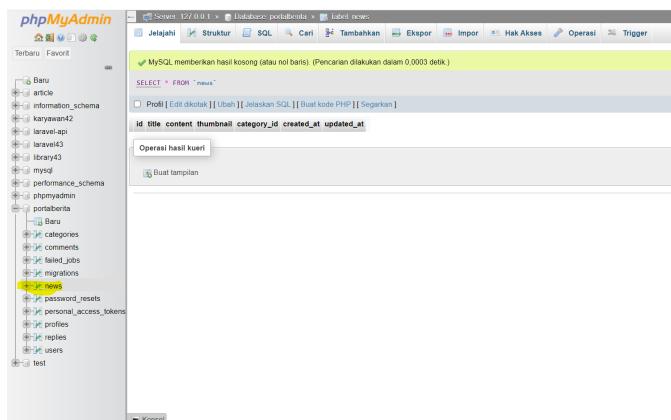
```
<?php

namespace App\Models;

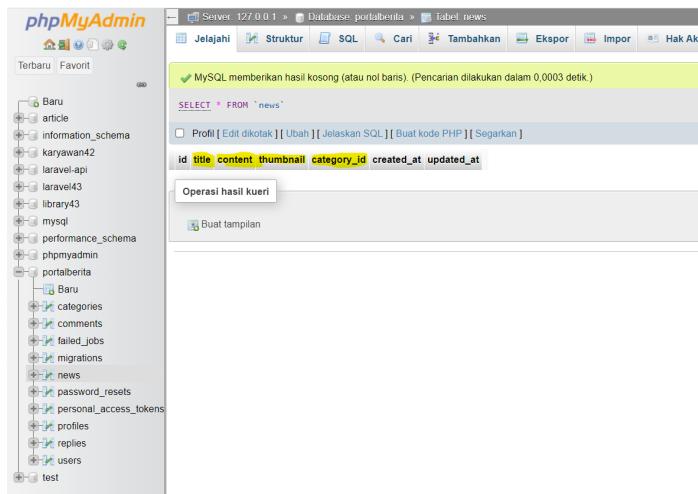
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class News extends Model
{
    use HasFactory;
    protected $table = 'news';
    protected $fillable = ['title', 'content', 'thumbnail',
'category_id'];
}
```

pada code diatas kita membuat protected \$table = 'news' sesuai dengan nama table yang ada di database



code diatas juga kita membuat protected \$fillable untuk mendaftarkan kolom-kolom apa saja yang akan kita manipulasi sesuai dengan yang ada di table news



Terakhir kita akan membuat controller resource untuk NewsController

```
php artisan make:controller NewsController --resource
```

maka akan terbuat fungsi CRUD untuk news controller dari index sampai destroy
app\Http\Controllers\NewsController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
```

```
use App\Models\News;
use App\Models\Categories;

class NewsController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        //
    }
}
```

```

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    //
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    //
}

/**
 * Remove the specified resource from storage.

```

```

*
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    //
}

```

pada code diatas kita memanggil kedua model yang sudah kita buat yaitu news `use App\Models\News;` dan categories `use App\Models\Categories;`

Create Data

pertama-tama kita akan membuat Create datanya dulu pada Tabel News menggunakan ORM dan upload file image

pertama kita masukan code pada function create dan function store pada NewsController

`app\Http\Controllers\NewsController.php`

```

public function create()
{
    $categories = Categories::all();

    return view('news.tambah', ['categories' =>
$categories]);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request

```

```

    * @return \Illuminate\Http\Response
    */
public function store(Request $request)
{
    $request->validate([
        'title' => 'required|max:255',
        'content' => 'required',
        'category_id' => 'required|integer',
        'thumbnail' =>
        'required|mimes:jpg,jpeg,png|max:2048',
    ]);

    $ImageName = time() . '.' .
$request->thumbnail->extension();

    $request->thumbnail->move(public_path('uploads'),
$ImageName);

    $news = new News;

    $news->title = $request->input('title');
    $news->content = $request->input('content');
    $news->category_id = $request->input('category_id');
    $news->thumbnail = $ImageName;

    $news->save();

    return redirect('/news')->with('success', 'Data Berita
berhasil Ditambahkan');
}

```

pada code diatas kita menambahkan code pada function create dan function store

1. fungsi pertama create untuk mengarahkan kita ke file view sub folder news file tambah.blade.php dengan melakukan passing data dari mengambil semua data dari table categories menggunakan model categories

2. fungsi kedua untuk melakukan validasi dari inputan title, content, image,dan category_id dengan rules required max:255 yaitu maksimal karakter yang dimasukan tidak boleh lebih dari 255 karakter, mimes:jpg, jpeg, png yaitu format file yang bisa di masukan hanya berypa formar jpg,jpeg,png dan integer dimana inputan hanya berupa nomor.

kemudian kita melakukan konversi nama dari file gambarnya supaya unik

```
$ImageName = time() . '.' . $request->thumbnail->extension();  
yang berupa waktu pembuatan file ditambah ekstensi format file (7980813132.png).  
lalu menyimpan file gambarnya pada subfolder upload di folder public berdasarkan  
request thumbnail $request->thumbnail->move(public_path('uploads'),  
$ImageName);
```

kemudian melakukan insert data menggunakan orm pada model news berdasarkan request input title, content, category_id pada tampilan form viewnya dan untuk thumbnail kita ambil dari \$imageName yang sudah di konversi setelah berhasil melakukan insert data ke database maka akan di arahkan ke halaman /news dengan pesan singkat menggunakan flash message

resources/views/news/tambah.blade.php

```
@extends('layouts.master')  
  
@section('content')  
  
<div class="container padding">  
    <h1 class="text-primary my-2">Add News</h1>  
    <form action="/news" method="post"  
        enctype="multipart/form-data">  
        @csrf  
        <div class="form-group">  
            <label for="name">Title</label>  
            <input type="text" class="form-control" id="title"  
                name="title">  
        </div>  
        @error('title')  
            <div class="alert alert-danger">{{ $message }}</div>  
        @enderror
```

```

<div class="form-group">
    <label for="content">Content</label>
    <textarea name="content" id="content"
class="form-control" cols="30" rows="10"></textarea>
</div>
@error('content')
    <div class="alert alert-danger">{{ $message }}</div>
@enderror
<div class="form-group">
    <label for="category_id">Category</label>
    <select name="category_id" id="category_id"
class="form-control">
        <option value="">Select a Category</option>
        @forelse ($categories as $item)
            <option
value="{{ $item->id }}">{{ $item->name }}</option>
        @empty
            <option value="">No Categories</option>
        @endforelse
        </select>
</div>
@error('category_id')
    <div class="alert alert-danger">{{ $message }}</div>
@enderror
<div class="form-group">
    <label for="thumbnail">Thumbnail</label>
    <input type="file" class="form-control"
id="thumbnail" name="thumbnail">
</div>
@error('thumbnail')
    <div class="alert alert-danger">{{ $message }}</div>
@enderror

<button type="submit" class="btn
btn-primary">Tambah</button>
</form>

```

```
</div>

@endsection
```

pada code diatas didalam form kita menggunakan attribute action /news dan method post sesuai dengan route resource news yang sudah di buat dan menggunakan attribute enctype="multipart/form-data" karena kita ingin menginput sebuah file gambar

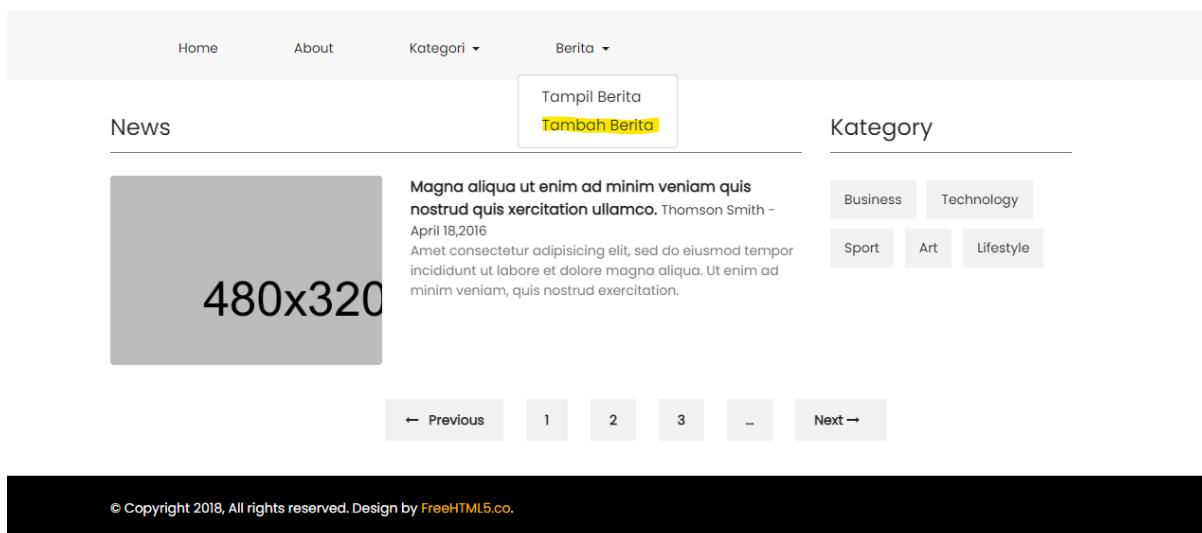
di setiap inputan, textarea dan select kita menggunakan attribute name sebagai request di controller function store nanti

di tag select kita melakukan looping untuk menampilkan semua categories di option select nya sesuai dengan yang kita passing di function create di NewsController

terdapat @error untuk manampilkan error validasi di setiap inputanya

dan terakhir button type submit yang akan menjalankan action form saat tombolnya di klik

kita akan coba test codenya pertama tama



tampilan validasi

Home About Kategorij • Berilo •

Add News

Title

The title field is required.

Content

The content field is required.

Category
 Select a Category

The category id field is required.

Thumbnail
 Choose File No file chosen

The thumbnail field is required.

© Copyright 2018, All rights reserved. Design by [ReapTime5.co](#).

input data

Title
sport News 1

Content

Justo, vel finibus arcu. Ut interdum aliquet massa, et sollicitudin nisi fermentum et. Morbi maximus ligula ac justo faucibus viverra. Duis sed rutrum sapien, id finibus velit. Sed tristique dolor sed cursus molestie. Praesent non mauris est. In posuere velit vitae ipsum tempus, eget aliquam lorem pulvinar. Cras nisi nibh, convallis imperdiet rutrum faucibus, fermentum eget metus. Mauris sit amet mauris condimentum. Rhoncus neque at, auctor lacus Vivamus erat dolor, porttitor vel cursus nec, placerat scelerisque metus.

Donec feugiat suscipit nulla non daicibus purus. Pellentesque cursus metus sed egestas tristique. Nunc nisi varius ante, a pellentesque urna odio id lacus. Nam tristique eros nec enim tincidunt, eget viverra lacus sodales. Sed magna quam, auctor et finibus et, pellentesque non elit. Etiam urna arcu, viverra eget erat non, fermentum vestibulum nunc. Morbi suscipit massa eget odio pellentesque, at sagittis felis luctus. Vivamus augue odio, eleifend vel erat eget, laoreet pharetra tellus. Curabitur nibh neque, consectetur ut placatat blandit, dictum et dolor. Fusce a sapien vel elit lacinia fermentum. Quisque venenatis in nibh sed vehicula. Duis id dolor diam. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

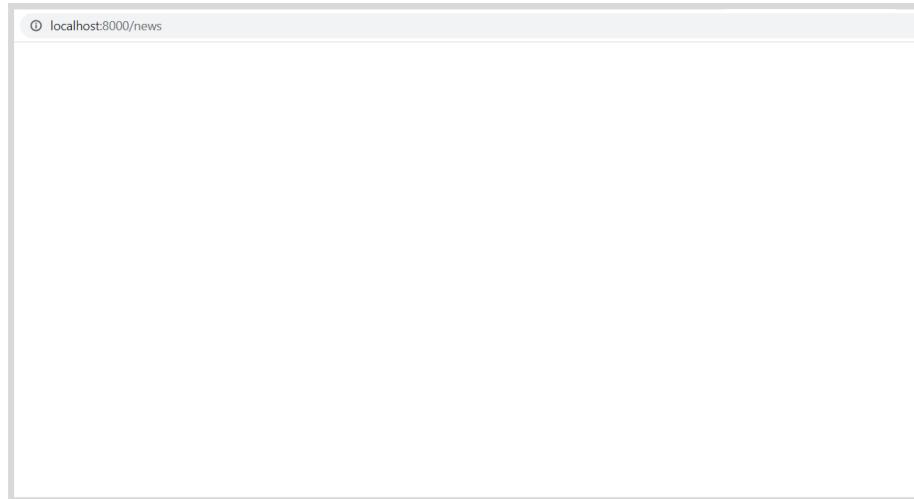
Category
Sport

Thumbnail
Choose File beragam-manfaat-olahraga.jpg

Tambah

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

tampilan pada saat semua inputan di sudah diisi dan tombol Tambah di klik



tampilan nya akan kosong seperti diatas karena kita belum membuat tampilan untuk Read Data

Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0.0004 detik.)

SELECT * FROM `news`

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

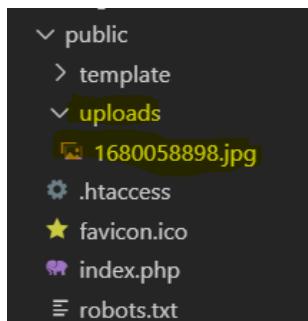
Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini

Extra options

	T	id	title	content	thumbnail	category_id	created_at	updated_at
<input type="checkbox"/>	Ubah Salin Hapus	1	sport News 1	Lorem ipsum dolor sit amet, consectetur adipiscing...	1680058898.jpg	3	2023-03-29 03:01:38	2023-03-29 03:01:38

Pilih Semua Dengan pilihan: [Ubah](#) [Salin](#) [Hapus](#) [Eksport](#)

tampilan data berhasil tersimpan di database dan juga file gambarnya masuk ke folder public subfolder uploads



Read Data

setelah berhasil melakukan Create data sekarang kita akan menampilkan datanya menggunakan card dan detail dari berita

pertama-tama kita atur dulu controller dan views disini kita akan menambahkan code pada function index dan function show

app\Http\Controllers\NewsController.php

```
public function index()
{
    $news = News::all();
    return view('news.tampil', ['news' => $news]);
}

public function show($id)
{
    $news = News::find($id);
    return view('news.detail', ['news' => $news]);
}
```

pada code diatas kita menambahkan code pada function index dan function show

1. fungsi pertama index untuk mengarahkan kita ke file view sub folder news file `tampil.blade.php` dengan melakukan passing data dari mengambil semua data dari table news menggunakan model News
2. fungsi kedua show untuk melakukan passing data dimana kita mengambil data berdasarkan parameter id pada model News yang di passing ke sub folder news file `detail.blade.php`

resources/views/news/tampil.blade.php

```
@extends('layouts.master')

@section('content')
<div class="container padding">
@if(session('success'))
    <div class="alert alert-success">
        {{ session('success') }}
    </div>
@endif
<h1 class="text-primary my-2">Berita</h1>
<div class="row">
@forelse ($news as $item)
    <div class="col-4">
        <div class="card px-2 py-2">
            
            <div class="card-body">
                <h1 >{{$item->title}}</h1>
                <p class="card-text">{{
Str::limit($item->content, 100) }}</p>

<small>{{$item->created_at->diffForHumans() }}</small>
                <a href="/news/{{ $item->id }}" class="btn
btn-primary btn-block">Read More</a>
            </div>
        </div>
    </div>
@empty
    <h3>Tidak Ada Berita</h3>
@endforelse
</div>
</div>
```

```
@endsection
```

di awal code kita menggunakan @session untuk menampilkan tampilan flash message
disini kita bagi kolom nya menjadi 3 menggunakan grid dari bootstrap dimana kita
melakukan looping dari variabel yang sudah di passing dari newscontroller function index

