



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web Pertemuan 1

Zuhri Halim, S.Kom., M.Kom.

Outline

1. Pendahuluan
2. Refleksi
3. Sejarah Web
4. Perkembangan Web
5. Jenis-Jenis Web
6. Roadmap
7. Contoh Penggunaan Web



Pendahuluan: Sistem Penilaian

2 SKS
(14 pertemuan).

Absensi 10%.
Maksimal tidak
hadir 3 kali.

Tugas 25%.

UTS 25%.
Jadwal UTS di
pertemuan ke-8.

UAS 40%.
Jadwal UAS di
pertemuan ke-15.

Pendahuluan: Capaian Hasil Belajar

Mahasiswa mampu menjelaskan sejarah dan perkembangan teknologi web terkini, dan perbedaan web.

Tugas 25%

- Tugas Individu1
- Tugas Individu2
- Tugas Individu3
- Tugas Kelompok

UTS

- Tugas Individu1
- Tugas Individu2
- Tugas Individu3
- Tugas Kelompok

UAS

Refleksi

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ

Bacalah dengan (menyebut) nama Tuhanmu Yang menciptakan (QS. Al
Alaq: 1)

Sejarah Web: World Wide Web (WWW)

Tim Berners-Lee menciptakan sebuah World Wide Web dan dengan waktu bersamaan bekerja di CERN 1989. Dia mengusulkan untuk sebuah konsep terkait teknologi untuk menghubungkan informasi-informasi umum. Dia mengembangkan server web pertama, browser web pertama dan protocol pemformatan dokumen, yang disebut Hypertext Markup Language (HTML).

Perkembangan Web: Versi 1.0

Web 1.0 merupakan teknologi Web yang dibangun untuk pengaksesan informasi dan memiliki sifat yang sedikit interaktif dimana pengunjung hanya dapat membaca karena tidak adanya proses untuk menginput data.



Perkembangan Web: Versi 2.0

Web 2.0 dicetuskan pertama kali oleh O'Reilly Media pada tahun 2003, dan dipopulerkan pada konferensi web 2.0 pertama pada tahun 2004. Web 2.0 adalah revolusi bisnis di industri komputer yang dipicu oleh penggunaan internet sebagai platform.



Perkembangan Web: Versi 2.0

Web 2.0 dicetuskan pertama kali oleh O'Reilly Media pada tahun 2003, dan dipopulerkan pada konferensi web 2.0 pertama pada tahun 2004. Web 2.0 adalah revolusi bisnis di industri komputer yang dipicu oleh penggunaan internet sebagai platform.



Perkembangan Web: Versi 3.0

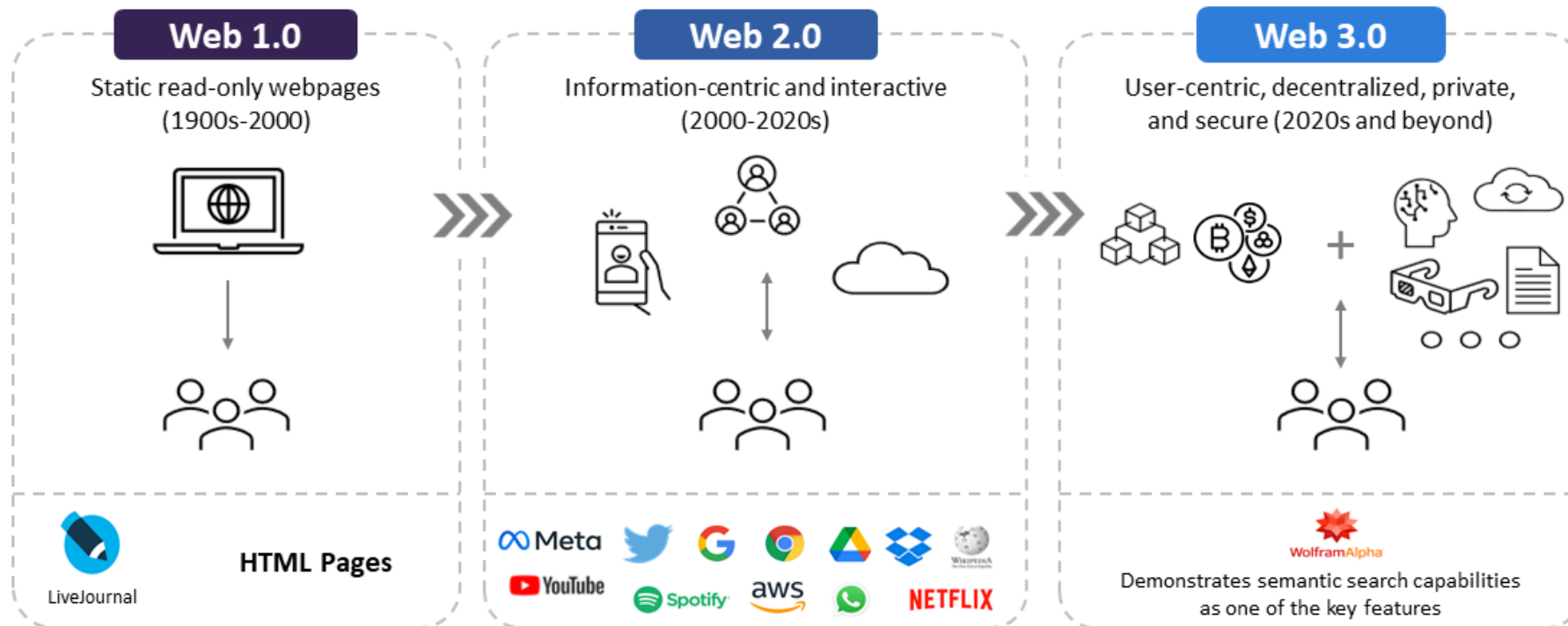
Web 3.0 adalah sekumpulan teknologi yang menawarkan cara baru efisien dalam membantu komputer mengorganisasi dan menarik kesimpulan dari data online. Web 3.0 memiliki fitur seperti ubikuitas, web semantic, artificial intelligence, dan grafik 3D.

1. Ubikuitas: sebuah web berada atau memiliki kapasitas untuk dapat diakses dimana-mana. Contoh, penggunaan seperti web of things.
2. Web Semantic: sebuah web dapat mempelajari hubungan antar kata, berbagi data seperti konten, transaksi, dan hubungan antara orang-orang.
3. Artificial intelligence: sebuah web dapat menguraikan makna dan sentiment dari kumpulan data, dan menciptakan mesin yang cerdas.
4. Web spasial dan grafik 3D: web dapat menampilkan gambar secara 3D dan imersif/nyata.

Perkembangan Web dan Perbedaan



Web 3.0 is the evolution of the internet towards user-centric intelligent services

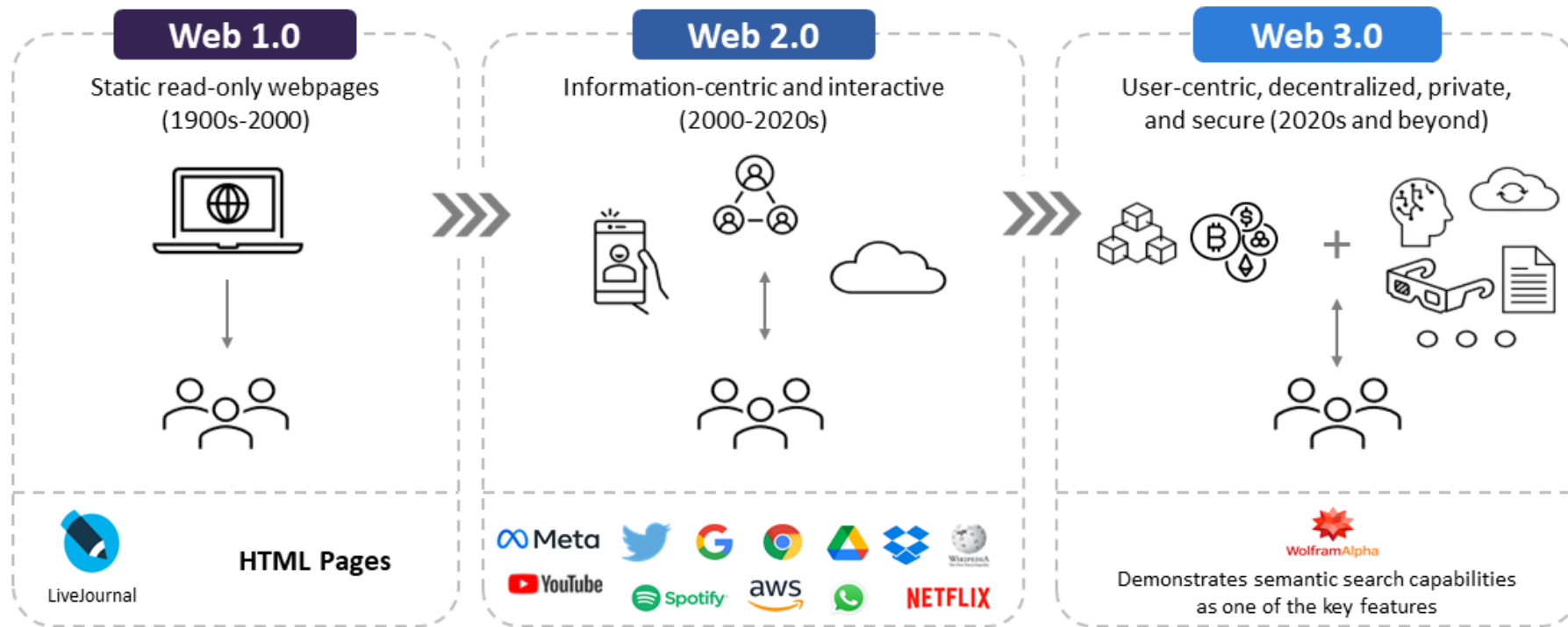


Source: GlobalData FutureTech Series Report

Perkembangan Web dan Perbedaan



Web 3.0 is the evolution of the internet towards user-centric intelligent services



Source: GlobalData FutureTech Series Report

Jenis-Jenis Web

Blog

E-commerce

Bisnis

Pemerintah

Hiburan

Portofolio

Portal
Berita

Pendidikan

Sosial

Komunitas

Referensi

1. https://en.wikipedia.org/wiki/History_of_the_World_Wide_Web

TERIMA KASIH



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web Pertemuan 2

Zuhri Halim, S.Kom., M.Kom.

Outline

1. Pendahuluan
2. Refleksi
3. Folder Structure
4. Naming Pages, Directories, and Supporting Files
5. Localhost & Web Server (Hosting/Cpanel))
6. MVC Structure



Pendahuluan: Capaian Hasil Belajar

Mahasiswa mampu menganalisis struktur web dan manajemen file dilocalhost/cpanel

Refleksi

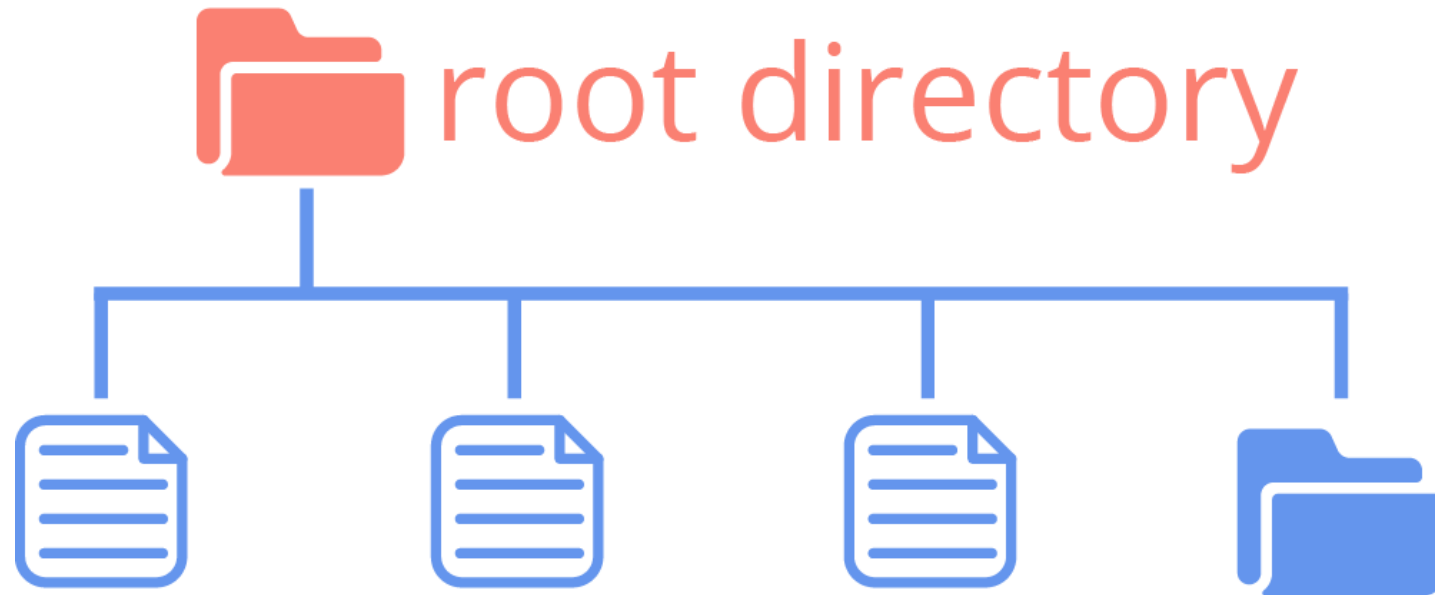
لَهُ مُعَقِّبَاتٌ مِنْ بَيْنِ يَدَيْهِ وَمِنْ خَلْفِهِ يَحْفَظُونَهُ مِنْ أَمْرِ اللَّهِ ^{قُلْ} إِنَّ اللَّهَ لَا يُغَيِّرُ مَا بِقَوْمٍ حَتَّى يُغَيِّرُوا مَا بِأَنْفُسِهِمْ ^{قُلْ} وَإِذَا أَرَادَ اللَّهُ بِقَوْمٍ سُوءًا فَلَا مَرَدَّ لَهُ ^ج وَمَا لَهُمْ مِنْ دُونِهِ مِنْ وَالٍ



Baginya (manusia) ada malaikat-malaikat yang selalu menjaganya bergiliran, dari depan dan belakangnya. Mereka menjaganya atas perintah Allah. **Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum sebelum mereka mengubah keadaan diri mereka sendiri.** Dan apabila Allah menghendaki keburukan terhadap suatu kaum, maka tak ada yang dapat menolaknya dan tidak ada pelindung bagi mereka selain Dia (QS. Ar-Ra'd: 11)