```
@forelse ($news as $item)
```

menampilkan thumbnail yang di ambil dari folder public

```
{asset('/uploads/'.$item->thumbnail)}}
```

disini juga kita menampilkan titlenya {{\$item->title}}

lalu untuk content kita limit data yang tampil hanya 100 karakter saja

```
Str::limit($item->content, 100) }}
```

dan menampilkan waktu pembuatannya

```
<small>{{$item->created_at->diffForHumans()}}</small>
```

terakhir kita membuat tombol untuk mengarah ke halaman detail id}}" class="btn btn-primary btn-block">Read More

resources/views/news/detail.blade.php

```
@extends('layouts.master')

@section('content')



 }})



## {{ $news->title }}



{{ Str::limit($news->content, 100) }}



<small>{{ $news->created_at->diffForHumans() }}</small>



Read More


```

```

<h3 class="text-primary">{$news->title}</h3>
<p>{$news->content}</p>
</div>
</div>
<a href="/news" class="btn btn-secondary
btn-block">Kembali</a>
</div>

@endsection

```

pada code di atas kita membagi 2 kolom-kolom 1 bernilai 4 dan kolom 2 bernilai 8 dimana di kolom pertama kita mempulkan gambar berdasarkan variabel yang kita passing di NewsController function show {{asset('/uploads/'.\$news->thumbnail)}} dan di kolom 2 kita manampilkan title dan content dari berita {{\$news->title}} {{\$news->content}} dan terakhir kita membuat tombol untuk kembali ke halaman tampil berita Kembali

tampilan tampil berita <http://localhost:8000/news>

The screenshot shows a news website layout. At the top, there is a navigation bar with links for Home, About, Kategori, and Berita. Below the navigation, there is a section titled "Berita" with a thumbnail image of a person running. The article title is "sport News 1". The content of the article is a short snippet of lorem ipsum text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer venenatis purus tellus, id imperdiet...". Below the text, it says "1 hour ago" and has a blue "Read More" button. At the bottom of the page, there is a footer bar with the text "© Copyright 2018, All rights reserved. Design by FreeHTML5.co."

tampilan saat tombol read more di klik (detail berita)

Home About Kategori Berita

sport News 1

... (long block of placeholder text)

Kembali

Update Data

setelah berhasil melakukan Create data dan Read Data sekarang kita akan membuat halaman dari edit data dan funcgsi update datanya

pertama-tama kita atur dulu controller dan views disini kita akan menambahkan code pada function edit dan function update dan menambahkan use file di controllernya yang bertujuan untuk menhapus file yang ada di folder public

app\Http\Controllers\NewsController.php

```
<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Models\News;  
use App\Models\Categories;  
use Illuminate\Support\Facades\File;
```

```

class NewsController extends Controller
{
    .....
    public function edit($id)
    {
        $categories = Categories::all();
        $news = News::find($id);
        return view('news.edit', ['news' => $news, 'categories'
=> $categories]);
    }

    public function update(Request $request, $id)
    {
        $request->validate([
            'title' => 'required|max:255',
            'content' => 'required',
            'category_id' => 'required|integer',
            'thumbnail' => 'mimes:jpg,jpeg,png|max:2048',
        ]);

        $news = News::find($id);

        $news->title = $request->input('title');
        $news->content = $request->input('content');
        $news->category_id = $request->input('category_id');
        if ($request->has('thumbnail')) {
            if ($news->thumbnail != null) {
                $path = "uploads/";
                File::delete($path . $news->thumbnail);
            }
            $ImageName = time() . '.' .
        $request->thumbnail->extension();

        $request->thumbnail->move(public_path('uploads'),

```

```

$ImageName) ;

        $news->thumbnail = $ImageName;
    }

    $news->save();

    return redirect('/news')->with('success', 'Data Berita
berhasil Diupdate');
}

.....
}

```

code diatas kita melakukan use Illuminate\Support\Facades\File yang berfungsi agar kita bisa menghapus file pada folder public. disini kita menambahkan 2 function yaitu function edit dan function update

- fungsi pertama edit untuk mengarahkan kita ke file view sub folder news file edit.blade.php dengan melakukan passing 2 data dari pertama mengambil semua data dari table categories menggunakan model categories dan passing kedua mengambil data berdasarkan id dari model news
 - fungsi kedua untuk melakukan validasi dari inputan title, content, image,dan category_id dengan rules required max:255 yaitu maksimal karakter yang dimasukan tidak boleh lebih dari 255 karakter, mimes:jpg, jpeg, png yaitu format file yang bisa di masukan hanya berypa formar jpg,jpeg,png dan integer dimana inputan hanya berupa nomor bedanya dengan validasi function store di sini thumbnail tidak kita berikan rules required
- kemudian kita melakukan kondisi jika thumbnail di masukan file if
`($request->has('thumbnail'))` dan didalamnya terdapat kondisi lagi jika
`$news->thumbnail != null` kolom thumbnail di table news tidak kosong maka
kita akan menghapus file yang lama menggunakan fungsi file berdasarkan path
nya
`$path = "uploads/"; File::delete($path .`
`$news->thumbnail);`
- kemudian kita melakukan konversi nama dari file gambarnya supaya unik
`$ImageName = time() . '.' . $request->thumbnail->extension();`

yang berupa waktu pembuatan file ditambah ekstensi format file (7980813132.png). lalu menyimpan file gambar yang baru pada subfolder upload di folder public berdasarkan request thumbnail

```
$request->thumbnail->move (public_path('uploads') ,  
$ImageName) ;  
  
kemudian melakukan update data berdasarkan id pada parameter menggunakan  
find($id) orm pada model news berdasarkan request input title, content,  
category_id pada tampilan form viewnya dan untuk thumbnail kita ambil dari  
$imageName yang sudah di konversi setelah berhasil melakukan update data ke  
database maka akan di arahkan ke halaman /news dengan pesan singkat  
menggunakan flash message
```

resources/views/news/**tampil.blade.php**

```
@extends('layouts.master')  
  
@section('content')  
  
<div class="container paddding">  
    @if(session('success'))  
        <div class="alert alert-success">  
            {{ session('success') }}  
        </div>  
    @endif  
    <h1 class="text-primary my-2">Berita</h1>  
    <div class="row">  
        @forelse ($news as $item)  
            <div class="col-4">  
                <div class="card px-2 py-2">  
                      
                    <div class="card-body">  
                        <h1>{{$item->title}}</h1>  
                        <p class="card-text">{{  
Str::limit($item->content, 100) }}</p>
```

```

<small>{{ $item->created_at->diffForHumans() }}</small>
        <a href="/news/{{ $item->id }}" class="btn
btn-primary btn-block">Read More</a>
        <div class="row my-2">
            <div class="col">
                <a href="/news/{{ $item->id }}/edit"
class="btn btn-warning btn-block btn-sm">Edit</a>
            </div>
            <div class="col">
                <form action="/news/{{ $item->id }}"
method="post">
                    @csrf
                    @method('delete')
                    <input type="submit" value="Delete"
class="btn btn-danger btn-block btn-sm">
                </form>
            </div>
        </div>
    </div>
    @empty
        <h3>Tidak Ada Berita</h3>
    @endforelse
</div>
</div>

@endsection

```

pada kode diatas kita membuat 2 kolom dibawah link ke halaman detail data kolom pertama bertungsi mangarahkan kita ke halaman edit form `/news/{{ $item->id }}/edit` dengan membawa parameter id, kolom ke 2 kita membuat form yang berfungsi menjalankan hapus data berdasarkan parameter id dengan attribute action `action="/news/{{ $item->id }}" method="post"`. disini kita juga mengubah

methodnya dari post menjadi delete @method('delete') dan menggunakan proteksi
@csrf dan input type submit

resources/views/news/**edit.blade.php**

```
@extends('layouts.master')

@section('content')



<h1 class="text-primary my-2">Update News</h1>
    <form action="/news/{{ $news->id }}" method="post"
        enctype="multipart/form-data">
        @csrf
        @method('put')
        <div class="form-group">
            <label for="name">Title</label>
            <input type="text" class="form-control"
                value="{{ $news->title }}" id="title" name="title">
        </div>
        @error('title')
            <div class="alert alert-danger">{{ $message }}</div>
        @enderror
        <div class="form-group">
            <label for="content">Content</label>
            <textarea name="content" id="content"
                class="form-control" cols="30"
                rows="10">{{ $news->content }}</textarea>
        </div>
        @error('content')
            <div class="alert alert-danger">{{ $message }}</div>
        @enderror
        <div class="form-group">
            <label for="category_id">Category</label>
            <select name="category_id" id="category_id"
                class="form-control">
                <option value="">Select a Category</option>
                @forelse ($categories as $item)
                    @if ($item->id === $news->category_id)


```

```

                <option value="{{$item->id}}" selected>{{$item->name}}</option>
            @else
                <option
value="{{$item->id}}">{{$item->name}}</option>
            @endif
            @empty
                <option value="">No Categories</option>
            @endforelse
        </select>
    </div>
    @error('category_id')
        <div class="alert alert-danger">{{ $message }}</div>
    @enderror
    <div class="form-group">
        <label for="thumbnail">Thumbnail</label>
        <input type="file" class="form-control"
id="thumbnail" name="thumbnail">
    </div>
    @error('thumbnail')
        <div class="alert alert-danger">{{ $message }}</div>
    @enderror

        <button type="submit" class="btn
btn-primary">Update</button>
    </form>
</div>

@endsection

```

pada code diatas didalam form kita menggunakan attribute action `/news/{{ $news->id }}` yang bertujuan untuk menangkap id dan method post yang kita ubah menjadi put @method="put" sesuai dengan route resource news yang sudah di buat dan menggunakan attribute enctype="multipart/form-data" karena kita ingin menginput sebuah file gambar

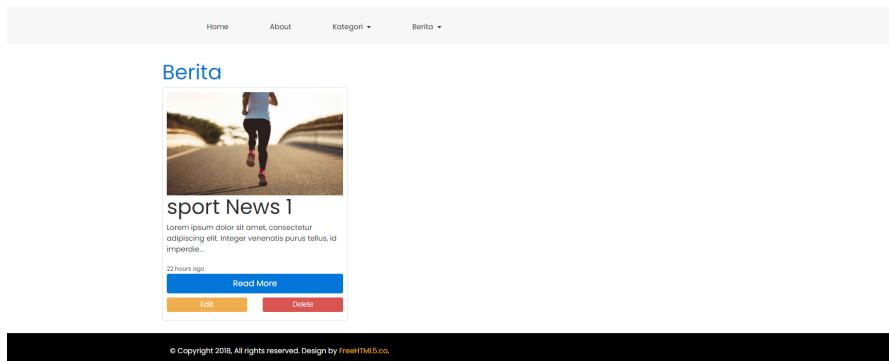
di setiap inputan, textarea dan select kita menggunakan attribute name sebagai request di controller function store nanti dan menampilkan data berdasarkan `$news` yang sudah di passing di function edit

di tag select kita melakukan looping untuk menampilkan semua categories dan menggunakan kondisi `@if ($item->id === $news->category_id)` yang berfungsi mengecek nilai apakah kolom `category_id` pada table news sama dengan id category yang telah di looping jika nilai sama maka di option kita tambahkan attribute `selected` tapi jika tidak maka option tidak di beri `selected` select nya sesuai dengan yang kita passing di function edit di NewsController

terdapat `@error` untuk menampilkan error validasi di setiap inputnya

dan terakhir button type submit yang akan menjalankan action form saat tombolnya di klik

tampilan halaman berita <http://localhost:8000/news>



pada saat tombol edit diklik

Home About Kategori ▾ Berita ▾

Update News

Title

Content

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer venenatis purus tellus, id imperdiet ante pulvinar vitae. Phasellus varius, lectus ultricies pharetra, est nulla iaculis tortor, id sodales est nunc vitae dolor. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec ullamcorper tempor arcu, in facilisis lorem mattis eget. Proin mattis lacina scelerisque. Maecenas faucibus a nibh nec condimentum. Nam vel massa, fermentum ut elementum sed, ornare eu dolor. Morbi nisi orci, fringilla quis ligula in, scelerisque laoreet odio.

 Aenean eu lacinia lorem, sit amet maximus ex. Nam sed lectus et mauris suscipit tincidunt sit amet in erat. Pellentesque dictum magna felis, porttitor congue velit mattis ac. Nam massa ex, dictum eu ex vitae, ullamcorper pellentesque arcu. Sed nisi mi, fringilla eget iaculis eget, gravida in lorem. Nam faucibus urna nec ipsum pulvinar consectetur. Ut nec arcu eros. Nunc vestibulum lobortis accumsan.

 Nulla facilisi. Nulla cursus mauris quis massa maximus cursus. Vestibulum dui augue, egestas in velit eu, tincidunt iaculis elit. Vestibulum tristique erat vel

Category

Thumbnail
 No file chosen

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

disini kita akan mencoba mengedit data title content dan menganti file thumbnail dari setiap inputan

Home About Kategori ▾ Berita ▾

Update News

Title

Content

 edit 123 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer venenatis purus tellus, id imperdiet ante pulvinar vitae. Phasellus varius, lectus ultricies pharetra, est nulla iaculis tortor, id sodales est nunc vitae dolor. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec ullamcorper tempor arcu, in facilisis lorem mattis eget. Proin mattis lacina scelerisque. Maecenas faucibus a nibh nec condimentum. Nam vel massa, fermentum ut elementum sed, ornare eu dolor. Morbi nisi orci, fringilla quis ligula in, scelerisque laoreet odio.

 Aenean eu lacinia lorem, sit amet maximus ex. Nam sed lectus et mauris suscipit tincidunt sit amet in erat. Pellentesque dictum magna felis, porttitor congue velit mattis ac. Nam massa ex, dictum eu ex vitae, ullamcorper pellentesque arcu. Sed nisi mi, fringilla eget iaculis eget, gravida in lorem. Nam faucibus urna nec ipsum pulvinar consectetur. Ut nec arcu eros. Nunc vestibulum lobortis accumsan.

 Nulla facilisi. Nulla cursus mauris quis massa maximus cursus. Vestibulum dui augue, egestas in velit eu, tincidunt iaculis elit. Vestibulum tristique erat vel

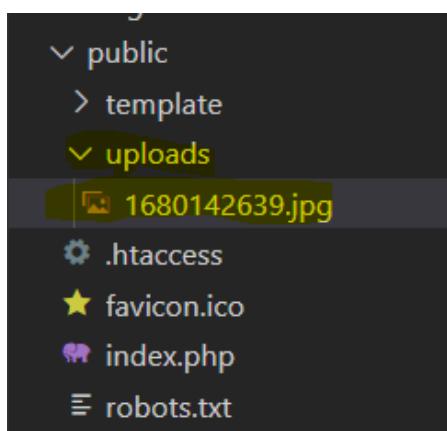
Category

Thumbnail
 fotojet-3-minjpg-20210819104440.jpg

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

setelah tombol update di klik

The screenshot shows a web interface with a navigation bar at the top containing 'Home', 'About', 'Kategori', and 'Berita'. A green banner below the navigation bar displays the message 'Data Berita berhasil Diupdate'. The main content area is titled 'Berita' and features a grid of small sport-related icons. Below the grid, a news item is displayed with the title 'edit 123 sport News 1'. The item includes a preview image showing various sports like running, tennis, and badminton, a short description 'edit 123 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer venenatis purus tellus, id...', a timestamp '23 hours ago', and three buttons: 'Read More' (blue), 'Edit' (orange), and 'Delete' (red). At the bottom of the page, a black footer bar contains the copyright notice '© Copyright 2018, All rights reserved. Design by FreeHTML5.co.'



pada gambar diatas kita berhasil melakukan update data dan menghapus gambar yang lama di folder public menjadi gambar yang baru

Delete Data

sekarang kita masuk ke fungsi terakhir pada CRUD nya yaitu delete data disini kita hanya perlu menambahkan code pada function destroy newscontroller

app\Http\Controllers\NewsController.php

```
public function destroy($id)
{
    $news = News::find($id);
    $path = "uploads/";
    File::delete($path . $news->thumbnail);

    $news->delete();

    return redirect('/news')->with('success', 'Data Berita berhasil Didelete');
}
```

pada code diatas kita akan mengambil data pada model News berdasarkan parameter id \$news = News::find(\$id); dan menghapus file yang berada di subfolder upload folder public berdasarkan model News thumbnail File::delete(\$path . \$news->thumbnail); kemudian menghapus data nya yang ada di database \$news->delete(); terakhir kita akan di arahkan ke halaman tampil news /news dengan membawa pesan flash message

tampilan halaman berita <http://localhost:8000/news>



Berita

edit 123 sport
News 1

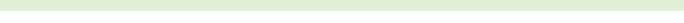
23 hours ago

Read More

Edit Delete

© Copyright 2018. All rights reserved. Design by FreeHTML5.co.

tampilan pada saat kita mengklik tombol Delete

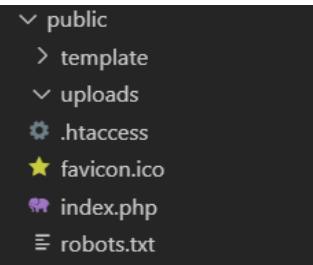


Data Berita berhasil Didelete

Berita

Tidak Ada Berita

© Copyright 2018. All rights reserved. Design by FreeHTML5.co.



pada gambar diatas kita berhasil melakukan delete data dan menghapus gambar pada subfolder uploads pada folder public dan menjalankan @empty pada forelse tampil.blade.php karena data news di database sudah kosong

Bab 14 Laravel Authentication

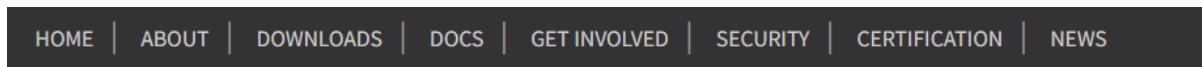
Sekarang kita akan belajar tentang Authentication di Laravel yaitu sistem pengamanan di Laravel lewat autentikasi seperti register dan login. selain auth kita juga akan belajar tentang middleware yaitu cara kita untuk mengamankan halamannya

Persiapan

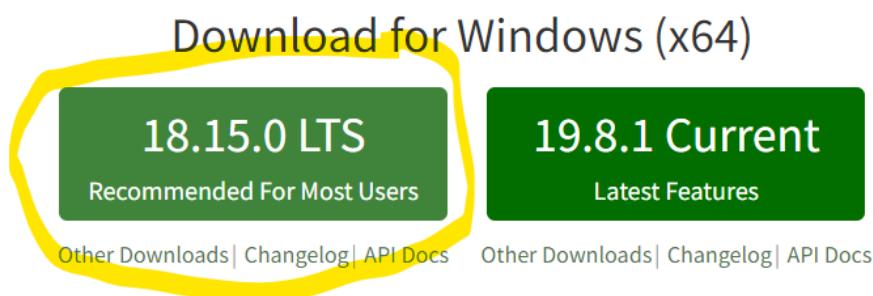
untuk fitur auth nya pertama-tama kita install dulu nodeJS <https://nodejs.org/en>

Node.js adalah sebuah platform runtime JavaScript open-source yang dibangun di atas engine JavaScript V8 milik Google. Node.js memungkinkan pengembang untuk menulis kode JavaScript di sisi server (server-side), menggantikan penggunaan bahasa pemrograman server-side tradisional seperti PHP, Python, atau Ruby.

silahkan download dulu nodeJS download versi LTS



Node.js® is an open-source, cross-platform JavaScript runtime environment.



For information about supported releases, see the [release schedule](#).

setelah download silahkan install secara default next next finish, selesai instalasi silahkan restart pcnya supaya nodeJS berjalan maksimal

check nodeJS di terminal/cmd

```
PS C:\Users\rezky> node --version  
v18.14.2  
PS C:\Users\rezky> npm --version  
8.12.2
```

disini kita membutuhkan nodejs untuk scaffolding laravel mix nanti yang dibutuhkan untuk laravel auth nanti

selanjutnya kita akan memasang auth di project Portal Berita yang sudah kita buat menggunakan laravel ui disini kita menggunakan laravel ui versi 4

Supported Versions	
Version	Laravel Version
1.x	5.8, 6.x
2.x	7.x
3.x	8.x
4.x	9.x, 10.x

```
composer require laravel/ui
```

tampilan setelah laravel/ui berhasil di install

```
PS D:\Rebook\UI\Rebook-Portal-Berita>  
* History restored  
> @php artisan package:discover --ansi  
[INFO] Discovering packages.  
  
fruitcake/laravel-cors ..... DONE  
laravel/sail ..... DONE  
laravel/sanctum ..... DONE  
laravel/tinker ..... DONE  
laravel/ui ..... DONE  
nesbot/carbon ..... DONE  
nunomaduro/collision ..... DONE  
nunomaduro/termwind ..... DONE  
spatie/laravel-ignition ..... DONE  
  
84 packages you are using are looking for funding.  
Use the "composer fund" command to find out more!  
> @php artisan vendor:publish --tag=laravel-assets --ansi --force  
[INFO] No publishable resources for tag [laravel-assets].
```

setelah itu jalankan

```
php artisan ui bootstrap --auth
```

untuk generate scaffolding

tampilan saat berhasil generate scaffolding

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> php artisan ui bootstrap --auth
INFO  Authentication scaffolding generated successfully.
INFO  Bootstrap scaffolding installed successfully.
WARN  Please run [npm install && npm run dev] to compile your fresh scaffolding.
```

disini kita ubah dulu package.json sebelum menjalankan npm install dan npm run dev
package.json disini kita mengatur manganti bagian package.json agar menjalankan vite

```
{
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build"
  },
  "devDependencies": {
    "@popperjs/core": "^2.11.6",
    "axios": "^0.25",
    "bootstrap": "^5.2.3",
    "laravel-mix": "^6.0.6",
    "lodash": "^4.17.19",
    "postcss": "^8.1.14",
    "sass": "^1.56.1",
    "vite": "^4.0.0",
    "laravel-vite-plugin": "^0.7.2"
  }
}
```

disini kita melakukan install pada scaffolding

```
npm install
```

tampilan jika npm install selesai (jika ada npm WARN abaikan saja)

```
● PS D:\ebook\im-ebook-portal-berita\portal-berita> npm install
  up to date, audited 749 packages in 8s
  78 packages are looking for funding
    run `npm fund` for details
  found 0 vulnerabilities
```

kemudian jalankan laravel mix

```
npm run dev
```

tampilan jika npm run dev berhasil

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> npm run dev
> dev
> vite

VITE v4.2.1 ready in 471 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h to show help

LARAVEL v9.52.4 plugin v0.7.4
→ APP_URL: http://localhost
```

disini kita menjalankan 2 terminal satu untuk menjalankan npm run dev dan satu lagi untuk menjalankan php artisan serve

```

PS D:\ebook\im-ebook-portal-berita\portal-berita> npm run dev
> dev
> vite

VITE v4.2.1 ready in 471 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h to show help

LARAVEL v9.52.4 plugin v0.7.4
→ APP_URL: http://localhost

```

tampilan login <http://localhost:8000/login>

tampilan register <http://localhost:8000/register>

kita akan coba untuk mengubah beberapa codenya pertama kita ubah dulu routes nya

routes/web.php

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;
use App\Http\Controllers\NewsController;

```

```

Route::get('/', [DashboardController::class, 'home'])->name('home');

Route::get('/about', [DashboardController::class, 'about'])->name('about');

//Create Data Category
Route::get('/categories/categories', [CategoriesController::class,
'create'])->name('category.create');
Route::post('/categories', [CategoriesController::class, 'store'])->name('category.store');

//Read Data Category
Route::get('/categories', [CategoriesController::class, 'index'])->name('category.index');

//Update Data Category
Route::get('/categories/{id}/edit', [CategoriesController::class,
'edit'])->name('category.edit');
Route::put('/categories/{id}', [CategoriesController::class,
'update'])->name('category.update');

//Delete Data Category
Route::delete('/categories/{id}', [CategoriesController::class,
'destroy'])->name('category.destroy');

//CRUD News
Route::resource('news', NewsController::class);

Auth::routes();

```

terdapat route Auth::routes() yaitu rute auth nya yang didapatkan pada saat kita mengenerate scaffolding dan satu route lagi yang di hapus yaitu Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');

setelah mengatur routes nya kita akan mengubah route service provider

app\Providers\RouteServiceProvider.php

```
<?php

namespace App\Providers;

use Illuminate\Cache\RateLimiting\Limit;
use Illuminate\Foundation\Support\Providers\RouteServiceProvider
as ServiceProvider;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\RateLimiter;
use Illuminate\Support\Facades\Route;

class RouteServiceProvider extends ServiceProvider
{
    /**
     * The path to the "home" route for your application.
     *
     * This is used by Laravel authentication to redirect users
     * after login.
     *
     * @var string
     */
    public const HOME = '/';

    /**
     * The controller namespace for the application.
     *
     * When present, controller route declarations will
     * automatically be prefixed with this namespace.
     *
     * @var string|null
     */
    // protected $namespace = 'App\\Http\\Controllers';

    /**

```

```

 * Define your route model bindings, pattern filters, etc.
 *
 * @return void
 */
public function boot()
{
    $this->configureRateLimiting();

    $this->routes(function () {
        Route::prefix('api')
            ->middleware('api')
            ->namespace($this->namespace)
            ->group(base_path('routes/api.php'));

        Route::middleware('web')
            ->namespace($this->namespace)
            ->group(base_path('routes/web.php'));
    });
}

/**
 * Configure the rate limiters for the application.
 *
 * @return void
 */
protected function configureRateLimiting()
{
    RateLimiter::for('api', function (Request $request) {
        return Limit::perMinute(60)->by($request->user() ?->id
?: $request->ip());
    });
}

```

pada code diatas kita mengubah code `public const HOME` ke halaman '/', berfungsi pada saat kita berhasil login atau register akan di arahkan ke halaman '/'

kemudian kita tambahkan beberapa code pada template utama `master.blade.php`

`resources/views/layouts/master.blade.php`

```
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no">
    <title>Portal News</title>
    <link href="{{asset('template/css/media_query.css')}}" rel="stylesheet" type="text/css"/>
    <link href="{{asset('template/css/bootstrap.css')}}" rel="stylesheet" type="text/css"/>
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet"
integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGofQNHSoD2xbE+QkPxCAF1NEevOEH3Sl0sibVcOQVnN" crossorigin="anonymous">
    <link href="{{asset('template/css/animate.css')}}" rel="stylesheet" type="text/css"/>
    <link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet">
    <link href="{{asset('template/css/owl.carousel.css')}}" rel="stylesheet" type="text/css"/>
    <link href="{{asset('template/css/owl.theme.default.css')}}" rel="stylesheet" type="text/css"/>
    <!-- Bootstrap CSS -->
    <link href="{{asset('template/css/style_1.css')}}" rel="stylesheet" type="text/css"/>
```

```

<!-- Modernizr JS -->
<script
src="{{asset('template/js/modernizr-3.5.0.min.js')}}"></script>

</head>
<body>
{{-- Header --}}
<div class="container-fluid bg-faded fh5co_padd_mediya
padding_786">
    <div class="container padding_786">
        <nav class="navbar navbar-toggleable-md navbar-light ">
            <button class="navbar-toggler navbar-toggler-right
mt-3" type="button" data-toggle="collapse"
                    data-target="#navbarSupportedContent"
                    aria-controls="navbarSupportedContent" aria-expanded="false"
                    aria-label="Toggle navigation"><span
class="fa fa-bars"></span></button>
            <a class="navbar-brand" href="#"></a>
            <div class="collapse navbar-collapse"
id="navbarSupportedContent">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item ">
                        <a class="nav-link" href="/">Home <span
class="sr-only">(current)</span></a>
                    </li>
                    <li class="nav-item ">
                        <a class="nav-link" href="/about">About
<span class="sr-only">(current)</span></a>
                    </li>
                    @auth
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle"
href="#" id="dropdownMenuButton2" data-toggle="dropdown"
                            aria-haspopup="true"
                            aria-expanded="false">Kategori <span

```

```

class="sr-only">(current)</span></a>
            <div class="dropdown-menu"
aria-labelledby="dropdownMenuLink_1">
                <a class="dropdown-item"
href="/categories">Tampil Kategori</a>
                <a class="dropdown-item"
href="/categories/categories">Tambah Kategori</a>
            </div>
        </li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle"
href="#" id="dropdownMenuButton2" data-toggle="dropdown"
aria-haspopup="true"
aria-expanded="false">Berita <span
class="sr-only">(current)</span></a>
            <div class="dropdown-menu"
aria-labelledby="dropdownMenuLink_1">
                <a class="dropdown-item"
href="/news">Tampil Berita</a>
                <a class="dropdown-item"
href="/news/create">Tambah Berita</a>
            </div>
        </li>
        @endauth
        @guest
        <li class="nav-item">
            <a class="nav-link bg-primary"
href="/login"> <span class="text-white">Login</span> <span
class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
            <a class="nav-link bg-primary"
href="/register"> <span class="text-white">Register</span> <span
class="sr-only">(current)</span></a>
        </li>
        @endguest

```

```

        @auth
            <a class="nav-link bg-danger" href="{{ route('logout') }} "
                onclick="event.preventDefault();
                document.getElementById('logout-form').submit();">
                Logout
            </a>

            <form id="logout-form" action="{{ route('logout') }}"
                method="POST" class="d-none">
                @csrf
            </form>
        @endauth
    </ul>
</div>
</nav>
</div>
</div>
{{-- End Header --}}


{{-- Content --}}
<div class="container-fluid pb-4 pt-4 padding">
    @yield('content')
</div>
{{-- End Content --}}


{{-- Footer --}}
<div class="container-fluid fh5co_footer_right_reserved" >
    <div class="container">
        <div class="row ">
            <div class="col-12 col-md-6 py-4 Reserved"> ©
            Copyright 2018, All rights reserved. Design by <a
            href="https://freehtml5.co" title="Free HTML5 Bootstrap
            templates">FreeHTML5.co</a>. </div>
        </div>
    </div>

```

```

        </div>
</div>
{{-- End Footer --}}


<div class="gototop js-top">
    <a href="#" class="js-gotop"><i class="fa fa-arrow-up"></i></a>
</div>




<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script
src="{{asset('template/js/owl.carousel.min.js')}}"></script>
<!--<script
src="https://code.jquery.com/jquery-3.1.1.slim.min.js"
integrity="sha384-A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k9Omga/EheAdgttzNs3hpfrag6Ed950n" crossorigin="anonymous"></script>-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js"
integrity="sha384-DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIXUYHKkDx6Qin1DkWx51bBrb"
crossorigin="anonymous"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/js/bootstrap.min.js"
integrity="sha384-vBWWzlZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyevkE+jo0ieGizqPLForn"
crossorigin="anonymous"></script>
<!-- Waypoints -->
<script
src="{{asset('template/js/jquery.waypoints.min.js')}}"></script>

```

```

<!-- Main -->
<script src="{{asset('template/js/main.js')}}"></script>

</body>
</html>

```

pada code diatas kita menambahkan data code @auth berfungsi untuk menampilkan code pada saat kita berhasil auth (login/register) @guest berfungsi untuk menampilkan code yang belum auth atau sebagai tamu di link berita, kategori dan juga logout kita memasang @auth jadi link halaman akan tampil pada saat kita sudah auth dan untuk link login dan register kita memasang @guest dimana tampilan link login dan register akan tampil pada saat kita belum auth atau sebagai tamu

halaman <http://localhost:8000/> sebagai tamu atau guest

The screenshot displays a responsive news website layout. At the top is a navigation bar with 'Home', 'About', 'Login' (highlighted in blue), and 'Register' buttons. Below the navigation are two main sections. The left section, titled 'News', contains a large gray placeholder image with the dimensions '480x320' overlaid. The right section, titled 'Kategory', lists several categories: Business, Technology, Sport, Art, and Lifestyle. At the bottom of the page is a footer bar with the copyright notice: '© Copyright 2018, All rights reserved. Design by FreeHTML5.co.'