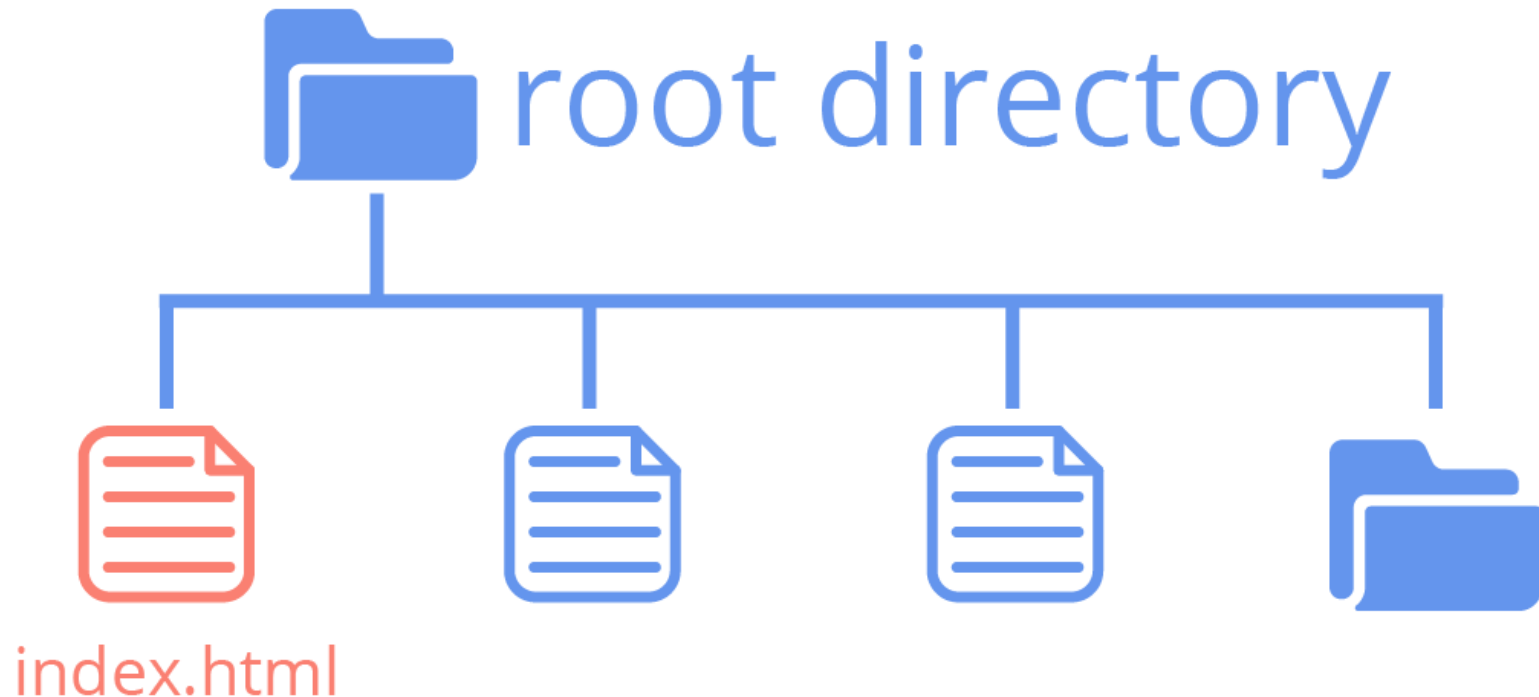
Folder Structure

Sebuah website yang terdiri dari beberapa file seperti HTML, CSS, Javascript, dan file yang terkait dengan media seperti images, video, dan audio. Dalam file-file tersebut menjadi satu dan dinamakan “folder structure”. Folders dan files dapat di upload secara online pada suatu server.



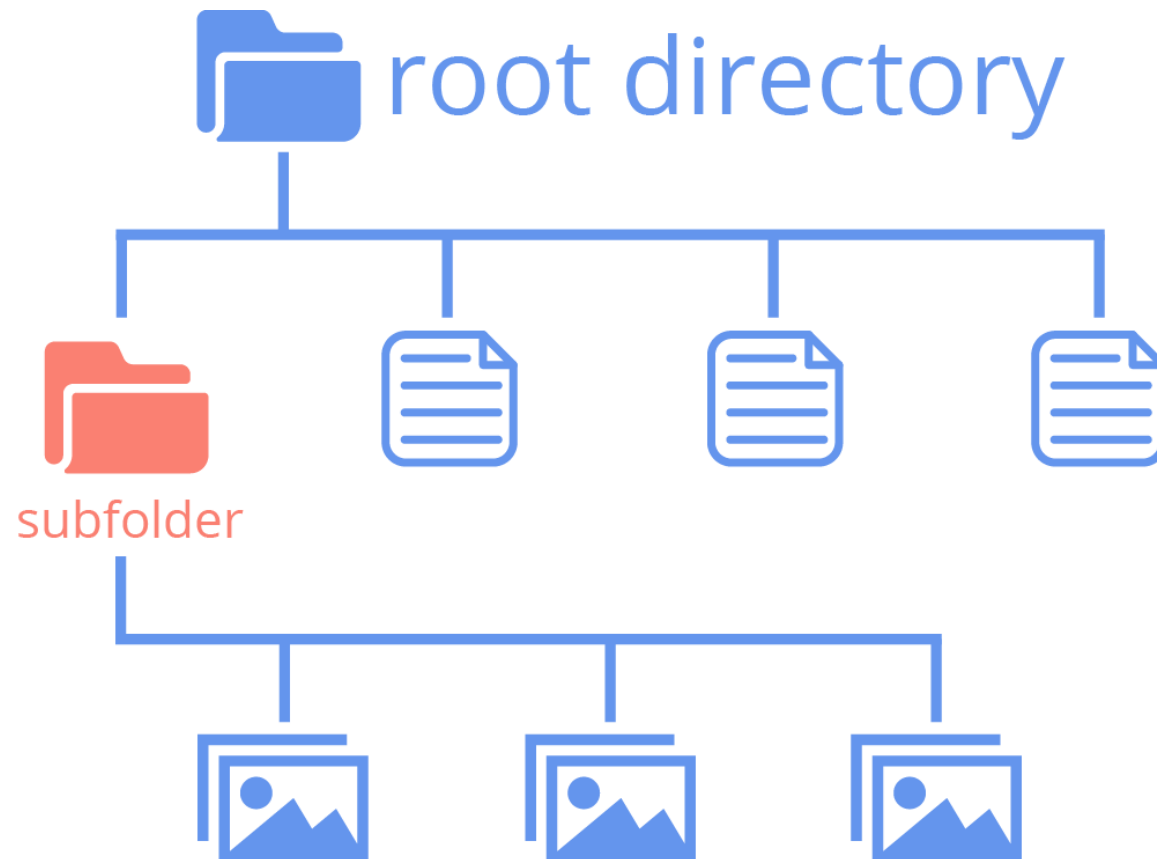
Folder Structure: Home Page

Sebuah website selalu mempunyai halaman home page dan dalam root directory pada folder website tersebut memiliki file index.html. Website dapat diakses jika pada root directory memiliki file index.html.



Folder Structure: Subfolder

Sebuah website selalu mempunyai folder dan subfolder. Contoh penggunaan subfolder yaitu images, yang dimana subfolder tersebut untuk menampung file-file gambar.



Folder Structure: Benefits

Keuntungan-keuntungan dapat mengatur folder, subfolder dan file-file dengan baik, antara lain sebagai berikut:

1. Understanding: programmer/tim dapat mudah mengerti struktur file-file pada website.
2. Flexibility & accessibility: setiap file dapat menjadi lebih fleksibel dimanipulasi, karena secara struktur file dapat dimengerti. Serta setiap struktur file harus mudah diakses.
3. Search optimization: mudah terindeks pada mesin pencarian.
4. Future growth & change: sistem tidak dapat mengubah file secara otomatis pada struktur file di website. Oleh karena itu, struktur file yang baik dapat memudahkan proses pengembangan lebih lanjut.

Folder Structure: The Hidden Semantics of Sites

Tujuan semantics organizations terhadap sebuah website, bahwa setiap prosedur dan klasifikasi file seperti HTML, komponen CSS, direktori, dan variasi-variasi dalam file logika memiliki aturan yang jelas. Sehingga setiap orang dalam tim mengerti dan mengikuti prosedur terhadap aturan-aturan dalam struktur website.

Naming Pages, Directories, and Supporting Files

- **Use language that anyone can understand:** dalam penamaan files dan directories harus mudah dibaca dan dimengerti. Jika nama files dan directories, maka gunakan tanda “-”, karena bertujuan dari files dan directories tersebut dapat di analisis dan dapat lebih mudah terindeks oleh search engines.
- **Visibile site structure:** penamaan files dan directories harus mudah dibaca dan dapat dimengerti, karena dapat berpengaruh terhadap semantic guide pada sebuah website. Contoh:

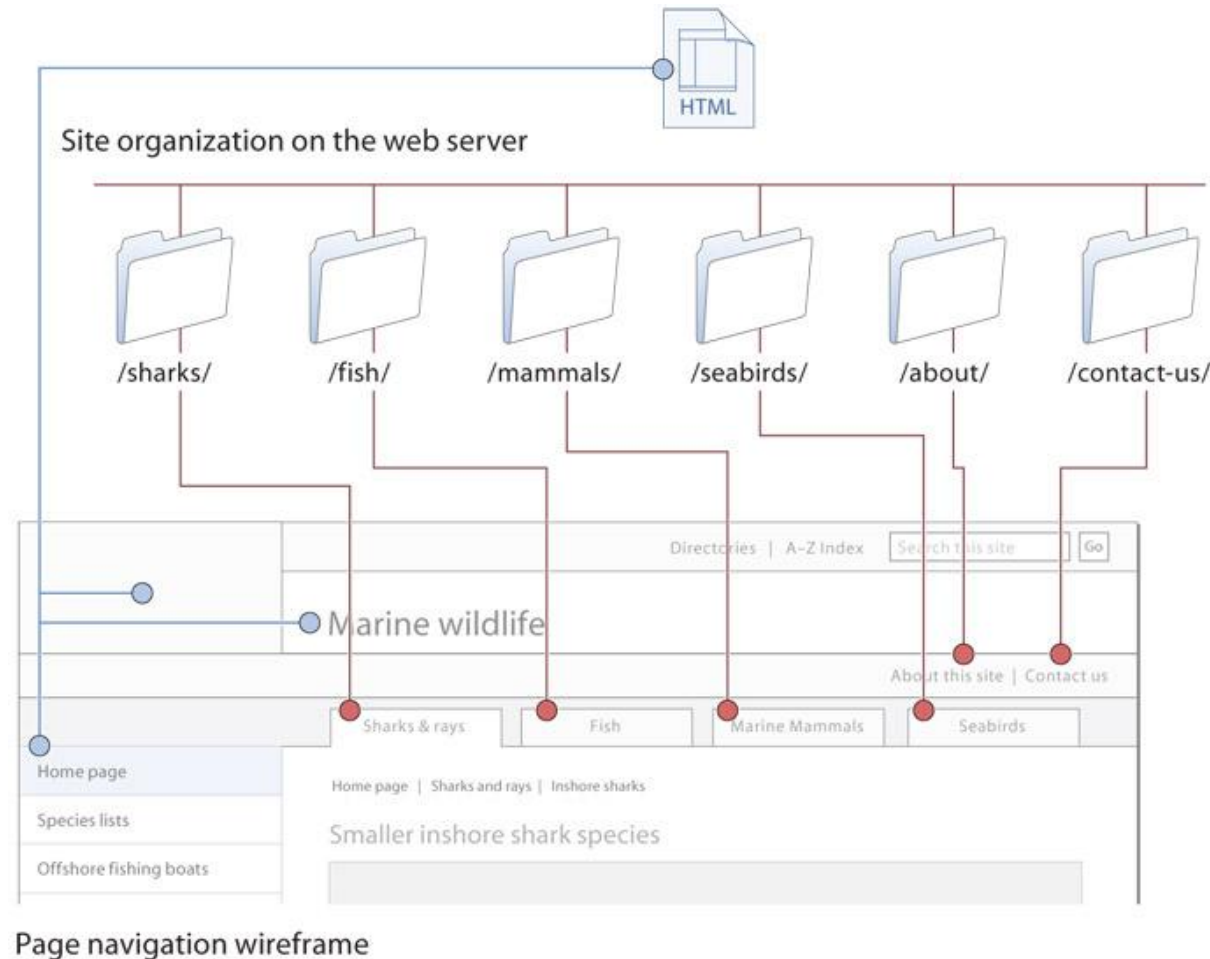
`www.whatever.edu/depts1/progs2/org004/bio_424.html`

In contrast, anyone (and any search engine) can parse this plain-language content arrangement at a glance:

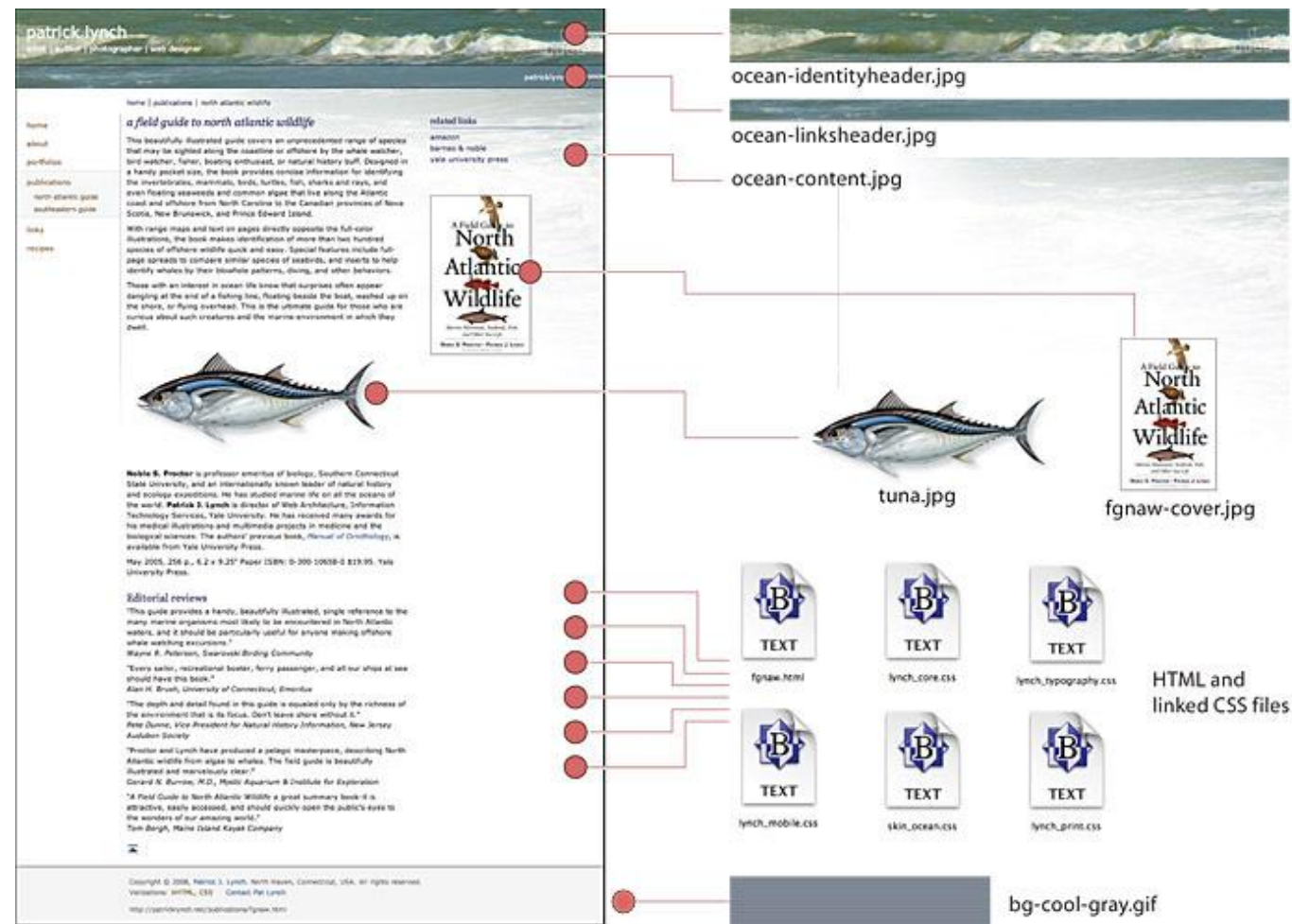
`www.whatever.edu/departments/biology/ornithology/field-ornithology-bio-224.html`