terakhir kita akan membuat model untuk profile dan controller profile

```
php artisan make:model Profile
```

app\Models\Profile.php

```
<?php

namespace App\Models;

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

class Profile extends Model
{
    use HasFactory;
    protected $table = 'profiles';
    protected $fillable = ['age', 'bio', 'address',
    'photo_profile', 'user_id'];
}
```

pada code diatas kita mengatur protected table dan fillable sesudai dengan nama table dan kolom di table tersebut yang akan kita manipulasi datanya

selanjutkan kita buat controller untuk profile

```
php artisan make:controller ProfileController
```

Register

sekarang kita akan mengatur bagian register pertama tama kita atur dulu tampilannya

resources/views/auth/**register.blade.php**

```
@extends('layouts.master')

@section('content')
<div class="container padding my-5">
    <h1 class="text-primary my-2">Register</h1>
    <form method="POST" action="{{ route('register') }}>
        @csrf

        <div class="form-group">
            <label for="name">Name</label>
            <input id="name" type="text" class="form-control"
name="name" value="{{ old('name') }}" autocomplete="name"
autofocus>

            @error('name')
                <span class="text-danger" role="alert">
                    <strong>{{ $message }}</strong>
                </span>
            @enderror
        </div>

        <div class="form-group">
            <label for="email" >Email</label>

            <input id="email" type="email" class="form-control"
name="email" value="{{ old('email') }}" autocomplete="email">

            @error('email')
                <span class="text-danger" role="alert">
                    <strong>{{ $message }}</strong>
                </span>
            @enderror
        </div>
    </form>
</div>
```

```

        </span>
    @enderror
</div>

<div class="form-group">
    <label for="password">Password</label>

    <input id="password" type="password"
class="form-control" name="password" autocomplete="new-password">

    @error('password')
        <span class="text-danger" role="alert">
            <strong>{{ $message }}</strong>
        </span>
    @enderror
</div>

<div class="form-group">
    <label for="password-confirm">Confirm
Password</label>

    <input id="password-confirm" type="password"
class="form-control" name="password_confirmation"
autocomplete="new-password">

</div>

<div class="form-group">
    <label for="age">Age </label>

    <input id="age" type="text" class="form-control"
@error('age') is-invalid @enderror" name="age" value="{{
old('age') }}">

    @error('age')
        <span class="text-danger" role="alert">

```

```

        <strong>{ { $message } }</strong>
    </span>
    @enderror
</div>

<div class="form-group">
    <label for="address" >Address </label>
    <textarea name="address" class="form-control" id="address" cols="10" rows="5"></textarea>

    @error('address')
        <span class="text-danger" role="alert">
            <strong>{ { $message } }</strong>
        </span>
    @enderror
</div>
<div class="form-group">
    <label for="bio" >Bio </label>
    <textarea name="bio" class="form-control" id="bio" cols="10" rows="5"></textarea>

    @error('bio')
        <span class="text-danger" role="alert">
            <strong>{ { $message } }</strong>
        </span>
    @enderror
</div>
<button type="submit" class="btn btn-primary btn-lg btn-block">
    {{ __('Register') }}
</button>
</form>
</div>
@endsection

```

pada code di atas kita menambahkan beberapa inputan seperti age, address, dan bio dan juga disini kita mengubah tampilannya dan menganti @extends ke master bukan ke app

selanjutnya kita akan mengatur Register Controller

app\Http\Controllers\Auth\RegisterController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use App\Models\User;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use App\Models\Profile;

class RegisterController extends Controller
{
    /*
    |--------------------------------------------------------------------------
    | Register Controller
    |--------------------------------------------------------------------------
    |
    | This controller handles the registration of new users as
    | well as their
    | validation and creation. By default this controller uses a
    | trait to
    | provide this functionality without requiring any additional
    | code.
    */

    /**
     * @var \Illuminate\Foundation\Auth\RegistersUsers
     */
    protected $registerTrait;

    /**
     * Create a new controller instance.
     *
     * @param \Illuminate\Foundation\Auth\RegistersUsers $registerTrait
     */
    public function __construct(RegistersUsers $registerTrait)
    {
        $this->registerTrait = $registerTrait;
    }
}
```

```

|
 */

use RegistersUsers;

/**
 * Where to redirect users after registration.
 *
 * @var string
 */
protected $redirectTo = RouteServiceProvider::HOME;

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guest');
}

/**
 * Get a validator for an incoming registration request.
 *
 * @param array $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255',
            'unique:users'],
        'password' => ['required', 'string', 'min:8'],
        'confirmed',
    ]);
}

```

```

        'bio' => ['required'],
        'age' => ['required'],
        'address' => ['required']
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\Models\User
 */
protected function create(array $data)
{
    $user = User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);

    Profile::create([
        'bio' => $data['bio'],
        'address' => $data['address'],
        'age' => $data['age'],
        'user_id' => $user->id
    ]);

    return $user;
}
}

```

pada code diatas kita memanggil model profile use App\Models\Profile; kemudian kita validasi untuk inputan age, bio dan address dan kita membuat variabel yang menampung create user berfungsi untuk mengambil id dari user untuk di isi di kolom user_id

model profile dan dicode diatas kita juga melakukan create untuk model Profile dimana user_id di ambil dari \$user->id terakhir tinggal melakukan return pada \$user;

tampilan validasi pada halaman register

The screenshot shows a registration form titled "Register". At the top, there is a navigation bar with links for "Home", "About", "Login", and "Register". The "Register" link is highlighted with a blue background and white text. Below the navigation bar, the form fields are displayed:

- Name**: An input field with the placeholder "Name". A red error message "The name field is required." is displayed below it.
- Email**: An input field with the placeholder "Email". A red error message "The email field is required." is displayed below it.
- Password**: An input field with the placeholder "Password". A red error message "The password field is required." is displayed below it.
- Confirm Password**: An input field with the placeholder "Confirm Password".
- Age**: An input field with the placeholder "Age". A red error message "The age field is required." is displayed below it.
- Address**: A text area with the placeholder "Address". A red error message "The address field is required." is displayed below it.
- Bio**: A text area with the placeholder "Bio". A red error message "The bio field is required." is displayed below it.

At the bottom of the form is a blue "Register" button.

disini kita akan mengisi setiap inputan dengan benar

The screenshot shows a registration form with the following fields filled:

- Name: rezy
- Email: rezy@mail.com
- Password: (hidden)
- Confirm Password: (hidden)
- Age: 27
- Address: Makassar
- Bio: Hello World

At the bottom is a large blue "Register" button.

dan data yang kita input akan berhasil masuk ke table user dan table profile

MySQL Workbench screenshot showing the 'users' table with one row:

	id	name	email	email_verified_at	password	remember_token	created_at	updated_at
	1	rezy	rezy@mail.com	NULL	S2y\$10\$NrKsrBl31bu8427BM3fMTelBvcmd1tpmXibyJN97G1k...	NULL	2023-03-31 07:12:14	2023-03-31 07:12:14

MySQL Workbench screenshot showing the 'profiles' table with one row:

	id	age	bio	address	photo_profile	user_id	created_at	updated_at
	1	27	Hello World	Makassar	NULL	1	2023-03-31 07:12:14	2023-03-31 07:12:14

setelah tombol register di klik maka kita akan di arahkan ke halaman <http://localhost:8000/> sesuai dengan yang sudah kita atur di route service provider

The screenshot shows a news website layout. At the top, there is a navigation bar with links for Home, About, Kategori, Berita, and Logout. Below the navigation bar, there are two main sections: 'News' on the left and 'Kategori' on the right. The 'News' section features a large gray placeholder image labeled '480x320'. The 'Kategori' section includes a sidebar with categories: Business, Technology, Sport, Art, and Lifestyle. At the bottom, there is a footer bar with copyright information and a navigation menu.

dan tampilan pada saat kita logout

The screenshot shows the same news website layout, but the navigation bar has changed. It now includes a 'Login' button in a blue box and a 'Register' button in a blue box, while the other links (Home, About, Kategori, Berita) are in a standard gray box. The rest of the page, including the news article and category sidebar, remains the same as in the previous screenshot.

Login

untuk bagian login kita cukup ubah saja tampilan halaman nya di view

resources/views/auth/login.blade.php

```
@extends('layouts.master')

@section('content')
<div class="container padding my-5">
    <h1 class="text-primary my-2">Login</h1>
    <form method="POST" action="{{ route('login') }}>
        @csrf

        <div class="form-group">
            <label for="email">Email Address</label>

            <input id="email" type="email"
class="form-control" name="email" value="{{ old('email') }}"
autocomplete="email" autofocus>

            @error('email')
                <span class="text-danger" role="alert">
                    <strong>{{ $message }}</strong>
                </span>
            @enderror
        </div>

        <div class="form-group">
            <label for="password">Password</label>

            <input id="password" type="password"
class="form-control" name="password"
autocomplete="current-password">

            @error('password')
```

```


    <strong>{{ $message }}</strong>

@enderror
</div>

<button type="submit" class="btn btn-primary
btn-block">
    {{ __('Login') }}
</button>

</form>
</div>
@endsection

```

tampilan validasi login

The screenshot shows a login interface with the following elements:

- Header:** A navigation bar with links for "Home", "About", "Login" (which is highlighted in blue), and "Register".
- Title:** The word "Login" in a large blue font.
- Email Address:** A text input field with the placeholder "Email Address". Below it, a red error message says "The email field is required.".
- Password:** A text input field with the placeholder "Password". Below it, a red error message says "The password field is required.".
- Login Button:** A large blue button labeled "Login".
- Footer:** A black footer bar containing the text "© Copyright 2018, All rights reserved. Design by FreeHTML5.co.".

tampilan pada saat berhasil login

The screenshot shows a news website layout. At the top is a navigation bar with links for Home, About, Kategori (with a dropdown), Berita (with a dropdown), and Logout. Below the navigation is a section titled "News" featuring a large placeholder image labeled "480x320". To the right is a "Kategori" sidebar with categories: Business, Technology, Sport, Art, and Lifestyle. A news item is displayed with the title "Magna aliqua ut enim ad minim veniam quis nostrud quis exercitation ullamco.", author "Thomson Smith", date "April 18,2016", and a short description. Below the news item is a pagination bar with links for Previous, page 1, page 2, page 3, ..., and Next →. The footer contains a copyright notice: "© Copyright 2018, All rights reserved. Design by FreeHTML5.co."

Dashboard

disini kita akan coba mengatur tampilan dari halaman Home

pertama-tama kita masukan dulu 3 data Kategori dan 10 Data Berita di database

The screenshot shows the MySQL Workbench interface with the database "portabenta" selected. In the "Tabel: categories" tab, a query is running: "SELECT * FROM `categories`". The results show three rows of data:

	id	name	created_at	updated_at
<input type="checkbox"/>	3	Sport	NULL	NULL
<input type="checkbox"/>	4	IT	NULL	NULL
<input type="checkbox"/>	5	Travel	NULL	NULL

Menampilkan baris 0 - 9 (total 10. Pencarian dilakukan dalam 0.0003 detik.)

```
SELECT * FROM `news`
```

Tampilan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key: Tidak ada | Extra options:

	#	T	ID	Judul	Konten	Thumbnail	Kategori ID	Dibuat pada	Diperbarui pada	
<input type="checkbox"/>	1		Ubah		Hapus	2 Berita Sport 1	Vestibulum ante ipsum primis in faucibus orci luctus et...	1680490781.jpg	3 2023-04-03 02:59:41	2023-04-03 02:59:41
<input type="checkbox"/>	2		Ubah		Hapus	3 Sport News 2	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680490976.jpg	3 2023-04-03 03:01:16	2023-04-03 03:02:00
<input type="checkbox"/>	3		Ubah		Hapus	4 Berita Travel 1	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680490993.jpg	5 2023-04-03 03:01:43	2023-04-03 03:01:43
<input type="checkbox"/>	4		Ubah		Hapus	5 Travel News 2	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680490954.jpg	5 2023-04-03 03:02:34	2023-04-03 03:02:34
<input type="checkbox"/>	5		Ubah		Hapus	6 Berita Wisata 3	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680490991.jpg	5 2023-04-03 03:03:00	2023-04-03 03:03:11
<input type="checkbox"/>	6		Ubah		Hapus	7 Apa itu PHP?	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680491073.png	4 2023-04-03 03:04:33	2023-04-03 03:04:33
<input type="checkbox"/>	7		Ubah		Hapus	8 Apa itu HTML?	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680491123.png	4 2023-04-03 03:05:23	2023-04-03 03:05:23
<input type="checkbox"/>	8		Ubah		Hapus	9 Apa itu MS Edge?	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680491159.png	4 2023-04-03 03:05:59	2023-04-03 03:05:59
<input type="checkbox"/>	9		Ubah		Hapus	10 Mengapa Chrome?	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680491198.png	4 2023-04-03 03:06:38	2023-04-03 03:06:38
<input type="checkbox"/>	10		Ubah		Hapus	11 Pembuatan HTML	...Lorem ipsum dolor sit amet, consectetur adipiscing...	1680491251.jpg	4 2023-04-03 03:07:31	2023-04-03 03:07:31

Pilih Semua Dengan pilihan Ubah Salin Hapus Eksport

pertama-tama kita edit dulu controllernya dan mengatur tampilan views

Halaman Dashboard

app\Http\Controllers\DashboardController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Categories;
use App\Models\News;
use App\Models\Profile;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class DashboardController extends Controller
{
    public function home()
    {
        $news = News::simplePaginate(5);
        $categories = Categories::get();
        return view('welcome', ['news' => $news, 'categories' => $categories]);
    }

    public function about()
}
```

```

    {
        $id_user = Auth::id();
        $profile = Profile::where('user_id', $id_user)->first();

        return view('about', ['profile' => $profile]);
    }
}

```

pada code di atas kita melakukan use pada model categories news, dan profile kemudian kita juga memanggil fungsi auth dari laravel

pada function home kita melakukan passing variabel \$news dan \$categories ke halaman view welcome.blade.php dimana variabel \$news kita melakukan pagination pada data dan di \$categories kita mengambil semua data categories

pada function about kita membuat variabel \$id_user yang menampung id dari user yang auth (Login/Register) dan membuat variabel \$profile yang menampung data berdasarkan kolom user_id dari id user yang login kemudian passing variabel \$profile ke halaman view about.blade.php

resources/views/welcome.blade.php

```

@extends('layouts.master')

@section('content')
<div class="container padding">
    <div class="row mx-0">
        <div class="col-md-8 animate-box"
            data-animate-effect="fadeInLeft">
            <div>
                <div class="fh5co_heading
                    fh5co_heading_border_bottom py-2 mb-4">News</div>
            </div>
            @forelse ($news as $item)

```

```

<div class="row pb-4">
    <div class="col-md-5">
        <div class="fh5co_hover_news_img">
            <div class="fh5co_news_img"></div>
            <div></div>
        </div>
    </div>
    <div class="col-md-7 animate-box">
        <a href="single.html" class="fh5co_magna
py-2"> Magna aliqua ut enim ad minim veniam quis
nostrud quis xercitation ullamco. </a>
Thomson Smith - April 18, 2016
        <div class="fh5co_consectetur"> Amet
consectetur adipisicing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim
ad minim veniam, quis nostrud exercitation.
        </div>
    </div>
</div>
@empty
<h3>Tidak Ada Berita</h3>
@endforelse

</div>
<div class="col-md-3 animate-box"
data-animate-effect="fadeInRight">
    <div>
        <div class="fh5co_heading
fh5co_heading_border_bottom py-2 mb-4">Kategory</div>
    </div>
    <div class="clearfix"></div>
    <div class="fh5co_tags_all">
        @forelse ($categories as $item)
        <a href="#">
            class="fh5co_tagg">{{ $item->name }}</a>

```

```

@empty
    <a href="">Tidak Ada Kategori</a>
@endforelse
</dunuiw>
</div>
</div>
<div class="row mx-0 animate-box"
data-animate-effect="fadeInUp">
    <div class="col-12 text-center pb-4 pt-4">
        {{ $news->links() }}
    </div>
</div>
</div>
@endsection

```

pada code diatas kita melakukan looping pada \$news dan \$categories dari yang dipassing di function home di DashboardController nya kemudian kita juga menambahkan {{ \$news->links() }} untuk menjalankan fungsi paginationnya dan pada looping \$news kita melakukan limit data ke 200 karakter

tampilan page pertama

The screenshot shows a news application interface. At the top, there's a navigation bar with links for Home, About, Kategori (Category), Berita (News), and Logout. Below the navigation, there's a section titled 'Kategori' with buttons for Sport, IT, and Travel. The main content area displays a grid of news articles. Each article card includes a thumbnail image, the title, a timestamp, and a brief description. The first article is titled 'Berita Sport 1' and describes a runner. The second is 'Sport News 2' with a group of people playing. The third is 'Berita Travel 1' with an image of a travel scene. The fourth is 'Travel News 2' with a landscape image. At the bottom, there are navigation buttons for 'Previous' and 'Next', with 'Next' being highlighted with a red circle.

tampilan page 2 (setelah next diklik)

The screenshot shows a web application interface. At the top, there is a navigation bar with links for 'Home', 'About', 'Kategori', 'Berita', and a red 'Logout' button. Below the navigation bar, the word 'News' is displayed. To the right, there is a 'Category' section with three buttons: 'Sport', 'IT', and 'Travel'. The main content area displays five news items, each with a thumbnail image, the title, and a brief description. The titles are: 'Apa itu PHP?', 'Apa itu HTML?', 'Apa itu MS Edge?', 'Mengapa Chrome?', and 'Permbuat HTML'. Each news item includes placeholder text for the full article content.