Naming Pages, Directories, and Supporting Files: Contoh Directories

Contoh directories

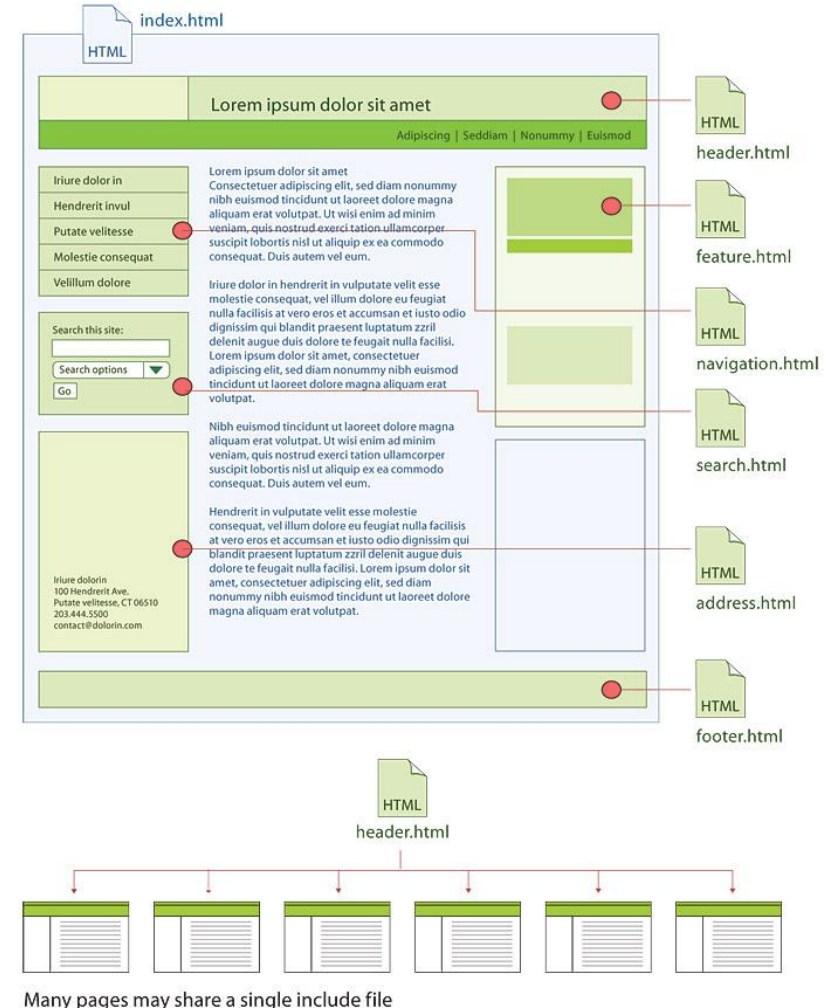


Contoh directories



Naming Pages, Directories, and Supporting Files: Include Files

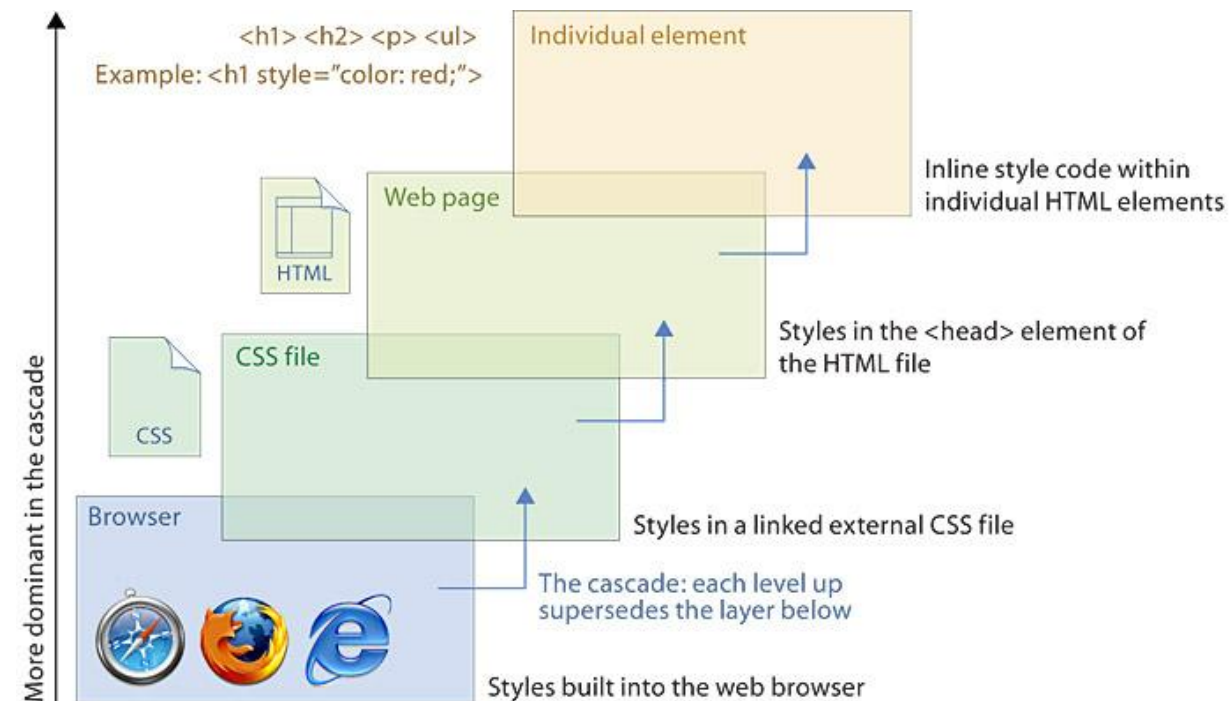
Dalam sebuah website berisi berberap folder dan file yang terdiri dari HTML, CSS, Javascript, photos, audio, dll, yang dimana ketika sebuah website diakses akan tertuju pada satu file HTML yaitu “index.html”. File index.html dapat berkaitan dengan directories dan file-file yang lain.



Naming Pages, Directories, and Supporting Files: CSS Hierarchy

CSS code secara umum dapat di embedded dalam sebuah tags HTML dan menjadi satu dalam file index.html.

Berikut ini merupakan struktur/hirarki dari CSS.



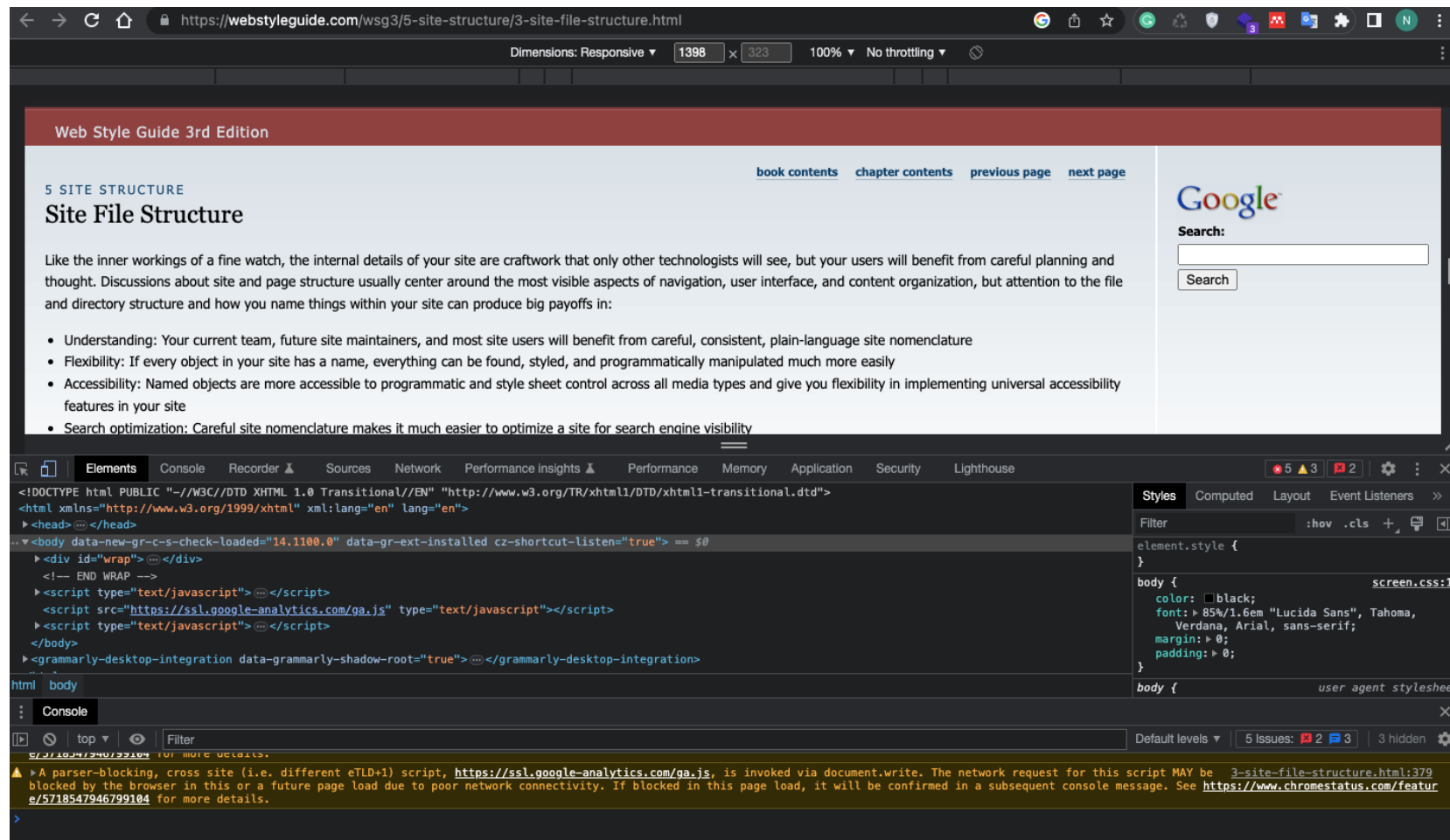
Naming Pages, Directories, and Supporting Files: HTML Content Containers

Dalam sebuah website memiliki sebuah strutur/wireframe, yang dimana untuk masing-masing struktur pada sebuah halaman website dapat diserbut sebagai **containers**.



Naming Pages, Directories, and Supporting Files: Mengetahui Directories Website

Cara mengetahui directories dari sebuah website



Localhost & Web Server (Hosting/Cpanel)

Localhost:

Sebuah program yang dapat berjalan pada PC/laptop individu dan dapat menjalankan tanpa terhubung internet.

Cara kerja localhost:

Dari sistem akan request berdasarkan IP protocols (misal: 127.1.1.1.) dan request tersebut akan diterima oleh IP protocols yang sama pada web server.

Web server (hosting/cpanel):

Sebuah fasilitas untuk mengakases folder dan file-file pada sebuah website, dan disediakan oleh penyedia web hosting service provider. Contoh penyedia hosting seperti dewaweb.com, rumahhosting.com, dll.

MVC Structure

Model-View-Controller (MVC): merupakan sebuah arsitektur dari sebuah web dan memiliki 3 komponen, yang dimana dari setiap komponen memiliki fungsi yang berbeda-beda.

Model:

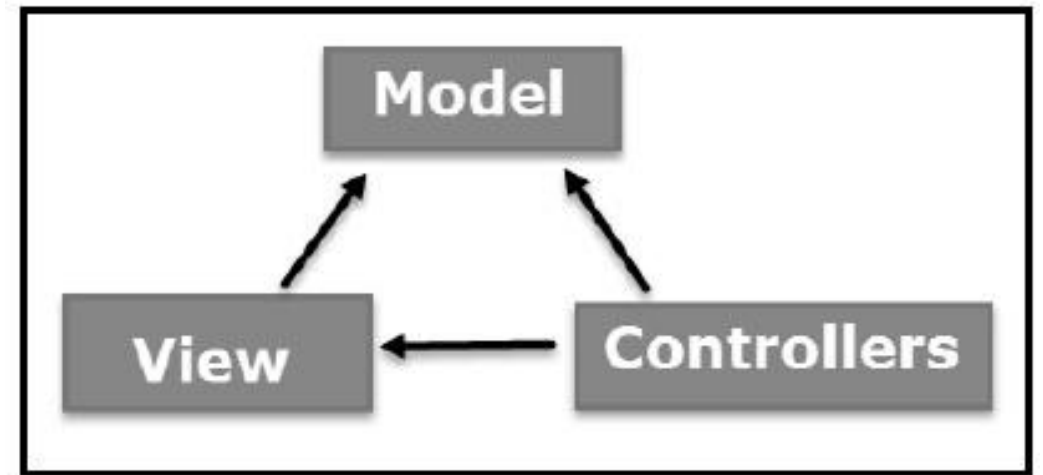
Sebuah komponen yang dapat mengerjakan sebuah proses-proses database.

View:

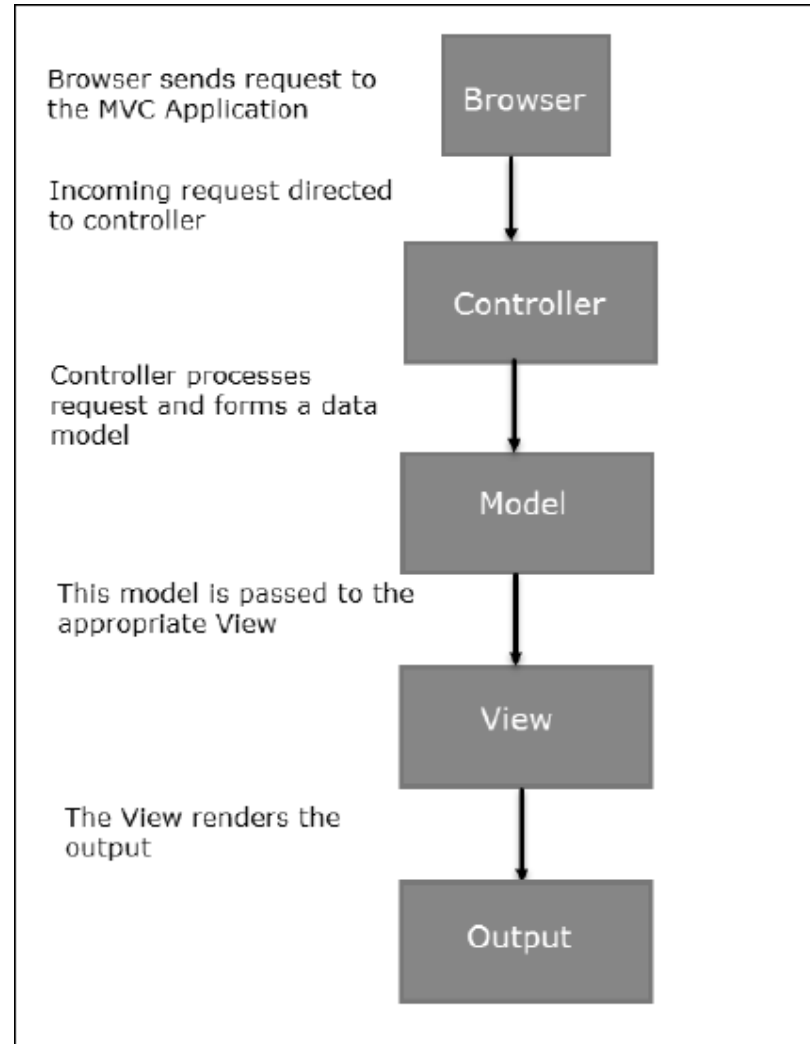
Sebuah komponen yang dapat mengerjakan proses-proses tampilan/interface.

Controller:

Sebuah komponen yang berkaitan dengan proses-proses business logic dan incoming request, dan dapat terhubung dengan komponen model dan view.



MVC Structure: Cara Kerja



Referensi

1. Lynch. (2011). Site File Structure Web Style Guide 3. <https://webstyleguide.com/wsg3/5-site-structure/3-site-file-structure.html>
2. Patrick J. Lynch, S. H. (2011). Web Style Guide, Third edition. In Web Style Guide. Sarah Horton, Patrick Lynch. <http://webstyleguide.com/wsg3/4-interface-design/>
3. Sinha, S. (2017). File Structure. In Beginning Laravel (pp. 15–20). https://doi.org/10.1007/978-1-4842-2538-7_3
4. <https://www.temok.com/blog/what-is-localhost/>
5. *Web hosting service* - *Wikipedia*. (n.d.). Retrieved March 15, 2023, from https://en.wikipedia.org/wiki/Web_hosting_service
6. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

TERIMA KASIH



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web Pertemuan 3

Zuhri Halim, S.Kom., M.Kom

Outline

1. Pendahuluan
2. Refleksi
3. HTML Multimedia
4. HTML Graphics



Pendahuluan: Capaian Hasil Belajar

Mahasiswa mampu menjelaskan penggunaan tag HTML multimedia (picture, audio, video, youtube).

Mahasiswa mampu menganalisis script yang menggunakan HTML multimedia dan Graphics.

Refleksi

لَن نَّزِيلَهُ إِلَّا بِالْحَقِّ وَأَن نَّبْشِطَ إِلَيْكَ آيَاتِنَا لَا يَصِلُنَا إِلَّا جَهَنَّمُ فَاخْشَوْا
فَإِن يَرَوْا كِسْفًا مِّنَ النُّجُومِ فَاصْبِرُوا وَلَا يُغْنِي عَنْهُمْ كِبَارُ الْعُمْرِ وَلَا
وَدَارُ الْآلِ وَالْأَقْبَابِ

Dan Kami turunkan dari Al Qur'an suatu yang menjadi penawar dan rahmat bagi orang-orang yang beriman dan Al Qur'an itu tidaklah menambah kepada orang-orang yang zalim selain kerugian (QS. Al Isra: 82)

HTML Multimedia



HTML Multimedia: What is Multimedia on Web Page?

Multimedia terdiri dari beberapa format seperti gambar, music, suara, videos, records, films, animations, dan lainnya. Dalam sebuah halaman web umumnya mengandung sebuah multimedia elements dengan tipe dan format yang berbeda-beda.



HTML Multimedia: Video

Element dari video yaitu **<video>**.

- Controls attribute dapat berfungsi untuk play, pause, dan volume pada video.
- Width dan height dapat berfungsi untuk mengatur lebar dan tinggi.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Video dapat di atur secara **autoplay**.

```
<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

```
<video width="320" height="240" autoplay muted>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Video Form

Video Formats

There are three supported video formats: MP4, WebM, and Ogg. The browser support for the different formats is:

Browser	MP4	WebM	Ogg
Edge	YES	YES	YES
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

Video Media

Tag	Description
<video>	Defines a video or movie
<u><source></u>	Defines multiple media resources for media elements, such as <video> and <audio>
<track>	Defines text tracks in media players

HTML Multimedia: Audio & Media Types

Contoh script audio

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

Media Types

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

HTML Multimedia: Plug-in

Plug-ins adalah semua program eksternal dari sebuah browser.

Plug-ins were designed to be used for many different purposes:

- To run Java applets
- To run Microsoft ActiveX controls
- To display Flash movies
- To display maps
- To scan for viruses
- To verify a bank id

Warning !

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browsers.

The support for Shockwave Flash has also been turned off in modern browsers.

Plug-in Obj

Element <object> dapat didukung oleh semua browser. Element tersebut dapat di embedded ke dalam sebuah HTML document.

```
<!DOCTYPE html>
<html>
<body>
  <object width="100px" height="50px" data="snippet.html"><
t
</body>
</html>
</html>
```

Alfreds Futterkiste	Berlin	Germany
Berglunds snabbkop	Lulea	Sweden
Centro comercial Moctezuma	Mexico D.F.	Mexico
Ernst Handel	Graz	Austria
FISSA Fabrica Inter. Salchichas SA.	Madrid	Spain
Galerfa del gastrone	Barcelona	Spain
Island Trading	Cowes	UK
Koniglich Essen	Brandenburg	Germany
Laughing Bacchus Wine Cellars	Vancouver	Canada
Magazzini Alimentari Riuniti	Bergamo	Italy
North/South	London	UK
Paris specialites	Paris	France
Rattlesnake Canyon Grocery	Albuquerque	USA
Simons bistro	Kebenhavn	Denmark
The Big Cheese	Portland	USA
Vaffeljemet	Arhus	Denmark
Wolski Zajazd	Warszawa	Poland

in Object E

Element <object> untuk menampilkan sebuah gambar.

```
<!DOCTYPE html>
<html
<body>

<object data="audi. j
peg"></obj ect>

</body>
</html>
```



Plug-in Embed

Element `<embed>` untuk menampilkan sebuah gambar.

```
<!DOCTYPE html>
<html>
<body>

<embed src="audi.
jpeg">

</body>
</html>
```



HTML Multimedia: YouTube Videos

Dalam sebuah halaman web dapat berisi sebuah video yang bersumber dari YouTube. Namun, terdapat kesulitan jika converting videos karena terdapat perbedaan formats dan bisa time-consuming.

Youtube Video Id:

YouTube akan menampilkan sebuah id (seperti tgbNymZ7vqY), programmer dapat menggunakan video tersebut untuk di embed ke dalam HTML document.

Playing Youtube Video in HTML:

- Upload the video to YouTube
- Take a note of the video id
- Define an `<iframe>` element in your web page
- Let the `src` attribute point to the video URL
- Use the `width` and `height` attributes to specify the dimension of the player
- Add any other parameters to the URL (see below)

toh Script Y

```
<!DOCTYPE
html>
<html>
<body>
<if rame          height=113
width=415 src="https://www.youtube.com/embed/C
6ziEu5wgv4">
e>

</body>
</html>
```



ntoh Script Y

```
<!DOCTYPE html
PE      >
<html>
<body>
<iframe      height=113
width=600; //www.youtube.com/embed/tgbNymZ7vqY?auto
utoplay=1&mute=1">
</iframe>

</body>
</html>
```



HTML Graphics

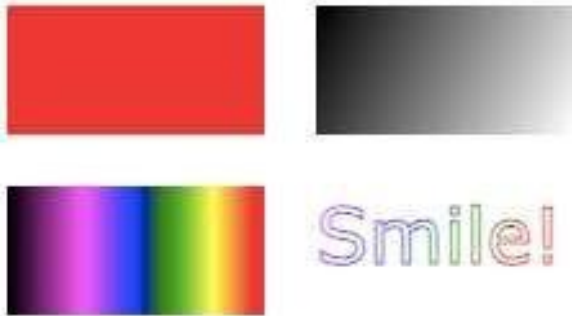


HTML Graphics: Canvas

What is HTML Canvas?

Element HTML `<canvas>` dapat digunakan untuk menggambar grafis dalam sebuah web page dengan menggunakan script Javascript. Element `<canvas>` dalam konsep graphics hanya sebuah container. Jika ingin menggambar dalam sebuah web page, maka harus menggunakan bahasa Javascript.

Dalam canvas terdapat beberapa metode, antara lain seperti **drawing paths, boxes, circles, text, dan adding images.**



The HTML `<canvas>` element is used to draw graphics on a web page.

The graphic to the left is created with `<canvas>`. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

Canvas

The markup looks like this:

```
<canvas id="rnyCanvas" width="200" height="100"></ canvas>
```

Note: Always specify an `id` attribute (to be referred to in a script), and a `width` and `height` attribute to define the size of the canvas. To add a border, use the `style` attribute.

Here is an example of a basic, empty canvas:



a JavaScri

Setelah membuat arean canvas, maka programmer dapat menambahkan Javascript untuk menggambar.

```
<!DOCTYPE html
E      >
<html>
<body>
<canvas      width="20"      height="10"      style="border:1      soli
#d3d3d3"      0"      0"      px      d
; Bani 6pay3ep ae ninrpsorye HTML Ter
</canvas.
>
<script
> var c documnent.getElernentByid("myC
= var anvas");
ctx.moveTo(0,0);
ctx.lineTo(200,1
00); stroke
( ) ;
</script>
</body>
</html>
```



Draw a Circle

Script dibawah ini untuk menggambar sebuah lingkaran dalam canvas.

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="100" style="border: 1px solid black;">
</canvas>

<script>
var c =
document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(50, 50, 40, 0, 2*Math.PI);
ctx.stroke();
</script>
</body>
</html>
</script>
```



Canvas a Text (H

Script dibawah ini untuk menggambar sebuah text **Hello World**.

```
<!DOCTYPE html
E
>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border: 1px solid black;">
; Bald opaylep ne nurrpiorye HTML -rer
canvas.
</canvas>
<script>
> var document.getElementById("myCa
c: var nvas");
ctx = c.getContext("2d");
ctx.fillText("Hello World",10,50);
</script>
</body>
>
</html>
>
```

Hel World

HTML Graphics: Draw Linear Gradient

Script dibawah ini untuk menggambar sebuah warna gradient.

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
var grd = ctx.createLinearGradient(0, 0, 200, 0);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10, 10, 150, 80);
</script>
```

HTML Graphics: Draw a Image

```
<!DOCTYPE html>
<html>
<body>

<p>Draw a image:</p>


<p>Draw a image:</p>
<canvas id="myCanvas" width="250" height="300"
style="border:1px solid #d3d3d3;">
  Draw a image.
</canvas>

<p><button onclick="myCanvas()">Tombol Ubah!</button></p>

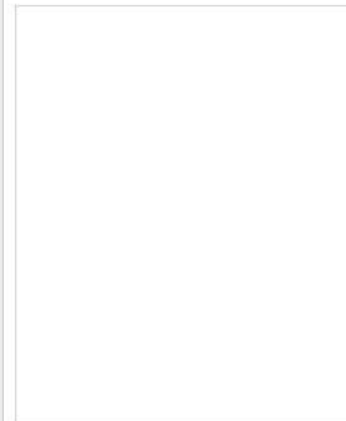
<script>
function myCanvas() {
  var c = document.getElementById("myCanvas");
  var ctx = c.getContext("2d");
  var img = document.getElementById("scream");
  ctx.drawImage(img,10,10);
}
</script>

</body>
</html>
```

Draw a image:



Draw a image:



Tombol Ubah!

HTML Graphics: Scalable Vector Graphics (SVG)

What is SVG?

- SVG stands for Scalable Vector Graphics
- SVG is used to define graphics for the Web
- SVG is a W3C recommendation

`<svg>` Element dapat digunakan untuk drawing paths, boxes, circles, text, dan graphic images.

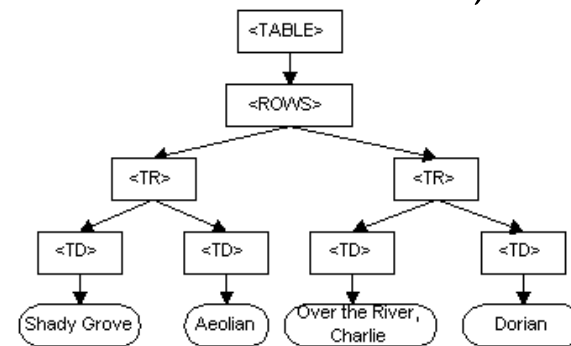
h Script Scal

```
<!DOCTYPE html>
<html>
<body>
<svg width=100 height=100>
  <circle cx=50 cy=50 r=50 stroke=green stroke-width=4 fill=yellow />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
1>
```



HTML Graphics: Perbedaan SVG & Canvas

- SVG is a language for describing 2D graphics in XML.
- Canvas draws 2D graphics, on the fly (with a JavaScript).
- SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.
- In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.
- Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.



DOM representation of the example table

Canvas

- Resolution dependent
- No support for event handlers
- Poor text rendering
 - Best suited for applications capabilities with large rendering areas
- You can save the resulting image as .png or .jpg
- Well suited for graphic-intensive games

SVG

- Resolution independent
- Support for event handlers
- (Google Maps)
- Slow rendering if complex (anything that uses the DOM a lot will be slow)
- Not suited for game applications

Referensi

1. Lynch. (2011). Site File Structure Web Style Guide 3. <https://webstyleguide.com/wsg3/5-site-structure/3-site-file-structure.html>
2. Patrick J. Lynch, S. H. (2011). Web Style Guide, Third edition. In Web Style Guide. Sarah Horton, Patrick Lynch. <http://webstyleguide.com/wsg3/4-interface-design/>
3. https://www.w3schools.com/html/html5_svg.asp
4. https://www.w3schools.com/html/html_media.asp
5. https://www.w3schools.com/html/html5_canvas.asp
6. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm



TERIMA KASIH



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web Pertemuan 4

Zuhri Halim, S.Kom., M.Kom.

Outline

1. Pendahuluan
2. Refleksi
3. HTML Forms
4. HTML Form Attributes
5. HTML Form Elements
6. HTML Input Types
7. HTML Input Attributes
8. HTML Input Form Attributes
9. Use of Forms in Bootstrap
10. Use of Forms in PHP Framework



Pendahuluan: Capaian Hasil Belajar

Mahasiswa mampu menganalisis kerangka sebuah web yang menggunakan HTML form input

Refleksi

وَأَطِيعُوا اللَّهَ وَرَسُولَهُ وَلَا تَنَازَعُوا فَتَفْشَلُوا وَتَذْهَبَ
رِيحُكُمْ وَاصْبِرُوا ^{قُلْ} إِنَّ اللَّهَ مَعَ الصَّابِرِينَ

Dan taatilah Allah dan Rasul-Nya dan **janganlah kamu berselisih**, yang menyebabkan kamu menjadi gentar dan kekuatanmu hilang dan bersabarlah. Sungguh, Allah beserta orang-orang sabar (QS. Al-Anfal: 46).

HTML Form



HTML Form

HTML form dapat digunakan untuk menginput data-data. Form input paling sering juga digunakan untuk mengirim data ke sebuah server untuk di proses.

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php">
  <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>

<p>If you click the "Submit" button, the form-data will be sent to a
page called "/action_page.php".</p>

</body>
</html>
```

HTML Forms

First name:

Last name:

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".

HTML Form: Element

Element form merupakan sebuah wadah untuk di isikan beberapa tipe element input seperti text fields, checkboxes, radio buttons, submit buttons, dll.