News

Kategori

Sport IT Travel

Apa itu PHP? 34 minutes ago
Cras pharetra rutrum nulla sed tristique. Donec ante morbi, ultrices ut blandit et, semper sit amet elit. Du...

Apa itu HTML? 34 minutes ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce efficitur mollis efficitur. Cras pharetra rutrum nulla sed tristique. Donec ante morbi, ultrices ut blandit et, semper sit amet elit. Du...

Apa itu MS Edge? 34 minutes ago
Cras pharetra rutrum nulla sed tristique. Donec ante morbi, ultrices ut blandit et, semper sit amet elit. Du...

Mengapa Chrome? 34 minutes ago
Cras pharetra rutrum nulla sed tristique. Donec ante morbi, ultrices ut blandit et, semper sit amet elit. Du...

Permbuat HTML 34 minutes ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce efficitur mollis efficitur. Cras pharetra rutrum nulla sed tristique. Donec ante morbi, ultrices ut blandit et, semper sit amet elit. Du...

« Previous Next »

© Copyright 2018. All rights reserved. Design by freeHTML5.co.

Halaman About

resources/views/about.blade.php

```
@extends('layouts.master')

@section('content')
<div class="container padding" style="height: 80vh">
    @auth
        <a href="/profile" class="btn btn-primary btn-lg my-5">Edit
    Profile</a>

    <div class="row mx-0">
        <div class="col-4">
            
</div>
<div class="col-8">
    <h1>{{{$profile->age}}}</h1>
    <h3>Biodata Saya adalah</h3>
    <p>
        {{$profile->bio}}
    </p>
</div>
</div>

@endauth

@guest
    Belum Login
@endguest
</div>
@endsection

```

pada code diatas kita menampilkan data Profile dari \$profile yang sudah di passing code diatas kita juga menggunakan @guest dan @auth untuk membedakan tampilan yang sudah login dan yang belum dan juga kita membuat tombol yang mengarah ke form edit profile Edit Profile

tampilan pada saat auth

The screenshot shows a user profile page. At the top, there is a navigation bar with links: Home, About, Kategori ▾, Berita ▾, and Logout. Below the navigation bar is a blue button labeled "Edit Profile". To the left is a large black circular placeholder for a profile picture, with the text "480 x 320" inside it. To the right of the placeholder is a user statistics block showing "(27)" and the text "Biodata Saya adalah". Underneath that, it says "Hello World". At the bottom of the page is a dark footer bar containing the text "© Copyright 2018, All rights reserved. Design by FreeHTML5.co."

tampilan jika belum auth atau Guest

The screenshot shows a login page. At the top, there is a navigation bar with links: Home, About, Login, and Register. The "Login" link is highlighted with a blue background. Below the navigation bar, the text "Belum Login" is displayed. At the bottom of the page is a dark footer bar containing the text "© Copyright 2018, All rights reserved. Design by FreeHTML5.co."

Edit Profile

disini kita akan membuat form dan fungsi untuk melakukan edit profile
pertama-tama ubah dulu code pada ProfileController dan mendaftarkan rute web.php

routes\web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;
use App\Http\Controllers\NewsController;
use App\Http\Controllers\ProfileController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');

//Create Data Category
Route::get('/categories/categories',
[CategoriesController::class,
'create'])->name('category.create');

Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');

//Read Data Category
Route::get('/categories', [CategoriesController::class,
'index'])->name('category.index');

//Update Data Category
Route::get('/categories/{id}/edit', [CategoriesController::class,
'edit'])->name('category.edit');

Route::put('/categories/{id}', [CategoriesController::class,
'update'])->name('category.update');
```

```
//Delete Data Category
Route::delete('/categories/{id}', [CategoriesController::class,
'destroy'])->name('category.destroy');

//Profile
Route::get('/profile', [ProfileController::class,
'edit'])->name('profile.edit');
Route::put('/profile/{id}', [ProfileController::class,
'update'])->name('profile.edit');

//CRUD News
Route::resource('news', NewsController::class);

Auth::routes();

Route::get('/home', [App\Http\Controllers\HomeController::class,
'index'])->name('home');
```

pada code diatas kita memanggil profileControllernya use

App\Http\Controllers\ProfileController; kemudian kita membuat 2 route yang pertama /profile yang mengarah ke ProfileController function edit dan /profile/{id} yang mengarah ke ProfileController function update dengan membawa parameter id

app\Http\Controllers\ProfileController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use App\Models\Profile;
use Illuminate\Support\Facades\File;

class ProfileController extends Controller
{
    public function edit()
    {
        $id_user = Auth::id();
        $profile = Profile::where('user_id', $id_user)->first();

        return view('profile.edit', ['profile' => $profile]);
    }

    public function update(Request $request, $id)
    {
        $request->validate([
            'age' => 'required',
            'bio' => 'required',
            'address' => 'required',
            'photo_profile' => 'mimes:jpg,jpeg,png|max:2048',
        ]);

        $profile = Profile::find($id);

        $profile->age = $request->input('age');
        $profile->bio = $request->input('bio');
        $profile->address = $request->input('address');
        if ($request->has('photo_profile')) {
```

```

        if ($profile->photo_profile != null) {
            $path = "uploads/profile/";
            File::delete($path . $profile->photo_profile);
        }

        $ImageName = time() . '.' .

$request->photo_profile->extension();

$request->photo_profile->move(public_path('uploads/profile/'),
$ImageName);

$profile->photo_profile = $ImageName;

}

$profile->save();

return redirect('/about')->with('success', 'Data Profile
berhasil Diupdate');
}
}

```

pada code diatas kita membuat 2 fungsi

function edit berfungsi untuk mengambil data profile berdasarkan kondisi user_id sama dengan id user yang login yang di tampung di variabel \$profile kemudian melakukan passing data dari \$profile ke view subfolder profile file **edit.blade.php**

function update berfungsi melakukan update data dari Profile dan melakukan validasi dari inputan form yang ada di **edit.blade.php** dan melakukan update data dan file image berdasarkan \$id dan setelah selesai akan di arahkan ke halaman /about dengan membawa flash message

resources/views/profile/edit.blade.php

```
@extends('layouts.master')

@section('content')



<h1 class="text-primary my-2">Update Profile</h1>
    <form action="/profile/{{ $profile->id }}" method="post"
        enctype="multipart/form-data">
        @csrf
        @method('put')
        <div class="form-group">
            <label for="age">Age</label>
            <input type="number" class="form-control"
                value="{{ $profile->age }}" id="age" name="age">
        </div>
        @error('age')
            <div class="alert alert-danger">{{ $message }}</div>
        @enderror
        <div class="form-group">
            <label for="bio">Biodata</label>
            <textarea name="bio" id="bio" class="form-control"
                cols="30" rows="10">{{ $profile->bio }}</textarea>
        </div>
        @error('bio')
            <div class="alert alert-danger">{{ $message }}</div>
        @enderror
        <div class="form-group">
            <label for="address">Address</label>
            <textarea name="address" id="address"
                class="form-control" cols="30"
                rows="10">{{ $profile->address }}</textarea>
        </div>
        @error('address')


```

```

        <div class="alert alert-danger">{{ $message }}</div>
    @enderror

    <div class="form-group">
        <label for="photo_profile">Photo Profile</label>
        <input type="file" name="photo_profile"
id="photo_profile" class="form-control">
    </div>
    @error('photo_profile')
        <div class="alert alert-danger">{{ $message }}</div>
    @enderror

        <button type="submit" class="btn
btn-primary">Update</button>
    </form>
</div>

@endsection

```

resources/views/about.blade.php

```

@extends('layouts.master')

@section('content')
<div class="container padding" style="height: 80vh">
    @auth
        @if(session('success'))
            <div class="alert alert-success">
                {{ session('success') }}
            </div>
        @endif
        <a href="/profile" class="btn btn-primary btn-lg my-5">Edit
Profile</a>

        <div class="row mx-0">
            <div class="col-4">

```

```

@if ($profile->photo_profile)
    
@else
    
@endif
</div>
<div class="col-8">
    <h1>{{ $profile->age }}</h1>
    <h3>Biodata Saya adalah</h3>
    <p>
        {{ $profile->bio }}
    </p>
</div>
</div>

@endauth

@guest
    <h1>Belum Login</h1>
@endguest
</div>
@endsection

```

pada file about.blade.php kita manambahkan code kondisi if else yaitu jika kolom photo_profile tidak kosong maka dia akan menampilkan tersimpan di folder public/uploads/profile berdasarkan nama file data pada kolom photo_profile di table profile tapi selain itu jika kolom photo_profile di table profile kosong maka akan menampilkan dummy image

dan juga kita memanipulkan session flash messagenya @if(session('success'))

disini kita akan mencoba melakukan edit data pada profilenya <http://localhost:8000/profile>

Home About Kategori ▾ Berita ▾ Logout

Update Profile

Age
30

Biodata
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Address
Makassar Sulawesi Selatan

Photo Profile
 Just.png

[Update](#)

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

tampilan setelah tombol update diklik

Home About Kategori ▾ Berita ▾ Logout

Data Profile berhasil Diupdate

[Edit Profile](#)

(30)
Biodata Saya adalah
Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

Middleware

Dalam dunia IT, Middleware dikenal sebagai suatu sistem yang berperan sebagai penengah antara satu aplikasi dengan aplikasi yang lain agar mudah diintegrasikan. Pada konteks project Laravel, middleware digunakan untuk memfilter Request yang masuk di aplikasi kita. Biasanya digunakan untuk memproteksi route-route yang ada agar route tersebut benar benar diakses oleh user sesuai dengan privilege-nya. Atau bisa juga digunakan untuk "logging", atau mencatat semua aktivitas di dalam web kita.

disini kita akan melakukan pengamanan pada beberapa supaya beberapa halaman hanya bisa diakses jika usernya sudah login

sebagai contoh user yang belum masih dapat mengakses halaman lalui url di web browsernya



Categories

#	Name	Action
1	Sport	Edit Delete
2	IT	Edit Delete
3	Travel	Edit Delete

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

pada gambar diatas user yang belum login dan register masih dapat mangakses halaman localhost:8000/categories

disini kita akan mengamankan beberapa route yang hanya bisa diakses jika user auth, kita akan menggunakan middleware auth bawaan dari laravel untuk melakukan keamanan dari routenya

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;
use App\Http\Controllers\NewsController;
use App\Http\Controllers\ProfileController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');

Route::middleware('auth')->group(function () {
    //Create Data Category
    Route::get('/categories/categories',
[CategoriesController::class,
'create'])->name('category.create');

    Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');

    //Read Data Category
    Route::get('/categories', [CategoriesController::class,
'index'])->name('category.index');

    //Update Data Category
    Route::get('/categories/{id}/edit',
[CategoriesController::class, 'edit'])->name('category.edit');
    Route::put('/categories/{id}', [CategoriesController::class,
'update'])->name('category.update');

    //Delete Data Category
    Route::delete('/categories/{id}',
```

```

[CategoriesController::class,
'destroy'])->name('category.destroy');

//Profile
Route::get('/profile', [ProfileController::class,
'edit'])->name('profile.edit');
Route::put('/profile/{id}', [ProfileController::class,
'update'])->name('profile.edit');
});

//CRUD News
Route::resource('news', NewsController::class);

Auth::routes();

Route::get('/home', [App\Http\Controllers\HomeController::class,
'index'])->name('home');

```

pada code di atas kita menambahkan code pada web.php yaitu route group yang diberi middleware auth yang didalamnya terdapat route untuk CRUD categories, edit dan update profile yang kita proteksi jadi jika route tersebut diakses jika belum auth(login/register) maka akan dilempar ke halaman /login
 middleware auth sendiri terdapat di app\Http\Middleware\Authenticate.php dan middleware auth sudah terdaftar di app\Http\kernel.php

app\Http\Middleware\Authenticate.php

```

<?php

namespace App\Http\Middleware;

use Illuminate\Auth\Middleware\Authenticate as Middleware;

```

```

class Authenticate extends Middleware
{
    /**
     * Get the path the user should be redirected to when they
     * are not authenticated.
     *
     * @param \Illuminate\Http\Request $request
     * @return string|null
     */
    protected function redirectTo($request)
    {
        if (! $request->expectsJson()) {
            return route('login');
        }
    }
}

```

pada code diatas user yang belum auth akan diarahkan ke halaman route login

app\Http\kernel.php

```

<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
     * These middleware are run during every request to your

```

```
application.