```
<form>
```

```
▪
```

```
form elements
```

```
▪
```

```
</form>
```


HTML Form: Input Element

HTML input element digunakan untuk membuat form input, yang dimana terdiri dari sebagai berikut ini.

Type	Description
<code><input type="text"></code>	Displays a single-line text input field
<code><input type="radio"></code>	Displays a radio button (for selecting one of many choices)
<code><input type="checkbox"></code>	Displays a checkbox (for selecting zero or more of many choices)
<code><input type="submit"></code>	Displays a submit button (for submitting the form)
<code><input type="button"></code>	Displays a clickable button

HTML Form: Text Fields & Labels

- Label banyak digunakan untuk screen-reader users.
- Membantu memberikan informasi-informasi.
- Contoh membuat text fields.

```
<!DOCTYPE html>
<html>
<body>

<h2>Text input fields</h2>

<form>
  <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">
</form>

<p>Note that the form itself is not visible.</p>

<p>Also note that the default width of text input fields is 20
characters.</p>

</body>
</html>
```

Text input fields

First name:

Last name:

Note that the form itself is not visible.

Also note that the default width of text input fields is 20 characters.

HTML Form: Radio Buttons

Radio buttons merupakan pilihan yang secara umum user harus memilih salah satu dari pilihan yang tersedia.

```
<!DOCTYPE html>
<html>
<body>

<h2>Radio Buttons</h2>

<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>

</body>
</html>
```

Radio Buttons

Choose your favorite Web language:

- ☐ HTML
- ☐ CSS
- ☐ JavaScript

HTML Form: Checkboxes

Checkboxes merupakan pilihan yang secara umum user harus memilih salah satu atau lebih dari pilihan yang tersedia.

```
<!DOCTYPE html>
<html>
<body>

<h2>Checkboxes</h2>
<p>The <strong>input type="checkbox"</strong> defines a checkbox:</p>

<form action="/action_page.php">
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

Checkboxes

The **input type="checkbox"** defines a checkbox:

- ☒ I have a bike
- ☒ I have a car
- ☐ I have a boat

Submit

HTML Form: Submit Button

Submit button merupakan tombol yang dapat dikatakan sebagai eksekutor untuk menangani proses-proses.

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

Button di HTML form



```
<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 di Bootstrap

HTML Form Attributes



HTML Form Attributes: The Action Attribute

Action attribute merupakan sebuah action yang dapat diproses lebih lanjut ketika form input sudah di submitted.

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php">
  <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>

<p>If you click the "Submit" button, the form-data will be sent to a
page called "/action_page.php".</p>

</body>
</html>
```

Submitted Form Data

Your input was received as:

fname=John&lname=Doe

The server has processed your input and returned this answer.

Note: This tutorial will not teach you how servers are processing input. Processing input is explained in our [PHP tutorial](#).

HTML Form Attributes: The Action Attribute

Action attribute merupakan sebuah action yang dapat diproses lebih lanjut ketika form input sudah di submitted.

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php">
  <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>

<p>If you click the "Submit" button, the form-data will be sent to a
page called "/action_page.php".</p>

</body>
</html>
```

Submitted Form Data

Your input was received as:

fname=John&lname=Doe

The server has processed your input and returned this answer.

Note: This tutorial will not teach you how servers are processing input. Processing input is explained in our [PHP tutorial](#).

HTML Form Attributes: The Target Attribute

Target attribute merupakan tampilan ketika respon yang diterima ketika form input sudah di submitted.

Value	Description
<code>_blank</code>	The response is displayed in a new window or tab
<code>_self</code>	The response is displayed in the current window
<code>_parent</code>	The response is displayed in the parent frame
<code>_top</code>	The response is displayed in the full body of the window
<code>framename</code>	The response is displayed in a named iframe

The default value is `_self` which means that the response will open in the current window.

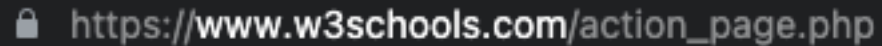
```
<form action="/action_page.php" target="_blank">
```

HTML Form Attributes: The Method Attribute

Target method attribute merupakan metode HTTP yang digunakan ketika form input sudah di submitted. Form data bisa dikirim sebagai URL variables (with method="get") atau sebagai HTTP post transaction (method="post").



https://www.w3schools.com/action_page.php?fname=John&lname=Doe



https://www.w3schools.com/action_page.php

Berdasarkan URL di atas mana yang menerapkan metode HTTP get dan post?

Catatan penting untuk HTTP get dan post.

Notes on GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- Useful for form submissions where a user wants to bookmark the result
- GET is good for non-secure data, like query strings in Google

Notes on POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

HTML Form Attributes: Lists

Attribute	Description
<u>accept-charset</u>	Specifies the character encodings used for form submission
<u>action</u>	Specifies where to send the form-data when a form is submitted
<u>autocomplete</u>	Specifies whether a form should have autocomplete on or off
<u>enctype</u>	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
<u>method</u>	Specifies the HTTP method to use when sending form-data
<u>name</u>	Specifies the name of the form
<u>novalidate</u>	Specifies that the form should not be validated when submitted
<u>rel</u>	Specifies the relationship between a linked resource and the current document
<u>target</u>	Specifies where to display the response that is received after submitting the form

HTML Form Elements



HTML Form Elements

Jenis-jenis form element di HTML, antara lain sebagai berikut.

The HTML `<form>` element can contain one or more of the following form elements:

- `<input>`
- `<label>`
- `<select>`
- `<textarea>`
- `<button>`
- `<fieldset>`
- `<legend>`
- `<datalist>`
- `<output>`
- `<option>`
- `<optgroup>`

HTML Input Types



HTML Input Types

Jenis-jenis input types element di HTML, antara lain sebagai berikut.

Here are the different input types you can use in HTML:

- `<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">`

HTML Input Attributes



HTML Input Types

Jenis-jenis input attributes HTML, antara lain sebagai berikut:

1. The value attribute
2. The readonly attribute
3. The disabled attribute
4. The size attribute
5. The maxlength attribute
6. The min and max attribute
7. The multiple attribute
8. The pattern attribute
9. The placeholder attribute
10. The required attribute
11. The step attribute
12. The autofocus attribute
13. The height and width attribute
14. The list attribute
15. The autocomplete attribute

HTML Input Form Attributes



HTML Input Types

Jenis-jenis input form element dalam HTML, antara lain sebagai berikut:

1. The form attribute
2. The formaction attribute
3. The formenctype attribute
4. The formmethod attribute
5. The formtarget attribute
6. The formnovalidate attribute
7. The novalidate attribute

Use of Forms in Bootstrap



Use of Forms in Bootstrap

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">
    <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" placeholder="Password">
  </div>
  <div class="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>
```

Copy

Use of Forms in PHP Framework



Referensi

1. Lynch. (2011). Site File Structure Web Style Guide 3. <https://webstyleguide.com/wsg3/5-site-structure/3-site-file-structure.html>
2. Patrick J. Lynch, S. H. (2011). Web Style Guide, Third edition. In Web Style Guide. Sarah Horton, Patrick Lynch. <http://webstyleguide.com/wsg3/4-interface-design/>
3. <https://www.w3schools.com/>
4. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

TERIMA KASIH



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web Pertemuan 5

Zuhri Halim, S.Kom., M.Kom.

Outline

1. Introduction
2. Reflection
3. CSS Introduction
4. How to Add CSS
5. Implementation CSS



Introduction: Learning Outcomes

Mahasiswa menganalisis layout dan estetika sebuah web yang menggunakan CSS.

Reflection

وَاطِيعُوا اللَّهَ وَرَسُولَهُ وَلَا تَنَازَعُوا فَتَفْشَلُوا وَتَذْهَبَ
رِيحُكُمْ وَاصْبِرُوا ^{قُلْ} إِنَّ اللَّهَ مَعَ الصَّابِرِينَ

Dan taatilah Allah dan Rasul-Nya dan **janganlah kamu berselisih**, yang menyebabkan kamu menjadi gentar dan kekuatanmu hilang dan bersabarlah. Sungguh, Allah beserta orang-orang sabar (QS. Al-Anfal: 46).

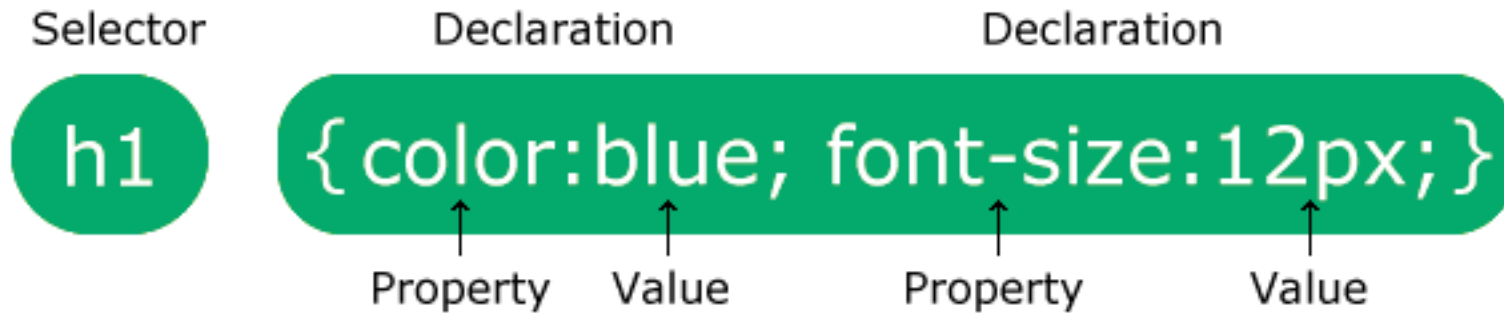
CSS Introduction



CSS Introduction: Definition

- Cascading Style Sheet (CSS) sering digunakan dalam kode HTML untuk menciptakan suatu kumpulan style yang terkadang dapat digunakan untuk memperluas kemampuan HTML
- Manfaat dari CSS antara lain:
 - Memperindah tampilan halaman website.
 - Mempermudah mengubah secara keseluruhan warna dan tampilan.
 - Memformat ulang situs Anda secara cepat.
 - Menghemat kode dan waktu dalam mendesain tampilan website.

CSS Introduction: Syntax



- Selector adalah sebuah element yang ingin dimodifikasi
- Declaration block mengandung satu atau lebih dari declarations dan terpisah dengan menggunakan semicolons.
- Setiap declaration termasuk sebuah CSS property seperti name dan sebuah value.

```
p {  
  color: red;  
  text-align: center;  
}
```

CSS Introduction : Type of Selector

CSS selector merupakan selects HTML element yang ingin dibuat style. CSS selectors terbagi menjadi beberapa kategori antara lain sebagai berikut:

- Simple selectors: menampilkan elements berdasarkan name, id, class.
- Combinator selectors: menampilkan elements berdasarkan specific relationship between them.
- Pseudo-class selectors: menampilkan elements berdasarkan certain state.
- Pseudo-elements selectors: menampilkan style dari sebuah bagian element.
- Attribute selectors: menampilkan elements berdasarkan sebuah attribute atau attribute value.

CSS Introduction : Element Selector

Element selector merupakan element yang berdasarkan dari element name. “P” adalah sebuah element selector yang berfungsi untuk center-aligned.

```
p {  
  text-align: center;  
  color: red;  
}
```

CSS Introduction: id Selector

Id selector menggunakan id attribute dalam sebuah HTML element untuk memilih sebuah specific element. id element merupakan sebuah keunikan dalam sebuah page. Biasanya ditulis dengan menggunakan karakter “#”. Namun id name tidak dapat tulis dengan awalan sebuah nomor.

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

CSS Introduction: Class Selector

Class selector merupakan sebuah specific class yang secara umum di tulis dengan awalan karakter “.” dan di ikuti dengan class name.

```
.center {  
  text-align: center;  
  color: red;  
}
```

In this example only <p> elements with class="center" will be red and center-aligned:

```
p.center {  
  text-align: center;  
  color: red;  
}
```


CSS Introduction: Universal Selector

Universal selector digunakan untuk semua HTML element dalam sebuah page.

The CSS rule below will affect every HTML element on the page:

```
* {  
  text-align: center;  
  color: blue;  
}
```

CSS Introduction: Grouping Selector

Grouping selector merupakan sebuah HTML element dengan jenis style yang sama. Contoh penggunaan h1, h2, dan p element yang mempunyai style sama.

```
h1 {  
    text-align: center;  
    color: red;  
}  
  
h2 {  
    text-align: center;  
    color: red;  
}  
  
p {  
    text-align: center;  
    color: red;  
}
```

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

How to Add CSS



How to Add CSS

Dalam CSS terdapat 3 cara untuk menambahkan syle sheet antara lain:

- External CSS
- Internal CSS
- Inline CSS

How to Add CSS: External CSS

Teknik penulisan syntax CSS dengan external style adalah teknik penulisan yang memisahkan file CSS dan HTML. Sebuah external style sheet dapat ditulis dalam text editor, dan harus disimpan dengan sebuah **.css** extension.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



How to Add CSS: Example External CSS

"mystyle.css"

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

Note: Do not add a space between the property value and the unit:

Incorrect (space): `margin-left: 20 px;`

Correct (nospace): `margin-left: 20px;`

How to Add CSS: Internal CSS

Internal style sheet dapat digunakan dan ditulis dalam single HTML page. Secara best practice dapat dibuka dengan tag `<style>` element, dan dimasukkan ke dalam head section.

Internal styles are defined within the `<style>` element, inside the `<head>` section of an HTML page:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body {
    background-color: linen;
  }

  h1 {
    color: maroon;
    margin-left: 40px;
  }
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```


How to Add CSS: Inline CSS

Inline style dapat diplikasi dalam sebuah single element. Dalam menggunakan inline styles, jenis style attribute harus relevant dengan element. Style attribute dapat mengandung sebuah CSS property.

Inline styles are defined within the "style" attribute of the relevant element:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

How to Add CSS: Multiple Style Sheets

Multiple style sheet dapat memiliki nama selector element yang sama, tetapi nilai/value dari selector tersebut dapat berbeda.