*
 * @var array<int, class-string|string>
*/
protected $middleware = [
    // \App\Http\Middleware\TrustHosts::class,
    \App\Http\Middleware\TrustProxies::class,
    \Fruitcake\Cors\HandleCors::class,
    \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
    \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
    \App\Http\Middleware\TrimStrings::class,
    \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
];

/***
 * The application's route middleware groups.
 *
 * @var array<string, array<int, class-string|string>>
*/
protected $middlewareGroups = [
    'web' => [
        \App\Http\Middleware\EncryptCookies::class,
        \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\StartSession::class,
        //
        \Illuminate\Session\Middleware\AuthenticateSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],
];
```

```

] ,

'api' => [
    //
\ Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStatefu
l::class,
    'throttle:api',

\ Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],
];

/***
 * The application's route middleware.
 *
 * These middleware may be assigned to groups or used
individually.
 *
 * @var array<string, class-string|string>
 */

protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' =>

\ Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'cache.headers' =>

\ Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' =>

\ App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' =>

\ Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' =>

\ Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' =>

\ Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' =>

```

```
\Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,  
];  
}
```

pada code diatas middleware autenticate sudah di daftarkan di \$routeMiddleware pada file kerner.php

kemudian selain kita bisa mengamankan halamannya menggunakan middleware auth di dalam routenya kita juga bisa mengamankan halamannya melalui function di controllernya contoh disini saya akan melakukan middleware pada NewsController

app\Http\Controllers\NewsController.php

```
<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Models\News;  
use App\Models\Categories;  
use File;  
  
  
class NewsController extends Controller  
{  
    public function __construct()  
    {  
        $this->middleware('auth');  
    }  
    .....  
}
```

pada code diatas kita membuat construct() yaitu function yang pertama kali jalan dimana di dalamnya kita melakukan middleware auth pada setiap function di controllernya, tapi jika

kita ingin hanya beberapa function saja yang di middleware kita bisa menggunakan only atau except di middleware penjelasan lebih lanjut tentang only dan except di dibaca di link berikut : <https://laravel.com/docs/9.x/controllers#controller-middleware>

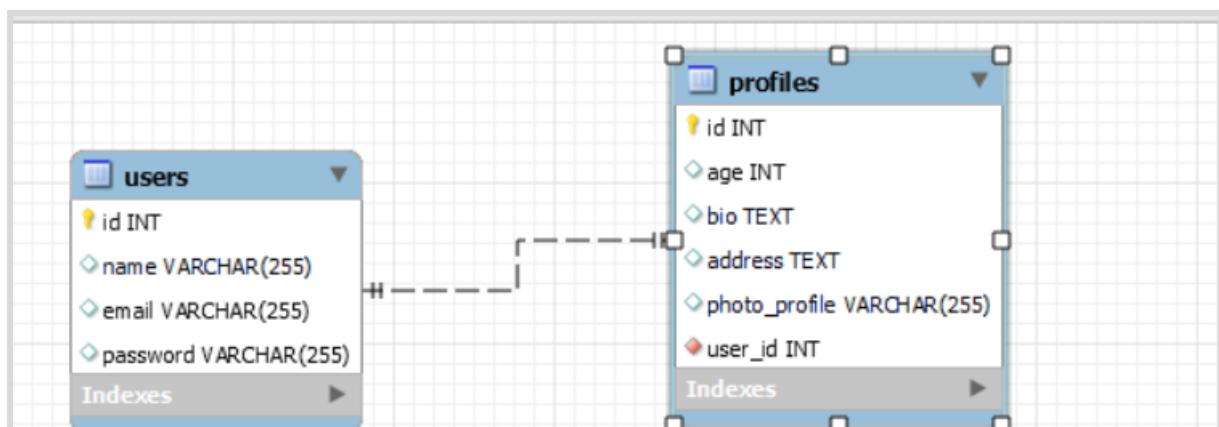
BAB 15 Laravel ORM Relationship

Laravel menyediakan ORM (Object-Relational Mapping) yang sangat powerful untuk mempermudah pengaksesan data dari database. Salah satu fitur ORM Laravel yang sangat berguna adalah Relationship atau hubungan antar tabel dalam database. Relationship memungkinkan kita untuk mengambil data dari beberapa tabel dengan menggunakan query yang lebih sederhana dan mudah dibaca.

Laravel menyediakan tiga jenis relationship utama, yaitu:

- One-to-One Relationship: Hubungan antara satu record pada satu tabel dengan satu record pada tabel lain
- One-to-Many Relationship: Hubungan antara satu record pada satu tabel dengan banyak record pada tabel lain.
- Many-to-Many Relationship: Hubungan antara banyak record pada satu tabel dengan banyak record pada tabel lain.

One to One



disini kita akan mencoba relasi one to one terlebih dahulu pada aplikasi portal berita yang sudah kita buat yaitu hubungan antar profile dan user

satu user memiliki satu profile dan satu profile dimiliki oleh user

disini kita akan mengatur modelnya terlebih dahulu

App/Models/User.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;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, 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',
    ];
}
```

```

/**
 * The attributes that should be cast.
 *
 * @var array<string, string>
 */
protected $casts = [
    'email_verified_at' => 'datetime',
];

public function profile()
{
    return $this->hasOne(Profile::class, 'user_id');
}

```

pada code diatas kita membuat function profile yang di dalamnya kita memanggil fungsi hasOne (memiliki satu) yang mengarah ke model Profile yang memiliki foreign key user_id. disini kita menggunakan hasOne pada model User karena table User merupakan table parent (table yang tidak memiliki foreign key) dari relasi ke table profile

app/Models/**Profile.php**

```

<?php

namespace App\Models;

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

class Profile extends Model
{
    use HasFactory;
    protected $table = 'profiles';
}

```

```

protected $fillable = ['age', 'bio', 'address',
'photo_profile', 'user_id'];

public function user()
{
    return $this->belongsTo(User::class);
}

}

```

pada code diatas kita membuat function user yang di dalamnya kita memanggil fungsi belongsTo (dimiliki oleh) yang mengarah ke model User yang memiliki foreign key user_id. disini kita menggunakan belongsTo (dimiliki oleh) pada model Profile karena table Profile merupakan table child (table yang foreign key) dari relasi ke table profile

setelah kita mengatur model kita kan mangatur view nya

resources/views/about.blade.php

```

@extends('layouts.master')

@section('content')
<div class="container padding" style="height: 80vh">
    @auth
        @if(session('success'))
            <div class="alert alert-success">
                {{ session('success') }}
            </div>
        @endif
        <a href="/profile" class="btn btn-primary btn-lg my-5">Edit
        Profile</a>

        <div class="row mx-0">
            <div class="col-4">
                @if ($profile->photo_profile)
                    
    @else
        
    @endif
</div>
<div class="col-8">
    <h1>{{ $profile->user->name }} ({{ $profile->age }})</h1>
    <small>{{ $profile->user->email }}</small>
    <h3>Biodata Saya adalah</h3>
    <p>
        {{ $profile->bio }}
    </p>
</div>
</div>

@endauth

@guest
    <h1>Belum Login</h1>
@endguest
</div>
@endsection

```

pada code di atas kita menampilkan name dan email berdasarkan relasi yang sudah kita buat pada function user di model profile

tampilan halaman about

The screenshot shows a user profile page. At the top, there is a navigation bar with links for Home, About, Kategori, Berita, and Logout. Below the navigation bar, there is a blue button labeled "Edit Profile". To the left is a circular profile picture with the text "Just Koding" on it. The profile information includes the name "rezky (30)", an email address "rezky@mail.com", and a bio: "Biodata Saya adalah". A large amount of placeholder text (Lorem Ipsum) follows the bio. At the bottom of the page, there is a copyright notice: "© Copyright 2018, All rights reserved. Design by FreeHTML5.co."

One To Many



disini kita akan mencoba relasi one to many pada aplikasi portal berita yang sudah kita buat yaitu hubungan antar news dan categories

satu categories memiliki banyak news dan satu news dimiliki oleh categories

disini kita akan mengatur modelnya terlebih dahulu

app/Http/Models/categories.php

```
<?php

namespace App\Models;

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

class Categories extends Model
{
    use HasFactory;
    protected $table = 'categories';
    protected $fillable = ['name'];

    public function news()
    {
        return $this->hasMany(news::class, 'category_id');
    }
}
```

pada code diatas kita membuat function news yang di dalamnya kita memanggil fungsi hasMany (memiliki banyak) yang mengarah ke model News yang memiliki foreign key category_id.

disini kita menggunakanhasOne (memiliki) pada model Categories karena table categories merupakan table parent (table yang tidak memiliki foreign key) dari relasi ke table news

app/Http/Models/**news.php**

```
<?php

namespace App\Models;

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

class News extends Model
{
    use HasFactory;
    protected $table = 'news';
    protected $fillable = ['title', 'content', 'thumbnail', 'category_id'];

    public function category()
    {
        return $this->belongsTo(Categories::class, 'category_id');
    }
}
```

pada code diatas kita membuat function category yang di dalamnya kita memanggil fungsi belongsTo (dimiliki oleh) yang mengarah ke model Categories yang memiliki foreign key category_id.

disini kita menggunakan belongsTo (dimiliki oleh) pada model News karena table news merupakan table child (table yang foreign key) dari relasi ke table categories

setelah kita mengatur model kita kan mangatur view, route, dan controller pada categories

resources/views/welcome.blade.php

```
@extends('layouts.master')

@section('content')
<div class="container padding">
    <div class="row mx-0">
        <div class="col-md-8 animate-box"
data-animate-effect="fadeInLeft">
            <div>
                <div class="fh5co_heading
fh5co_heading_border_bottom py-2 mb-4">News</div>
            </div>
            @forelse ($news as $item)
                <div class="row pb-4">
                    <div class="col-md-5">
                        <div class="fh5co_hover_news_img">
                            <div class="fh5co_news_img"></div>
                            <div></div>
                        </div>
                    </div>
                    <div class="col-md-7 animate-box">
                        <a href="/news/detail/{{ $item->id }}"
class="fh5co_magna py-2"> {{ $item->title}} </a>
                        <small>{{ $item->created_at->diffForHumans() }}</small>
                        <div class="fh5co_consectetur"> {{
str::limit($item->content, 200) }}</div>
                    </div>
                </div>
                @empty
                    <h3>Tidak Ada Berita</h3>
                @endforelse
            </div>
        </div>
    </div>
</div>
```

```

<div class="col-md-3 animate-box"
data-animate-effect="fadeInRight">
    <div>
        <div class="fh5co_heading
fh5co_heading_border_bottom py-2 mb-4">Kategori</div>
    </div>
    <div class="clearfix"></div>
    <div class="fh5co_tags_all">
        @forelse ($categories as $item)
            <a href="/categories/{{ $item->id }}"
class="fh5co_tagg">{{ $item->name }}</a>
        @empty
            <a href="">Tidak Ada Kategori</a>
        @endforelse
    </div>
    </div>
</div>
<div class="row mx-0 animate-box"
data-animate-effect="fadeInUp">
    <div class="col-12 text-center pb-4 pt-4">
        {{ $news->links() }}
    </div>
</div>
@endsection

```

pada kode diatas kita menambahkan link pada list categories dan list news untuk menampilkan detail dari berita

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;
use App\Http\Controllers\NewsController;
use App\Http\Controllers\ProfileController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');

Route::middleware('auth')->group(function () {
    //Create Data Category
    Route::get('/categories/categories',
[CategoriesController::class,
'create'])->name('category.create');

    Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');

    //Read Data Category
    Route::get('/categories', [CategoriesController::class,
'index'])->name('category.index');

    //Update Data Category
    Route::get('/categories/{id}/edit',
[CategoriesController::class, 'edit'])->name('category.edit');
    Route::put('/categories/{id}', [CategoriesController::class,
'update'])->name('category.update');

    //Delete Data Category
    Route::delete('/categories/{id}',
[CategoriesController::class,
```

```

'destroy'])->name('category.destroy');

//Profile
Route::get('/profile', [ProfileController::class,
'edit'])->name('profile.edit');
Route::put('/profile/{id}', [ProfileController::class,
'update'])->name('profile.edit');
});

Route::get('/categories/{id}', [CategoriesController::class,
'show'])->name('category.show');

Route::get('/news/detail/{id}', [NewsController::class,
'detail'])->name('news.detail');

//CRUD News
Route::resource('news', NewsController::class);

Auth::routes();

```

pada code diatas kita menambah satu route dengan url /categories/{id} yang mengarah ke CategoriesController function show dengan membawa parameter id dan route kedua mengarah dengan url /news/detail/{id} yang mengarah ke NewsController function detail dengan membawa parameter id

app/Http/Controllers/**CategoriesController.php**

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Models\Categories;

```

```

class CategoriesController extends Controller
{
    .....
    public function show($id)
    {
        $categories = Categories::find($id);

        return view('category.detail', ['categories' =>
$categories]);
    }
}

```

pada code diatas kita memanggil model categories use App\Models\Categories; dan kita membahkan satu function show yang memiliki parameter dimana kita akan mengambil data categories berdasarkan parameter id yang ditampung di parameter \$categories yang akan di passing ke subfolder category file detail.blade.php dengan membawa \$categories

resources/views/category/detail.blade.php

```

@extends('layouts.master')

@section('content')



# {{ $categories->name }}



@forelse ($categories->news as $item)


![...]({{ asset('/uploads/' . $item->thumbnail) }})


```

```

<h1>{{ $item->title }}</h1>
<p class="card-text">{{ Str::limit($item->content, 100) }}</p>

<small>{{ $item->created_at->diffForHumans() }}</small>
    <a href="/news/detail/{{ $item->id }}" class="btn btn-primary btn-block">Read More</a>
</div>
</div>
</div>
@empty
<h3>Tidak Ada Berita di Kategori Ini</h3>
@endif
</div>

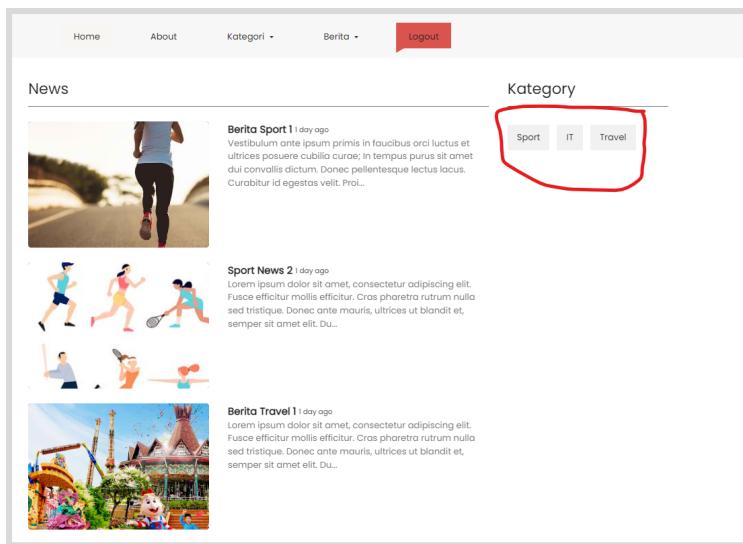
<a href="/" class="btn btn-secondary btn-block my-5">Home</a>
</div>

@endsection

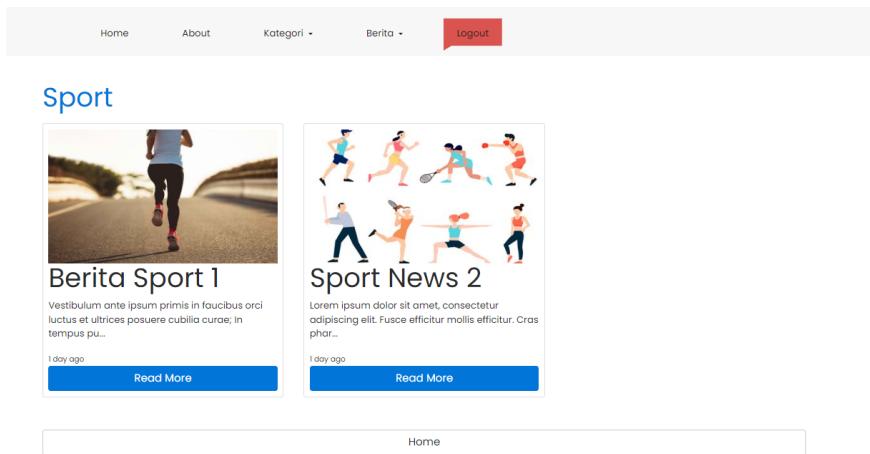
```

pada code diatas kita melakukan looping untuk menampilkan semua data berita berdasarkan relasi yang sudah yanghasMany yang sudah kita buat di model categories.php yang akan menampilkan semua berita berdasarkan category yang kita pilih pada parameter id

pada saat kita klik salah satu categories di halaman home



tampilan detail categories



© Copyright 2018. All rights reserved. Design by [FreeHTML5.co](#).

selanjutnya kita akan membuat relasi untuk belongsto nya tapi pertama-tama kita buat function dan view untuk halaman /news/detail/{id}

app/Http/Controller/**NewsController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\News;
```

```

use App\Models\Categories;
use File;

class NewsController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth')->except('detail');
    }

    .....
    public function detail($id)
    {
        $news = News::find($id);
        return view('detail', ['news' => $news]);
    }
}

```

pada code diatas kita menambahkan except pada middleware auth di `__construct` supaya function detail bisa di akses secara public tanpa perlu auth(login/register) kemudian kita juga menambahkan function detail untuk yang membawa parameter `$id` yang berfungsi mengambil data berdasarkan `$id` yang di tampung di variabel `$news` yang akan di passing ke halaman view `detail.blade.php`

resources/views/welcome.blade.php

```

@extends('layouts.master')

@section('content')
<div class="container padding">
    <div class="row mx-0">
        <div class="col-md-8 animate-box"
data-animate-effect="fadeInLeft">
            <div>

```

```

        <div class="fh5co_heading
fh5co_heading_border_bottom py-2 mb-4">News</div>
        </div>
        @forelse ($news as $item)
        <div class="row pb-4">
            <div class="col-md-5">
                <div class="fh5co_hover_news_img">
                    <div class="fh5co_news_img"></div>
                    <div></div>
                </div>
            </div>
            <div class="col-md-7 animate-box">
                <a href="/news/detail/{{ $item->id}}"
class="fh5co_magna py-2"> {{ $item->title}} </a>
                <small>{{ $item->created_at->diffForHumans() }}</small>
                <h6>
                    <span class="badge badge-primary">
                        {{ $item->category->name}}>
                    </span>
                </h6>
                <div class="fh5co_consectetur"> {{
Str::limit($item->content, 200) }}>
                </div>
            </div>
        </div>
        @empty
            <h3>Tidak Ada Berita</h3>
        @endforelse

        </div>
        <div class="col-md-3 animate-box"
data-animate-effect="fadeInRight">
            <div>
                <div class="fh5co_heading
fh5co_heading_border_bottom py-2 mb-4">Kategory</div>

```

```

        </div>

        <div class="clearfix"></div>

        <div class="fh5co_tags_all">
            @forelse ($categories as $item)
                <a href="/categories/{{ $item->id }}">
                    {{ $item->name }}</a>
                @empty
                    <a href="">Tidak Ada Kategori</a>
                @endforelse
            </div>
        </div>
    </div>
    <div class="row mx-0 animate-box"
data-animate-effect="fadeInUp">
        <div class="col-12 text-center pb-4 pt-4">
            {{ $news->links() }}
        </div>
    </div>
</div>
@endsection

```

di file welcome.blade.php kita nambahkan code untuk menampilkan nama categorinya berdasarkan relasi belongsTo yang sudah dibuat di model news

resources/views/**detail.blade.php**

```

@extends('layouts.master')

@section('content')


 }})

# {$news->title}</h1>


```

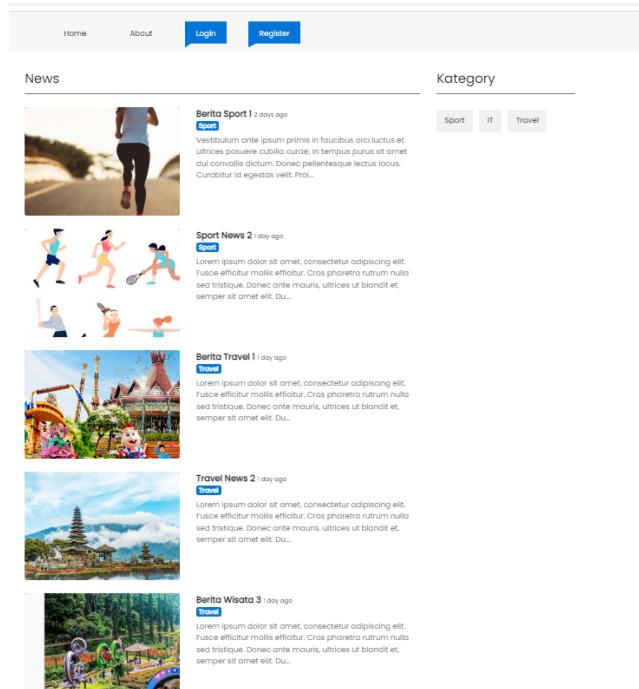
```

<span class="badge badge-primary">
    <a href="/categories/{$news->category->id}" style="text-decoration: none;">{$news->category->name}</a>
</span>
<hr>
<p>{$news->content}</p>
<hr>
<a href="/" class="btn btn-lg btn-secondary">Home</a>
</div>
</div>
@endsection

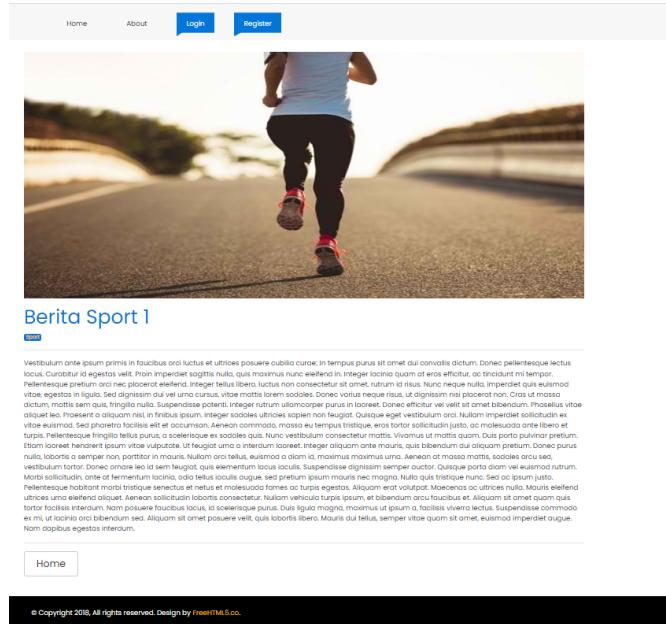
```

pada code di atas kita mampailkan detail data dari news berdasakan \$news yang sudah di passing pada function detail di newsController dan juga kita menampilkan nama category berdasarkan relasi belongsTo yang sudah di buat di model News

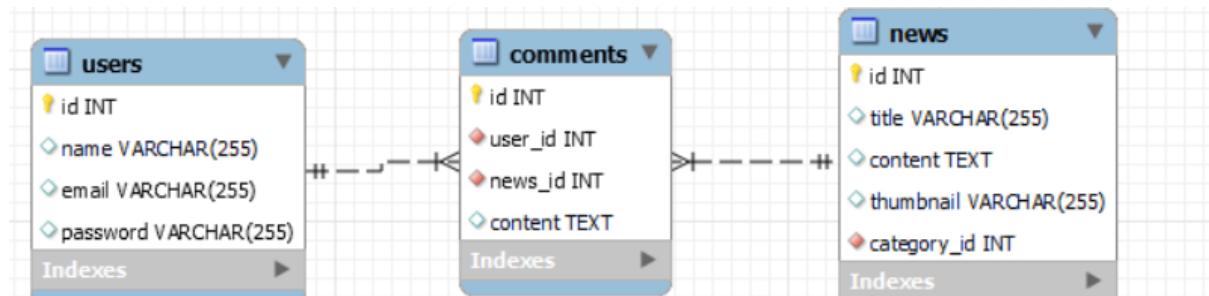
tampilan halaman home/welcome



tampilan pada saat link judul berita di klik



Many to Many



terakhir kita akan mencoba relasi many to many pada aplikasi portal berita yang sudah kita buat yaitu hubungan antar user, news dan comments

satu users memiliki banyak melakukan comments di banyak berita

satu news memiliki banyak user yang melakukan comments

sebelum kita atur buat relasi di setiap model pertama-tama kita buat dulu fungsi untuk menambahkan sebuah comment kita atur dulu route, controller, model dan viewnya

pertama-tama kita buat dulu controller dan modelnya

```
php artisan make:controller CommentController
```

```
php artisan make:model Comment
```

kemudian kita atur routesnya

routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\CategoriesController;
use App\Http\Controllers\NewsController;
use App\Http\Controllers\ProfileController;
use App\Http\Controllers\CommentController;

Route::get('/', [DashboardController::class,
'home'])->name('home');

Route::get('/about', [DashboardController::class,
'about'])->name('about');

Route::middleware('auth')->group(function () {
    //Create Data Category
    Route::get('/categories/categories',
[CategoriesController::class,
'create'])->name('category.create');
    Route::post('/categories', [CategoriesController::class,
'store'])->name('category.store');

    //Read Data Category
    Route::get('/categories', [CategoriesController::class,
'index'])->name('category.index');

    //Update Data Category
    Route::get('/categories/{id}/edit',
```

```

[CategoriesController::class, 'edit'])->name('category.edit');

Route::put('/categories/{id}', [CategoriesController::class,
'update'])->name('category.update');

//Delete Data Category
Route::delete('/categories/{id}',
[CategoriesController::class,
'destroy'])->name('category.destroy');

//Profile
Route::get('/profile', [ProfileController::class,
'edit'])->name('profile.edit');

Route::put('/profile/{id}', [ProfileController::class,
'update'])->name('profile.edit');

Route::post('/comment/{news_id}', [CommentController::class,
'store'])->name('comment.store');
});

Route::get('/categories/{id}', [CategoriesController::class,
'show'])->name('category.show');

Route::get('/news/detail/{id}', [NewsController::class,
'detail'])->name('news.detail');

//CRUD News
Route::resource('news', NewsController::class);

Auth::routes();

```

pada code diatas kita memanggil CommentController use

App\Http\Controllers\CommentController; kemudianambahkan satu route yang masuk ke route group middleware auth dengan url /comment/{id} yang mengarah ke CommentController function store dengan method post dan juga dengan membawa parameter id {id}

lalu kita atur model comment

app\Models\Comment.php

```
<?php

namespace App\Models;

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

class Comment extends Model
{
    use HasFactory;
    protected $table = 'comments';
    protected $fillable = ['news_id', 'user_id', 'content'];
}
```

pada code di atas kita membuat protected \$table comments sesuai dengan nama table comments yang ada di database dan membuat \$fillable yang kita samakan dengan kolom yang ada di table comments

selanjutkan kita akan membuat tampilan tambah komentar pada halaman detail.blade.php

resources/views/detail.blade.php

```
@extends('layouts.master')

@section('content')
<div class="container padding">
    
    <div class="my-3">
        <h1 class="text-primary"> {{$news->title}} </h1>
        <span class="badge badge-primary">
            <a href="/categories/{{$news->category->id}}">
```

```

style="text-decoration: none;">{{ $news->category->name }}</a>
    </span>
    <hr>
    <p>{{ $news->content }}</p>
    <hr>
    <h3>List Komentar</h3>
    {{-- List Komentar --}}
    @auth
    <hr>
    <form action="/comment/{{ $news->id }}" method="POST">
        @csrf
        <div class="form-group">
            <textarea name="content" id="content"
placeholder="Isi Komentar" class="form-control" cols="30"
rows="10"></textarea>
        </div>
        @error('content')
            <div class="alert alert-danger">{{ $message
}}</div>
        @enderror
        <input type="submit" class="btn btn-primary btn-lg"
value="kirim">
    </form>
    @endauth
    <hr>
    <a href="/" class="btn btn-lg btn-secondary">Home</a>
</div>
</div>
@endsection

```

pada code di atas kita membuat text List Komentar untuk menampilkan List komentar berdasarkan relasi model yang akan kita buat nanti kemudian kita membuat form untuk menambah komentar yang akan tampil jika user login @auth kemudian formnya memiliki attribute method post dan action /comment/{{ \$news->id }} untuk mengambil id dari

berita yang sudah sesuai dengan route yang kita daftarkan di web.php yang didalam formnya terdapat validasi dan inputan untuk content

selanjutkan kita akan mengatur controller nya di CommentController

app\Http\Controllers\CommentController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Comment;
use Illuminate\Support\Facades\Auth;

class CommentController extends Controller
{
    public function store($id, Request $request)
    {
        $request->validate([
            'content' => 'required',
        ]);

        $id_user = Auth::id();

        $comment = new Comment;

        $comment->content = $request->input('content');
        $comment->user_id = $id_user;
        $comment->news_id = $id;

        $comment->save();

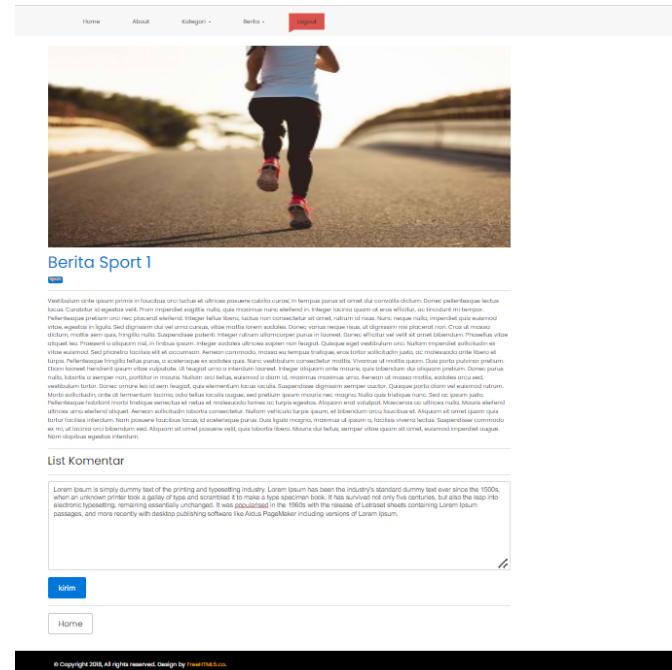
        return redirect()->back()->with('success', 'Data Comment berhasil Ditambahkan');
    }
}
```