Assume that an **external style sheet** has the following style for the <h1> element:

```
h1 {  
  color: navy;  
}
```

Then, assume that an **internal style sheet** also has the following style for the <h1> element:

```
h1 {  
  color: orange;  
}
```

How to Add CSS: Multiple Style Sheets (Continue)

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

Contoh script di atas merupakan penggunaan CSS setelah men-declare external style sheet untuk **<h1>** element.

```
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Contoh script di atas merupakan penggunaan CSS sebelum men-declare external style sheet untuk **<h1>** element.

How to Add CSS: Cascading Order

Jika dalam sebuah HTML page terdapat satu atau lebih style, maka terdapat aturan untuk mengatur penempatan style code dalam source code, antara lain sebagai berikut:

- Inline style: style yang berada dalam HTML element.
- External dan internal style sheets: dalam head section.

Inline style merupakan highest priority, dan akan menimpa style dari external dan internal dalam sebuah source code.

Implementation CSS



Implementation CSS on Table

A Fancy Table

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK
Paris spécialités	Marie Bertrand	France

Implementation CSS on Table (Continue)

```
<!DOCTYPE html>
<html>
<head>
<style>
#customers {
  font-family: Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 100%;
}

#customers td, #customers th {
  border: 1px solid #ddd;
  padding: 8px;
}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #04AA6D;
  color: white;
}
</style>
</head>
<body>
```

```
<h1>A Fancy Table</h1>
<table id="customers">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Berglunds snabbköp</td>
    <td>Christina Berglund</td>
    <td>Sweden</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
  <tr>
    <td>Ernst Handel</td>
    <td>Roland Mendel</td>
    <td>Austria</td>
  </tr>
  <tr>
    <td>Island Trading</td>
    <td>Helen Bennett</td>
    <td>UK</td>
  </tr>
  <tr>
    <td>Königlich Essen</td>
    <td>Philip Cramer</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Laughing Bacchus Winecellars</td>
    <td>Yoshi Tannamuri</td>
    <td>Canada</td>
  </tr>
  <tr>
    <td>Magazzini Alimentari Riuniti</td>
    <td>Giovanni Rovelli</td>
    <td>Italy</td>
  </tr>
  <tr>
    <td>North/South</td>
    <td>Simon Crowther</td>
    <td>UK</td>
  </tr>
  <tr>
    <td>Paris spécialités</td>
    <td>Marie Bertrand</td>
    <td>France</td>
  </tr>
</table>
</body>
</html>
```


Implementation CSS on Form

Buatlah form dengan menggunakan CSS style dan buatlah menjadi external style sheet (file CSS terpisah).

Using CSS to style an HTML Form

First Name

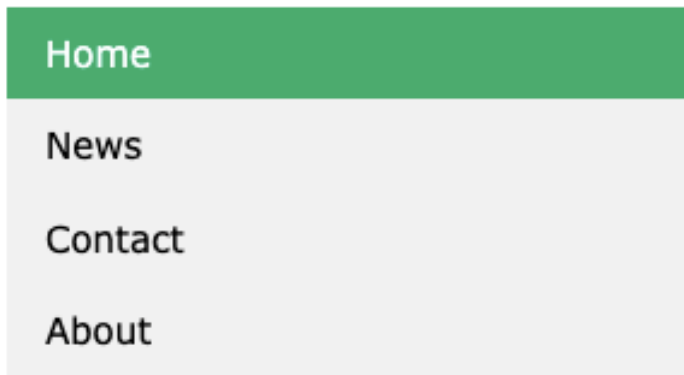
Last Name

Country

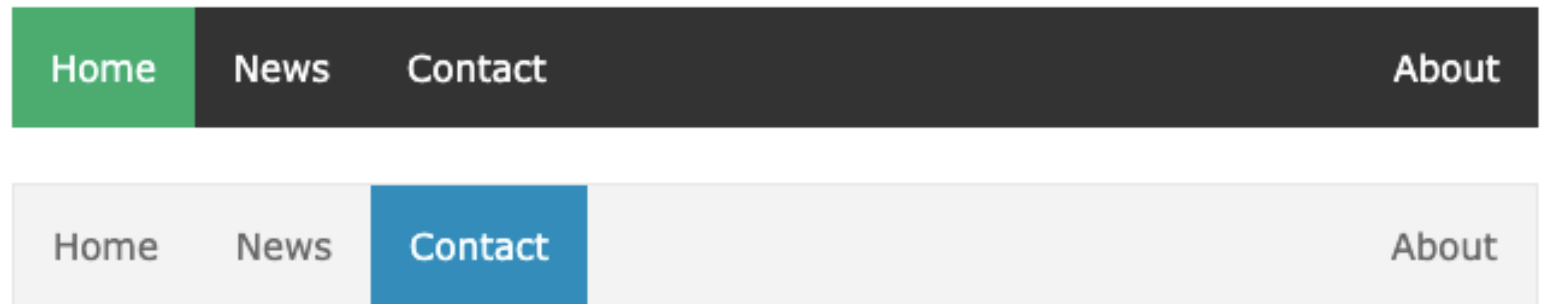


Implementation CSS on Navigation Bars

Vertical



Horizontal



Implementation CSS on Navigation Bars (Continue)

Home
News
Contact
About

Fixed Full-height Side Nav

Try to scroll this area, and see how the sidenav sticks to the page

Notice that this div element has a left margin of 25%. This is because the side navigation is set to 25% width. If you remove the margin, the sidenav will overlay/sit on top of this div.

Also notice that we have set overflow:auto to sidenav. This will add a scrollbar when the sidenav is too long (for example if it has over 50 links inside of it).

Some text..

Some text..

Some text..

Some text..

Some text..

Some text..

Some text..

Implementation CSS on Navigation Bars (Continue)

```
<html>
<head>
<style>
body {
  margin: 0;
}

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 25%;
  background-color: #f1f1f1;
  position: fixed;
  height: 100%;
  overflow: auto;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

li a.active {
  background-color: #04AA6D;
  color: white;
}

li a:hover:not(.active) {
  background-color: #555;
  color: white;
}
</style>
```

```
</head>
<body>

<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

<div style="margin-left:25%;padding:1px 16px;height:1000px;">
  <h2>Fixed Full-height Side Nav</h2>
  <h3>Try to scroll this area, and see how the sidenav sticks to the page</h3>
  <p>Notice that this div element has a left margin of 25%. This is because
the side navigation is set to 25% width. If you remove the margin, the sidenav
will overlay/sit on top of this div.</p>
  <p>Also notice that we have set overflow:auto to sidenav. This will add a
scrollbar when the sidenav is too long (for example if it has over 50 links
inside of it).</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
</div>

</body>
</html>
```

Implementation CSS on Form (Continue)

```
<html>
<style>
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type=submit]:hover {
  background-color: #45a049;
}

div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}
</style>
```

```
<body>

<h3>Using CSS to style an HTML Form</h3>

<div>
  <form action="/action_page.php">
    <label for="fname">First Name</label>
    <input type="text" id="fname" name="firstname" placeholder="Your
name..">

    <label for="lname">Last Name</label>
    <input type="text" id="lname" name="lastname" placeholder="Your last
name..">

    <label for="country">Country</label>
    <select id="country" name="country">
      <option value="australia">Australia</option>
      <option value="canada">Canada</option>
      <option value="usa">USA</option>
    </select>

    <input type="submit" value="Submit">
  </form>
</div>

</body>
</html>
```

Referensi

1. Lynch. (2011). Site File Structure Web Style Guide 3. <https://webstyleguide.com/wsg3/5-site-structure/3-site-file-structure.html>
2. Patrick J. Lynch, S. H. (2011). Web Style Guide, Third edition. In Web Style Guide. Sarah Horton, Patrick Lynch. <http://webstyleguide.com/wsg3/4-interface-design/>
3. <https://www.w3schools.com/css>

TERIMA KASIH



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web Pertemuan 6 Dasar-Dasar Javascript

Zuhri Halim, S.Kom., M.Kom.

Outline

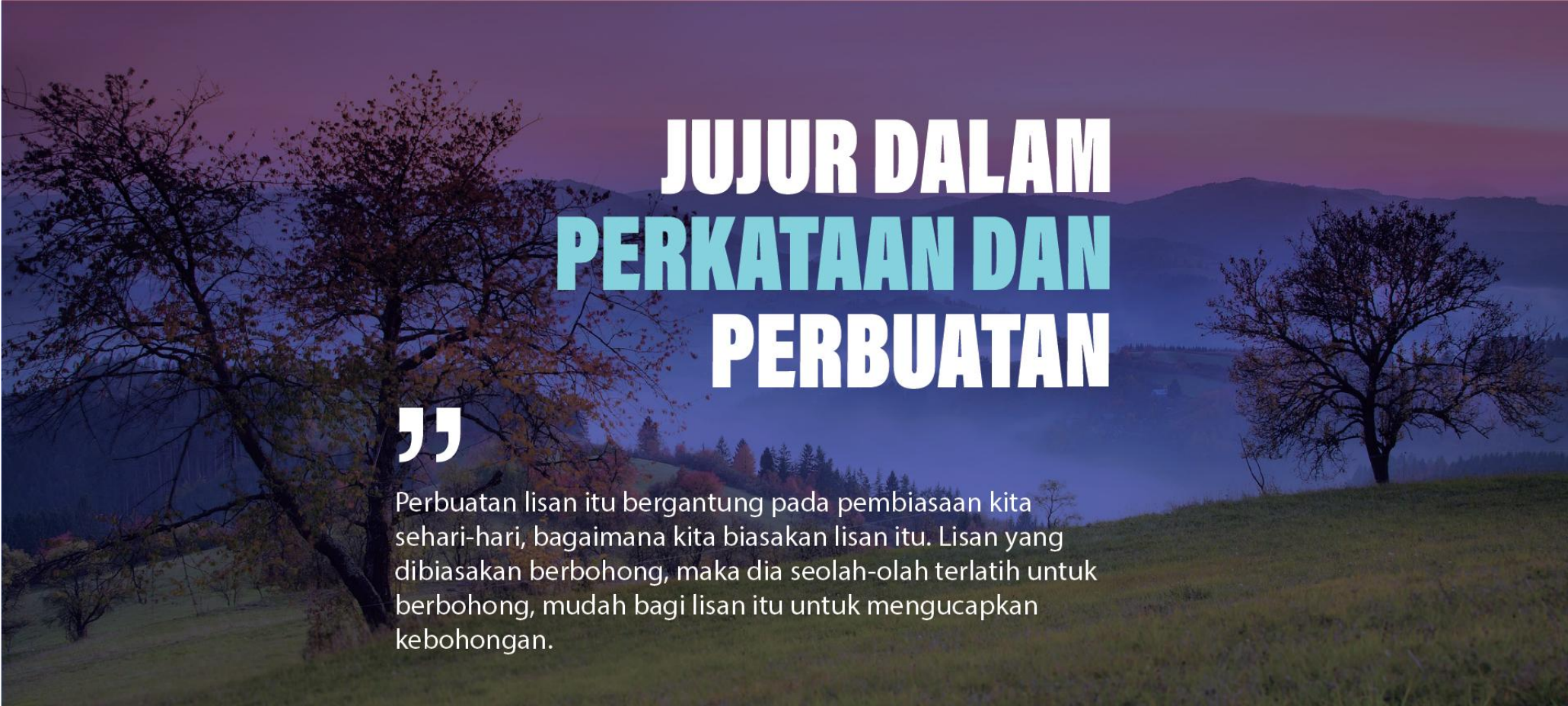
1. Introduction
2. Reflection
3. Javascript Introduction
4. Statement
5. Syntax
6. Variable
7. Function
8. Examples



Introduction: Learning Outcomes

Mahasiswa mampu menganalisis kedinamisan dan keinteraktifan sebuah web yang menggunakan Javascript dasar.

Reflection



JUJUR DALAM PERKATAAN DAN PERBUATAN

”

Perbuatan lisan itu bergantung pada pembiasaan kita sehari-hari, bagaimana kita biasakan lisan itu. Lisan yang dibiasakan berbohong, maka dia seolah-olah terlatih untuk berbohong, mudah bagi lisan itu untuk mengucapkan kebohongan.

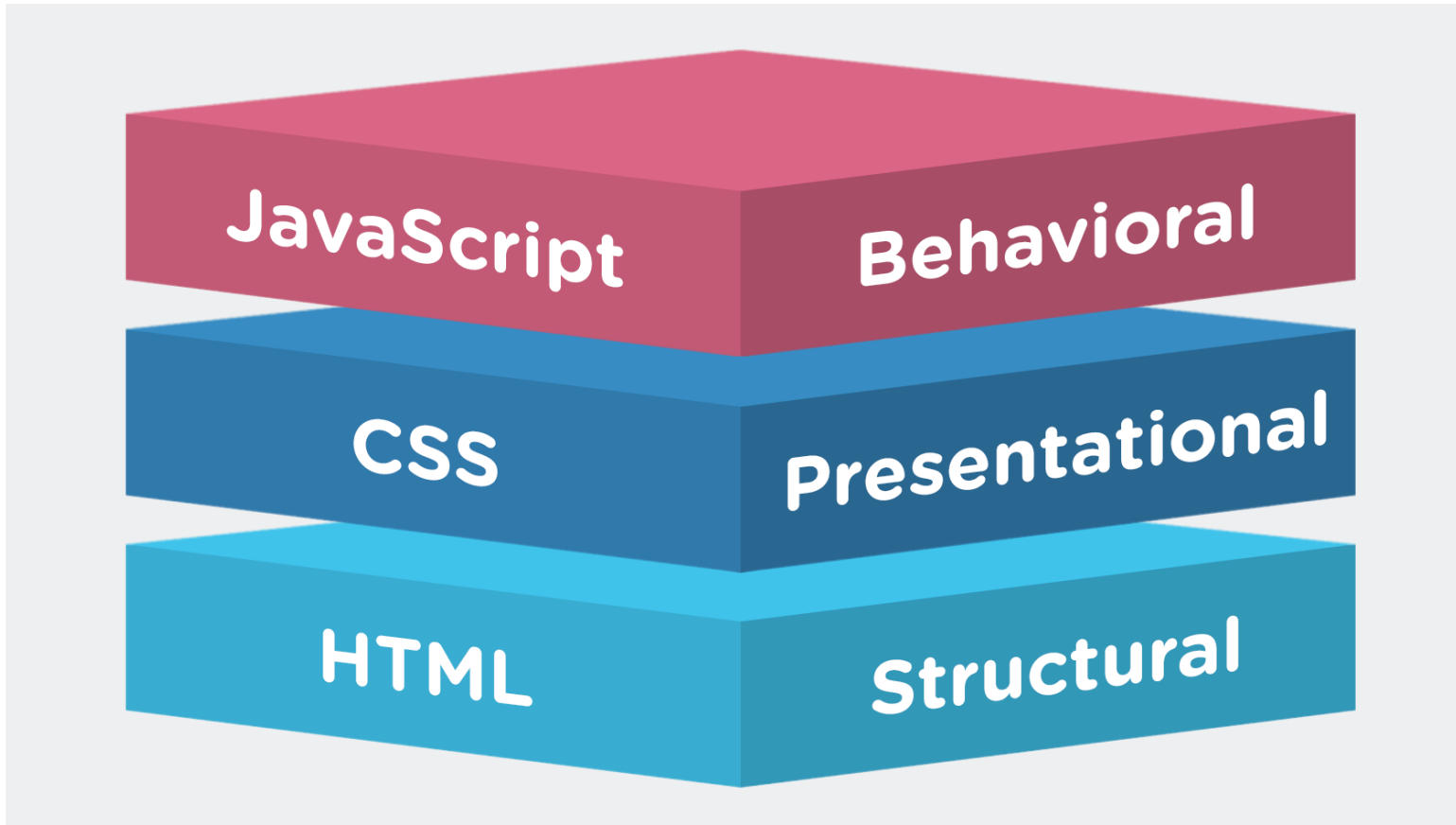
Javascript Introduction



Javascript Introduction

- Javascript merupakan bahasa pemrograman yang khusus dibuat untuk mengembangkan website.
- Javascript merupakan komponen ke-3 yang harus dikuasai oleh seorang web developer setelah HTML dan CSS.
- Perbedaan fungsi dari Javascript dengan 2 komponen lain adalah sebagai berikut:
 - HTML: berfungsi untuk mendefinisikan struktu/layout dari sebuah web.
 - CSS: berfungsi untuk memberikan style/mempercantik halaman website.
 - Javascript: berfungsi untuk mengatur behaviour dari sebuah halaman web.

Javascript Introduction (Continue)



Javascript Introduction: How to Use Javascript

- Sama seperti CSS, anda bisa menggunakan Javascript dengan 3 cara.
 - Di dalam elemen body, menggunakan tag script.
 - Di dalam elemen head, menggunakan tag script.
 - Menggunakan file terpisah dengan menggunakan ekstensi .js. Misal main.js. Nama file bisa bebas. Kemudian anda harus menambahkan baris berikut di elemen head dari dokumen HTML: `<script src="main.js"></script>`

Javascript Introduction: in Tag Body

Javascript di dalam tag body.

```
<html>
  <head>
    <title>Ini adalah judul website</title>
  </head>
  <body>
    <button onclick="test()">Munculkan Alert</button>
    <script>
      function test() {
        alert("Hello World");
      }
    </script>
  </body>
</html>
```

Javascript Introduction: in Tag Head

Javascript di dalam tag head.

```
1  <html lang="en">
2  <head>
3      <meta charset="UTF-8">
4      <meta name="viewport" content="width=device-width, initial-scale=1.0">
5      <meta http-equiv="X-UA-Compatible" content="ie=edge">
6      <title>Document</title>
7      <script>
8          function test() {
9              alert("Hello World");
10         }
11     </script>
12 </head>
13 <body>
14     <button onclick="test()">Munculkan Alert</button>
15 </body>
16 </html>
```

Javascript Introduction: External Files

Javascript pada external files.

```
1  <html lang="en">
2  <head>
3      <meta charset="UTF-8">
4      <meta name="viewport" content="width=device-width, initial-scale=1.0">
5      <meta http-equiv="X-UA-Compatible" content="ie=edge">
6      <title>Document</title>
7      <script src="test3.js"></script>
8  </head>
9  <body>
10     <button onclick="test()">Munculkan Alert</button>
11 </body>
12 </html>
```

Javascript Introduction: External Files (Continue)

Javascript pada external files.

```
1  function test() {  
2      alert("Hello World");  
3  }
```

Javascript Introduction: Why Use?

1. Javascript bisa mengubah konten/isi dari sebuah elemen HTML.
2. Javascript bisa mengubah nilai atribut dari sebuah elemen HTML.
3. Javascript bisa mengubah style CSS dari sebuah elemen HTML.
4. Javascript bisa menyembunyikan elemen HTML
5. Dan masih banyak fungsi lainnya.

Javascript Introduction: Change the Contents

```
<html>
  <head>
    <title>Ini adalah judul website</title>
  </head>
  <body>
    <h1 id="heading1">Hello World!</h1>
    <button onclick="test()">Ubah Teks</button>
    <script>
      function test() {
        var heading1 = document.getElementById("heading1");
        heading1.innerHTML = "Hello Indonesia";
      }
    </script>
  </body>
</html>
```

Javascript Introduction: Change the Attribute Values

```
1  <html lang="en">
2  <head>
3      <title>Test 4</title>
4      <script>
5          function changeGoogleCom() {
6              var a = document.getElementById('myGoogle');
7              a.href = 'http://google.com';
8          }
9
10         function changeGoogleCoId() {
11             var a = document.getElementById('myGoogle');
12             a.href = 'http://google.co.id';
13         }
14     </script>
15 </head>
16 <body>
17     <a id="myGoogle" href="http://google.com">Goto Google</a><br/><br/>
18     <button onclick="changeGoogleCom()">Change google.com</button>
19     <button onclick="changeGoogleCoId()">Change google.co.id</button>
20 </body>
21 </html>
```

Javascript Introduction: Change the CSS Style

```
1  <html lang="en">
2  <head>
3      <title>Test 5</title>
4      <script>
5          function changeReguler() {
6              var p = document.getElementById('myText');
7              p.style.fontWeight = 'normal';
8          }
9
10         function changeBold() {
11             var p = document.getElementById('myText');
12             p.style.fontWeight = 'bold';
13         }
14     </script>
15 </head>
16 <body>
17     <p id="myText" style="font-size:20px;">Ini adalah test</p>
18     <button onclick="changeReguler()">Change Reguler</button>
19     <button onclick="changeBold()">Change Bold</button>
20 </body>
21 </html>
```

Javascript Introduction: Hide the HTML element

```
1  <html lang="en">
2  <head>
3      <title>Test 6</title>
4      <script>
5          function show() {
6              var p = document.getElementById('myText');
7              p.style.display = 'block';
8          }
9
10         function hide() {
11             var p = document.getElementById('myText');
12             p.style.display = 'none';
13         }
14     </script>
15 </head>
16 <body>
17     <p id="myText" style="font-size:20px;">Ini adalah test</p>
18     <button onclick="show()">Show</button>
19     <button onclick="hide()">Hide</button>
20 </body>
21 </html>
```


Javascript Introduction: Show the Date

```
<html>
  <head>
    <title>Ini adalah judul website</title>
  </head>
  <body>
    <button onclick="test()">Munculkan Alert</button>
    <script>
      function test() {
        var date = new Date();
        alert(date);
      }
    </script>
  </body>
</html>
```

Statement



Statement: What is a Program?

1. Sebuah program merupakan kumpulan dari instruksi yang akan dijalankan oleh komputer.
2. Dalam bahasa pemrograman, instruksi tersebut dikenal dengan istilah statement.
3. Jadi sebuah program javascript merupakan kumpulan dari statement dalam bahasa javascript.
4. Di dalam file HTML, program javascript dijalankan di dalam sebuah browser.

Statement (Continue)

- Sebuah statement javascript tersusun oleh beberapa hal entiti berikut: Values, Operators, Expressions, Keywords, and Comments.
- Perhatikan sebuah statement berikut:

```
document.getElementById( 'myText' );
```

- Sebuah program javascript umumnya mengandung lebih dari satu statement.
- Semua statement tersebut dijalankan baris per baris sesuai dengan urutan barisnya.

Statement: Semicolon

- Semicolon (;) digunakan sebagai akhir (pembatas) dari sebuah statement.
- Semicolon jg digunakan oleh hampir semua bahasa pemrograman kecuali python.
- Perhatikan program berikut:

```
document.getElementById( 'myText' );
```

- Ada berapa statement pada program di atas?

Statement: Semicolon (Continue)

- Beberapa statement juga bisa ditulis dalam satu baris. Perhatikan program berikut:

```
a = 5; b = 6; c = a + b;
```

- Ada berapa statement pada program di atas?
- Pada beberapa web, anda mungkin melihat ada statement dari program javascript yang tidak diakhiri dengan semicolon.
- Statement tanpa semicolon tidak akan menyebabkan error, tapi penggunaan semicolon pada statement sangat dianjurkan

Statement: White Space

- Spasi pada javascript tidak memberi arti apa2 (diabaikan)
- Tapi anda bisa menambahkan spasi pada statement javascript untuk membuat program anda mudah dibaca dan dipahami
- Program program berikut:

```
var person = "Hege";  
var person="Hege";
```

- Dua statement di atas adalah sama.

Statement: White Space (Continue)

- Salah satu tips yang bagus adalah memberikan spasi pada setiap operator (= + - * /):
- Perhatikan contoh program berikut:

```
var x = y + z;
```

- Pemberian spasi pada operator, membuat program lebih mudah dibaca.

Statement: Line Length & Line Breaks

- Agar sebuah program mudah dibaca, biasanya programmer membatasi panjang sebuah statement adalah 80 karakter.
- Statement yang lebih dari 80 karakter akan dipecah menjadi beberapa dengan memberi line break (enter).
- Posisi yang tepat untuk memecah sebuah statement adalah setelah operator.
- Perhatikan contoh program berikut:

```
document.getElementById("demo").innerHTML =  
"Hello Dolly!";
```

Statement: Code Block

- Beberapa statement dapat dikelompokkan dalam sebuah blok ({}).
- Tujuan dari pengelompokkan ini adalah agar statement-statement tersebut dijalankan bersama.
- Contoh sebuah blok pada javascript adalah fungsi. Perhatikan contoh berikut:

```
function myFunction() {  
    document.getElementById("demo1").innerHTML = "Hello Dolly!";  
    document.getElementById("demo2").innerHTML = "How are you?";  
}
```

Statement: Keyword

- Sebuah statement sering dimulai dengan sebuah keyword untuk menunjukkan aksi yang akan dikerjakan oleh statement tersebut.
- Beberapa keyword pada javascript adalah sebagai berikut (lihat slide berikut):

Statement: Keyword (Continue)

Keyword	Description
break	Terminates a switch or a loop
continue	Jumps out of a loop and starts at the top
debugger	Stops the execution of JavaScript, and calls (if available) the debugging function
do ... while	Executes a block of statements, and repeats the block, while a condition is true
for	Marks a block of statements to be executed, as long as a condition is true
function	Declares a function
if ... else	Marks a block of statements to be executed, depending on a condition
return	Exits a function
switch	Marks a block of statements to be executed, depending on different cases
try ... catch	Implements error handling to a block of statements
var	Declares a variable

Syntax



Syntax: Definitions

- Syntax merupakan kumpulan dari aturan2 bagaimana menyusun sebuah program dalam Javascript.
- Perhatikan sintaks-sintaks berikut:

```
var x, y;           // How to declare variables
x = 5; y = 6;       // How to assign values
z = x + y;          // How to compute values
```


Syntax: Definitions (Continue)

- Sintaks mendefinisikan dua jenis value: **Fixed Value** dan **Variable Value**.
- Fixed Value disebut dengan Literals
- Variable value disebut dengan Variable.

Syntax: Literals

- Terdapat dua jenis literal yang paling sering dipakai, yaitu:
- Number: untuk menuliskan angka/bilangan.

```
10.50
```

```
1001
```

- String: untuk menuliskan sebuah text.

```
"John Doe"
```

```
'John Doe'
```

Syntax: Variables

- Dalam sebuah bahasa pemrograman, variable digunakan untuk menyimpan nilai data.
- Untuk mendefinisikan sebuah variable pada Javascript, anda bisa menggunakan keyword **var**.
- Untuk memberikan nilai pada sebuah variable, anda bisa menggunakan operator =
- Perhatikan contoh berikut:

```
var x;  
  
x = 6;
```

Syntax: Operator

- Operator biasa dipakai untuk operasi matematika.
- Beberapa operator yang biasa dipakai adalah $+$, $-$, $*$, $/$
- Perhatikan contoh program berikut:

```
(5 + 6) * 10
```

- Untuk memberikan nilai sebuah variable, gunakan operator =

Syntax: Expression

- Expression merupakan gabungan dari value, variable dan operator.
- Dalam sebuah expression tersirat sebuah perhitungan yang menghasilkan nilai tertentu.

```
5 * 10
```

- Contoh expression yang mengandung variable

```
x * 10
```

Syntax: Comment

- Anda bisa memberikan komentar/keterangan pada program anda dengan menggunakan // atau /* dan */
- Comment pada program tidak akan dijalankan dan biasanya digunakan untuk meletakan dokumentasi
- Perhatikan contoh program berikut:

```
var x = 5;    // I will be executed  
  
// var x = 6;    I will NOT be executed
```


Syntax: Case Sensitive

- Javascript merupakan bahasa pemrograman yang case sensitive
- Artinya huruf besar dan huruf kecil adalah dua hal yang berbeda.
- Perhatikan contoh program berikut:

```
var lastname, lastName;  
lastName = "Doe";  
lastname = "Peterson";
```

- Pada contoh program di atas, **lastName** dan **lastname** adalah dua buah variable yang berbeda.

Syntax: CamelCase

- Sangat dianjurkan untuk memberikan nama variable mengikuti pola CamelCase, yaitu mengikuti beberapa aturan berikut:
 - Terdiri dari huruf kecil
 - Jika terdiri dari lebih dari 2 kata maka ditulis bersambung
 - Huruf pertama dari kata kedua dan seterusnya dimulai dengan huruf besar.
- Perhatikan contoh berikut: firstName, lastName, masterCard, interCity, dll

Variable



Variable: Definitions

- Variable adalah komponen untuk menyimpan sebuah nilai.
- Untuk mendeklarasikan sebuah variable, anda bisa menggunakan keyword var.
- Perhatikan contoh berikut:

```
var x = 5;  
var y = 6;  
var z = x + y;
```

- Variable bisa bertipe string, number, boolean, array, dll

Variable: Type of String Data

- String merupakan kumpulan dari beberapa karakter. Atau biasa disebut dengan text.
- String dapat dituliskan menggunakan double quote (") atau single quote (')
- Perhatikan contoh berikut:

```
var carName1 = "Volvo XC60";    // Using double quotes  
var carName2 = 'Volvo XC60';    // Using single quotes
```

Variable: Type of String Data (Continue)

- Anda bisa meletakkan quote (tanda kutip) dalam sebuah string selama tanda kutip tersebut berbeda dengan tanda kutip yang digunakan untuk membentuk string tersebut.
- Perhatikan contoh berikut:

```
var answer1 = "It's alright";           // Single quote inside double quotes
var answer2 = "He is called 'Johnny'";  // Single quotes inside double quotes
var answer3 = 'He is called "Johnny"';  // Double quotes inside single quotes
```


Variable: Type of Number Data

- Berbeda dengan bahasa pemrograman yang lain, Javascript hanya memiliki satu jenis tipe data number.
- Tipe data number tersebut bisa digunakan untuk menyimpan nilai integer atau desimal.
- Perhatikan contoh berikut:

```
var x1 = 34.00;    // Written with decimals  
var x2 = 34;       // Written without decimals
```

Variable: Type of Array Data

- Array merupakan koleksi/kumpulan dari beberapa elemen.
- Elemen-elemen tersebut bisa berupa nilai atau variable yang lain.
- Untuk mendeklarasikan sebuah array, anda bisa menggunakan tanda kurung siku.
- Setiap elemen dipisah menggunakan tanda koma.
- Perhatikan contoh berikut:

```
var cars = ["Saab", "Volvo", "BMW"];
```

Variable: Type of Array Data (Continue)

- Untuk mengakses elemen-elemen dari array, adan bisa menggunakan index dari elemen tersebut.
- Index pada array dimulai dari 0.