```
}
```

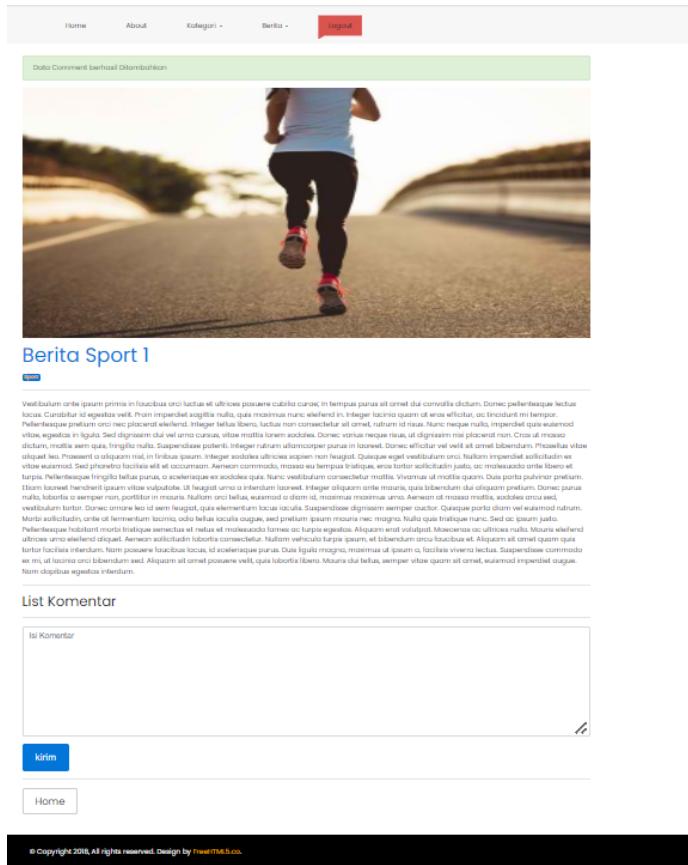
pada kode di atas kita memanggil model comment `use App\Models\Comment;` dan juga kita memanggil fungsi auth bawaan laravel `use Illuminate\Support\Facades\Auth;`

di dalam function store kita akan terdapat parameter id dan juga request \$request untuk melakukan request inputan di view kemudian kita membuat variabel \$id_user yang menampung id dari user yang auth `$id_user = Auth::id()` setelah itu untuk kolom content pada comment kita insert dengan request input name comment yang terdapat di view `$comment->content = $request->input('content');` dan untuk kolom user_id kita insert berdasarkan user yang auth `$comment->user_id = $id_user;` dan untuk kolom news_id kita input berdasarkan parameter id `$comment->news_id = $id;` setelahnya kita akan melakukan save untuk menyimpan data ke database lalu kita akan mengarah ke halaman detail berita sebelumnya `return redirect()->back()` dengan membawa pesan flash message `->with('success', 'Data Comment berhasil Ditambahkan');`

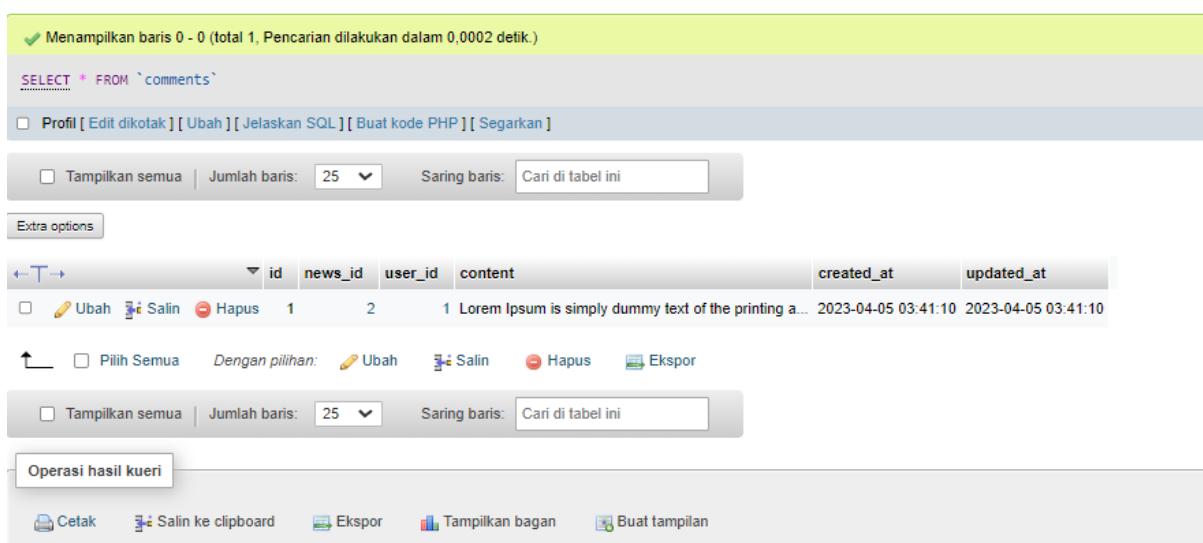
disini kita akan melakukan input comment



setelah tombol kirim di klik



data yang kita input masuk ke table comment di database



setelah kita sudah membuat fungsi tambah comment selanjutnya kita akan membuat relasi antar modelnya untuk menampilkan list komentar pada halaman detail berita

app/Models/News.php

```
<?php

namespace App\Models;

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

class News extends Model
{
    use HasFactory;
    protected $table = 'news';
    protected $fillable = ['title', 'content', 'thumbnail',
    'category_id'];

    public function category()
    {
        return $this->belongsTo(Categories::class,
    'category_id');
    }

    public function comments()
    {
        return $this->hasMany(Comment::class, 'news_id');
    }
}
```

pada code di atas kita membuat relasihasMany(mempuyai banyak) yang terhubung ke model Comment dengan foreign key news_id

app/Models/User.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;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, 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',
    ];
}
```

```

/**
 * The attributes that should be cast.
 *
 * @var array<string, string>
 */
protected $casts = [
    'email_verified_at' => 'datetime',
];

public function profile()
{
    return $this->hasOne(Profile::class, 'user_id');
}

public function news()
{
    return $this->belongsToMany(News::class, 'comments',
'user_id', 'news_id');
}

```

pada code diatas kita menambahkan function news yang berfungsi menampilkan berita apa saja yang di comment user tersebut dengan menggunakan relasi belongsToMany(memiliki banyak) dengan table pivot 'comment' foreign key 1 'user_id' dan foreign key 2 'news_id'

app/Models/**Comment.php**

```

<?php

namespace App\Models;

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

```

```

class Comment extends Model
{
    use HasFactory;
    protected $table = 'comments';
    protected $fillable = ['news_id', 'user_id', 'content'];

    public function user()
    {
        return $this->belongsTo(User::class, 'user_id');
    }
}

```

pada code diatas kita membuat relasi ke user yaitu belongsTo(dimiliki oleh) yang berelasi ke model user foreign user_id untuk memanipulkan nama user yang melakukan comment

resources/views/Detail.blade.php

```

@extends('layouts.master')

@section('content')


@if(session('success'))
    <div class="alert alert-success">
        {{ session('success') }}
    </div>
@endif

<div class="my-3">
    <h1 class="text-primary"> {{$news->title}} </h1>
    <span class="badge badge-primary">
        <a href="/categories/{{ $news->category->id}}">{{ $news->category->name}} </a>


```

```

        </span>
        <hr>
        <p>{ ${news->content} }</p>
        <hr>

        <h3>List Komentar</h3>
        {{-- List Komentar --}}
        @forelse ($news->comments as $item)
            <div class="media">
                
                <div class="media-body">
                    <h5 class="mt-0 text-info">{{$item->user->name}}</h5>
                    {{$item->content}}
                    @auth
                    <hr>
                    <form action="" method="post">
                        @csrf
                        <div class="form-group">
                            <textarea name="content" id="content" placeholder="Isi Komentar" class="form-control" cols="10" rows="5"></textarea>
                        </div>
                        @error('content')
                            <div class="alert alert-danger">{{$message }}</div>
                        @enderror
                        <input type="submit" class="btn btn-info btn-sm" value="Reply">
                    </form>
                    @endauth
                </div>
            </div>
        @empty
    
```

```

<h6>Tidak Komentar</h6>
@endforelse

@auth
<hr>
<form action="/comment/{{ $news->id }}" method="POST">
    @csrf
    <div class="form-group">
        <textarea name="content" id="content"
placeholder="Isi Komentar" class="form-control" cols="30"
rows="10"></textarea>
    </div>
    @error('content')
        <div class="alert alert-danger">{{ $message
}}</div>
    @enderror
    <input type="submit" class="btn btn-primary btn-lg"
value="kirim">
</form>
@endauth
<hr>
<a href="/" class="btn btn-lg btn-secondary">Home</a>
</div>
</div>
@endsection

```

pada kode diatas kita membuat looping berdasarkan relasi hasMany pada model news untuk menampilkan dengan menggunakan forelse @forelse (\$news->comments as \$item) jadi jika ada nya ada maka akan tampil list komentar tapi jika tidak ada data komentar maka akan menampilkan text <h6>Tidak Komentar</h6> didalam forelse terdapat list komentar yang nemapilkan photo_profile berdasarkan relasi belongsto di model comment dan belong to profile di model user {{asset('/uploads/profile/' .
\$item->user->profile->photo_profile)}} dan juga kita menampilkan nama user berdasarkan relasi belongsto di model comment {{\$item->user->name}} dan menampilkan content dari komment {{\$item->content}} dan pada code diatas kita juga membuat inputan form untuk reply yang hanya tampil jika sudah auth(login/register)

resources/views/about.blade.php

```
@extends('layouts.master')

@section('content')


@auth
@if(session('success'))


{{ session('success') }}


@endif
Edit
Profile</a>



@if \(\$profile->photo\_profile\)
!\[Cinque
Terre\]\({{asset\('/uploads/profile/' .
\$profile->photo\_profile\)}}\)


@forelse \(\$profile->user->news as \$item\)
- <a class="text-primary"
href="/news/detail/{{ \$item->id}}">{{ \$item->title}}</a></li>
@empty
- >belum ada berita yang
dikomentari</li>
@endforelse


@else
!\[Cinque Terre\]\(https://dummyimage.com/480x320/000/fff\)
@endif


```

```

<div class="col-8">
    <h1>{{ $profile->user->name }} ({{ $profile->age }})</h1>
    <small>{{ $profile->user->email }}</small>
    <h3>Biodata Saya adalah</h3>
    <p>
        {{ $profile->bio }}
    </p>
</div>
</div>

@endauth

@guest
    <h1>Belum Login</h1>
    @endguest
</div>
@endsection

```

pada file halaman about diatas kita menambahkan code untuk menampilkan berita apa saja yang sudah dikomentari oleh user tersebut disini kita melakukan looping berupa list berdasarkan relasi belongsTo user pada model profile dan relasi belongsMany news pada model user @forelse (\$profile->user->news as \$item) didamana didalam looping forelse nya kita menampilkan judul berita yang berbentuk link yang mengarah ke detail berita id }}"> {{ \$item->title }} dan jika datanya kosong @empty maka akan menampilkan <li class="">belum ada berita yang dikomentari

tampilan list komentar pada halaman detail berita

The screenshot shows a web page with a header containing navigation links: Home, About, Kategori, Berita, and Log In. Below the header is a large image of a person running. The main content area has a title "Berita Sport 1". Underneath the title is a long block of placeholder text. Below this is a section titled "List Komentar" with a sub-section "Isi Komentar". A red circle highlights the "Isi Komentar" input field and its associated "Simpan" button. Another red circle highlights the "Kirim" button below it. At the bottom of the page is a copyright notice: "© Copyright 2018. All rights reserved. Design by [meantabu](#).css".

tampilan list berita jika belum ada komentar

The screenshot shows a web page with a header containing navigation links: Home, About, Kategori, Berita, and Log In. Below the header is a grid of five small sport-related illustrations. The main content area has a title "Sport News 2". Underneath the title is a long block of placeholder text. Below this is a section titled "List Komentar" with a sub-section "Isi Komentar". A red circle highlights the "Isi Komentar" input field and its associated "Simpan" button. Another red circle highlights the "Kirim" button below it. At the bottom of the page is a copyright notice: "© Copyright 2018. All rights reserved. Design by [meantabu](#).css".

tampilan list berita yang pernah di komentar oleh user di halaman about

Home About Kategori ▾ Berita ▾ Logout

[Edit Profile](#)

rezky (30)
rezky@mail.com

Biodata Saya adalah

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

• Berita Sport 1

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

tampilan jika user belum pernah melakukan komentar di berita manapun di halaman about

Home About Kategori ▾ Berita ▾ Logout

[Edit Profile](#)

user123 (27)
user@user.com

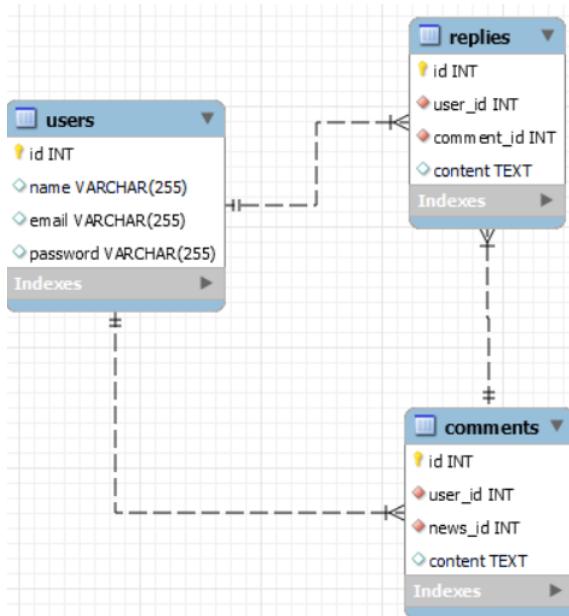
Biodata Saya adalah

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem ipsum as their default model text, and a search for lorem ipsum will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

• belum ada berita yang dikomentari

© Copyright 2018, All rights reserved. Design by [FreeHTML5.co](#).

Tantangan Bersertifikat



Pada challenge kali ini kamu akan diminta untuk menambahkan fitur reply pada aplikasi portal berita yang sudah dibuat. fitur reply di buat di halaman detail.blade.php berikut fitur fitur yang harus dibuat

- buat route untuk fungsi tambah reply di web.php dan buat controller dan model untuk reply
- buat fungsi tambah reply pada controller reply
- buat protected table dan fillable pada model reply
- buat relasi pada model reply, comment
- buat fitur untuk tambah reply yang datanya masuk ke database table replies pada saat tombol submit reply di klik di halaman detail.blade.php

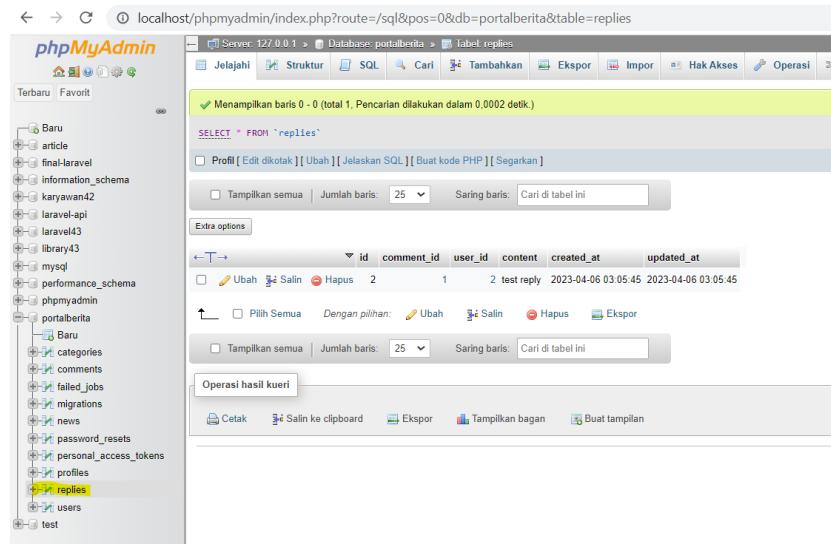
tampilan reply di halaman detail berita

The screenshot shows a news article titled "Berita Sport 1". At the top, there is a navigation bar with links for Home, About, Kategori, Berita, and Reply. Below the navigation is a large image of a person jogging. The main content area contains several paragraphs of text. At the bottom of the article, there is a "List Komentar" section. A red circle highlights a reply box labeled "test reply" with a "Reply" button. Below this, there is a "Beri Komentar" input field and a "Kirim" button.

tampilan pada saat tombol reply di klik

This screenshot is identical to the previous one, but the "Reply" button has been clicked. A red circle highlights the now-expanded reply form. The "test reply" text is still present in the input field, and the "Reply" button is visible below it. The rest of the page, including the article text and navigation bar, remains the same.

data yang kita submit masuk ke database



Teknis Pengumpulan

Berkas atau file yang perlu Anda siapkan adalah:

1. push project laravel yang telah selesai di repositori gitlab yang sudah di buat sebelumnya **IM Ebook portal berita** lakukan metode git add . kemudian commit -m "selesai" kemudian push git push origin main

```
PS D:\ebook\im-ebook-portal-berita\portal-berita> git add .
warning: LF will be replaced by CRLF in portal-berita/app/Http/Controllers/CommentController.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in portal-berita/app/Http/Controllers/ReplyController.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in portal-berita/app/Models/Comment.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in portal-berita/app/Models/Reply.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in portal-berita/routes/web.php.
The file will have its original line endings in your working directory
● ps D:\ebook\im-ebook-portal-berita\portal-berita> git commit -m "selesai"
[main 3dc78aa] selesai
 6 files changed, 53 insertions(+), 3 deletions(-)
PS D:\ebook\im-ebook-portal-berita\portal-berita> git push origin main
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 16 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (16/16), 1.83 KiB | 1.83 MiB/s, done.
Total 16 (delta 12), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/rezkyputra/im-ebook-portal-berita.git
 889f439..3dc78aa  main -> main
● PS D:\ebook\im-ebook-portal-berita\portal-berita> []
```

2. buat demo aplikasi, bisa menggunakan screen recorder **Catatan**: Video demo aplikasi, disiapkan dalam bentuk link ke situs penyimpanan Youtube
3. **Untuk mendapatkan sertifikat, silahkan kumpulkan hasil pekerjaan Kamu (link repositori github atau gitlab.. pastikan bisa diakses public untuk pengecekan oleh tim sanbercode) di menu submit challenge**