```
var name = cars[0];
```

Variable: Object

- Mirip seperti array, yakni sebuah variable yang mengandung beberapa variable sebagai properti dari object tersebut.
- Untuk mendefinisikan sebuah object, anda bisa menggunakan tanda kurung kurawal.
- Properti dari object dituliskan dalam pasangan key:value
- Antara properti satu dengan yang lain dipisah dengan tanda koma,
- Perhatikan contoh berikut:

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

Variable: typeof Operator

- **typeof** operator digunakan untuk menentukan jenis tipe data dari sebuah variable.
- Perhatikan contoh berikut:

```
typeof ""           // Returns "string"  
typeof "John"       // Returns "string"  
typeof "John Doe"   // Returns "string"
```

```
typeof 0            // Returns "number"  
typeof 314          // Returns "number"  
typeof 3.14         // Returns "number"  
typeof (3)          // Returns "number"  
typeof (3 + 4)      // Returns "number"
```

Variable: undefined

- Sebuah variable yang belum diberikan value, maka nilai dari variable tersebut adalah **undefined**.
- Begitu juga dengan tipe data untuk variable tersebut adalah **undefined**.
- Perhatikan contoh berikut:

```
var car;    // Value is undefined, type is undefined
```


Function



Variable: Function

- Sebuah fungsi merupakan blok dari statement yang bertujuan untuk mengerjakan task tertentu.
- Perhatikan contoh fungsi berikut:

```
function myFunction(p1, p2) {  
    return p1 * p2;    // The function returns the product of p1 and p2  
}
```

Variable: Function (Continue)

- Sebuah fungsi didefinisikan menggunakan keyword **function** kemudian diikuti oleh nama fungsi dan kurung buka-tutup.
- Nama fungsi bisa mengandung huruf, angka, garis bawah dan dollar.
- Di antara kurung buka-tutup bisa disisipkan parameter (dipisah dengan koma jika lebih dari satu)
- Kode-kode dari fungsi tersebut diletakan di dalam tanda kurung kurawal.
- Perhatikan format berikut:

Variable: Function (Continue)

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

Variable: Call Function

- Sebuah fungsi yang sudah didefinisikan bisa dipanggil dari baris kode yang lain
- Pemanggilan fungsi adalah dengan cara menuliskan nama fungsi tersebut dan kurang buka-tutup
- Jangan lupa memberikan parameter diantara kurung buka-tutup jika fungsi tersebut membutuhkan parameter.
- Perhatikan contoh berikut:

```
myFunction();  
myFunction(a, b);
```

Variable: Function Return

- Sebuah fungsi dapat mengembalikan nilai kepada **pemanggilnya**.
- Untuk mendefinisikan sebuah fungsi dapat mengembalikan nilai, anda bisa menggunakan keyword return.
- Perhatikan contoh berikut:

```
var x = myFunction(4, 3); // Function is called, return value will end up in x

function myFunction(a, b) {
  return a * b;           // Function returns the product of a and b
}
```

Variable: Why Function?

- Reuse Code: anda bisa menggunakan code yang sama berulang kali tanpa harus menuliskan-nya lagi.
- Anda bisa menggunakan coe yang sama menggunakan parameter yang berbeda untuk menghasilkan nilai yang berbeda.

Variable: Local Variable

- Sebuah variable yang dideklarasikan dalam sebuah fungsi disebut sebagai local variable untuk fungsi tersebut.
- Sebuah local variable tidak dapat diakses di luar scope dimana variable tersebut dideklarasikan.
- Dalam hal ini scope untuk variable tersebut adalah fungsi dimana variable tersebut dideklarasikan.
- Perhatikan contoh berikut:

```
// code here can NOT use carName

function myFunction() {
    var carName = "Volvo";
    // code here CAN use carName
}

// code here can NOT use carName
```

Variable: Built-in Functions

- Pada contoh sebelumnya, kita telah mencoba fungsi **getElementById**. Fungsi ini adalah untuk mencari elemen HTML dengan id tertentu pada dokumen HTML.
- Selain **getElementById**, ada beberapa fungsi lain:
 1. **getElementsByTagName**: mencari elemen HTML berdasarkan nama tag. Misal: **document.getElementsByTagName("h1")**.
 2. **getElementsByClassName**: mencari elemen HTML berdasarkan css class. Misal: **document.getElementsByClassName("center")**.
 3. **querySelectorAll**: mencari elemen HTML berdasarkan css selector. Misal: **document.querySelectorAll("p.center")**.

Variable: Javascript `getElementsBy`

- Perhatikan baik-baik perbedaannya, ketiga fungsi terakhir bisa mengembalikan lebih dari satu elemen, berbeda dengan **`getElementById`** yang hanya mengembalikan 1 elemen.
- Ketiga fungsi terakhir mengembalikan object berupa array.

Examples



Examples: Convert Celsius to Fahrenheit

main.js



Run

Output

```
1 // program to convert celsius to fahrenheit
2 // ask the celsius value to the user
3 const celsius = prompt("Enter a celsius value: ");
4
5 // calculate fahrenheit
6 const fahrenheit = (celsius * 1.8) + 32
7
8 // display the result
9 console.log(`${celsius} degree celsius is equal to ${fahrenheit} degree fahrenheit.`);
```

```
node /tmp/c1QFzaaCp8.js
Enter a celsius value: 34
34 degree celsius is equal to 93.2 degree fahrenheit.
|
```

Examples: Multiplication Table Up to 20

main.js



Run

Output

```
1 // program to generate a multiplication table
2
3 // take input from the user
4 const number = parseInt(prompt('Enter an integer: '));
5
6 //creating a multiplication table
7 for(let i = 1; i <= 10; i++) {
8
9     // multiply i with number
10    const result = i * number;
11
12    // display the result
13    console.log(`${number} * ${i} = ${result}`);
14 }
```

```
node /tmp/8g1AeGBKtr.js
Enter an integer: 20
20 * 1 = 20
20 * 2 = 40
20 * 3 = 60
20 * 4 = 80
20 * 5 = 100
20 * 6 = 120
20 * 7 = 140
20 * 8 = 160
20 * 9 = 180
20 * 10 = 200
|
```

Examples: Generate Random Strings

main.js



Run

Output

```
1 // program to generate random strings
2
3 // declare all characters
4 const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
5
6 function generateString(length) {
7     let result = ' ';
8     const charactersLength = characters.length;
9     for ( let i = 0; i < length; i++ ) {
10         result += characters.charAt(Math.floor(Math.random() * charactersLength));
11     }
12     ..
13     return result;
14 }
15
16 console.log(generateString(5));
```

```
node /tmp/Khe7kgrbii.js
af4Ew
```


References

1. Lynch. (2011). Site File Structure Web Style Guide 3. <https://webstyleguide.com/wsg3/5-site-structure/3-site-file-structure.html>
2. Patrick J. Lynch, S. H. (2011). Web Style Guide, Third edition. In Web Style Guide. Sarah Horton, Patrick Lynch. <http://webstyleguide.com/wsg3/4-interface-design/>
3. <https://www.w3schools.com/js/default.asp>
4. <https://www.programiz.com/javascript/examples>

THANK YOU



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web Pertemuan 7 Penggunaan Bootstrap

Zuhri Halim, S.Kom., M.Kom.

Outline

1. Introduction
2. Reflection
3. Bootstrap Introduction
4. Use of Bootstrap
5. Container in Bootstrap
6. Column in Bootstrap
7. Class in Bootstrap
8. Exercises



Introduction: Learning Outcomes

Mahasiswa menganalisis layout dan estetika sebuah web yang menggunakan Bootstrap.

Reflection

فَبِمَا رَحْمَةٍ مِنَ اللَّهِ لِنْتَ لَهُمْ^ط وَلَوْ كُنْتَ فَظًّا غَلِيظَ الْقَلْبِ لَانْفَضُّوا مِنْ
حَوْلِكَ^ط فَاعْفُ عَنْهُمْ وَاسْتَغْفِرْ لَهُمْ وَشَاوِرْهُمْ فِي الْأَمْرِ^ط فَإِذَا عَزَمْتَ فَتَوَكَّلْ
عَلَى اللَّهِ^ج إِنَّ اللَّهَ يُحِبُّ الْمُتَوَكِّلِينَ ﴿١٥٩﴾

Maka disebabkan rahmat dari Allah-lah kamu berlaku lemah lembut terhadap mereka. Sekiranya kamu bersikap keras lagi berhati kasar, tentulah mereka menjauhkan diri dari sekelilingmu. Karena itu maafkanlah mereka, mohonkanlah ampun bagi mereka, dan bermusyawaratlah dengan mereka dalam urusan itu. Kemudian apabila kamu telah membulatkan tekad, maka bertawakkallah kepada Allah. Sesungguhnya Allah menyukai orang-orang yang bertawakkal kepada-Nya (QS. Ali Imran: 159).

Bootstrap Introduction



Bootstrap Introduction

Bootstrap adalah front-end framework yang solek, bagus dan luar biasa yang mendapatkan tampilan untuk mobile device guna mempercepat dan mempermudah pengembangan website. Bootstrap menyediakan HTML, CSS dan Javascript siap pakai dan mudah untuk dikembangkan.



Bootstrap Introduction: Why?

- Mobile friendly: dengan sistem grid, proses membuat website yang mobile friendly tidak membutuhkan waktu lama.
- Easy to resize images: dengan menambahkan class “.img-responsive” ke gambar, maka gambar tersebut akan otomatis di resize sesuai ukuran layer pengguna.
- Easy to add element: bootstrap menyediakan berbagai elemen yang dapat langsung Anda gunakan di website. Misalnya, navigasi, menu dropdown, thumbnail, dan sebagainya.
- Intractive: memungkinkan plugin custom JQuery. Misal, menambahkan berbagai element interaktif seperti popup transisi image, carouse, dan lain-lain.

Bootstrap Introduction: Preparation

- Jika ingin menggunakan bootstrap secara offline, maka harus mendownload pada website resmi di <https://getbootstrap.com/docs/5.3/getting-started/download/>.
- Memasukkan link bootstrap dari CDN (Content Delivery Network).
- Software code editor (PHPStorm, Notepad++, dll).

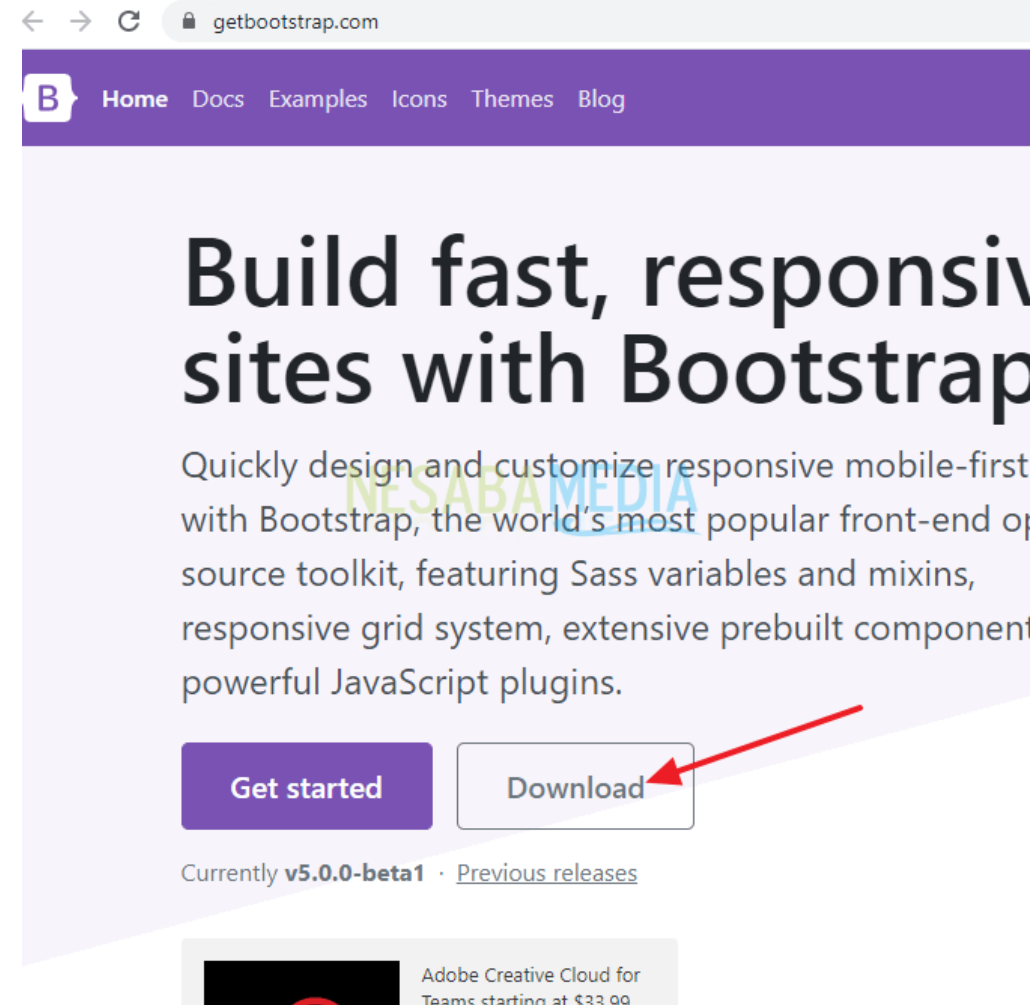


Use of Bootstrap



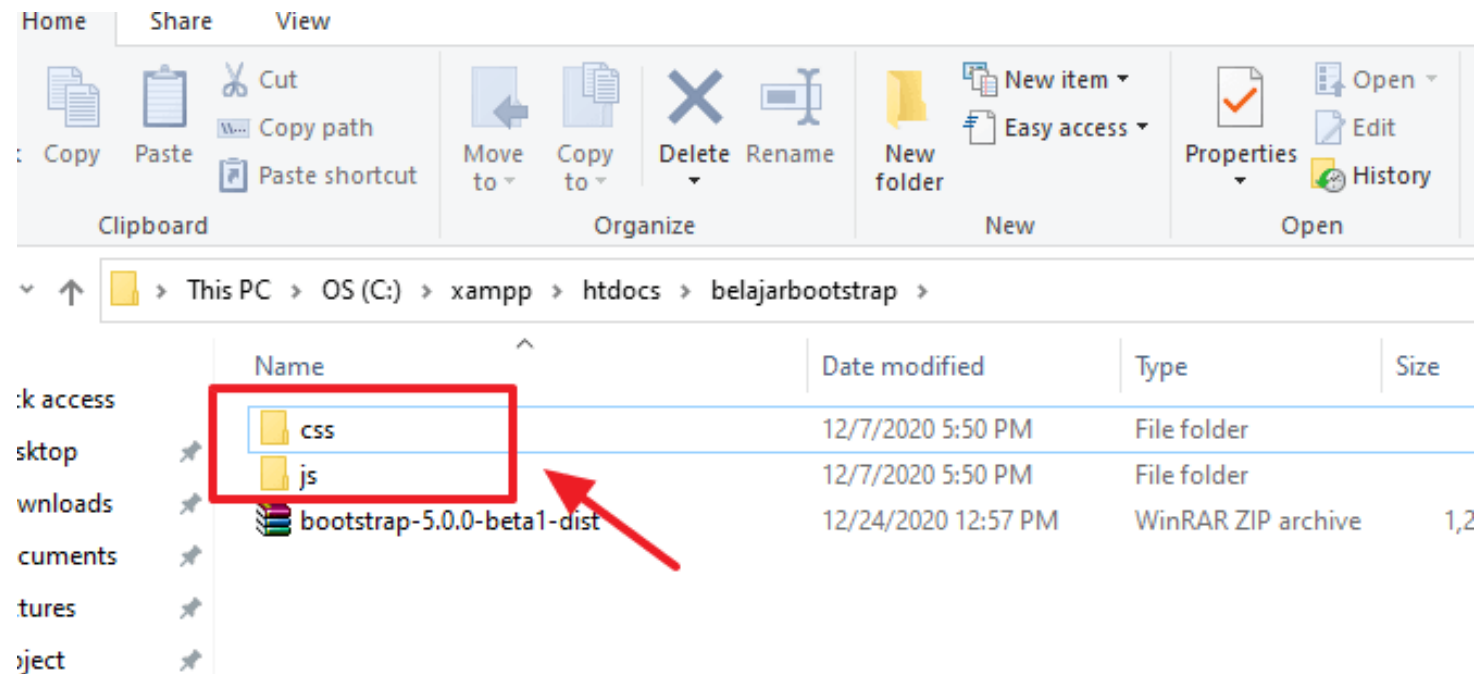
Use of Bootstrap: Download Bootstrap

Buka web browser di PC ataupun laptop Anda. Lalu, silahkan Anda ketikkan <https://getbootstrap.com/>.



Use of Bootstrap: Download Bootstrap

- Setelah proses mengunduh selesai, Anda akan mendapatkan file Bootstrap yang berupa .rar.
- Lalu, silahkan Anda buat folder baru di dalam htdocs, misalnya Anda membuat folder baru dengan nama “belajarbootstrap”. Silahkan Anda pindahkan file .rar tadi ke dalam folder baru tersebut. Seperti pada gambar dibawah ini.



Use of Bootstrap: Insert Bootstrap (Offline)

Buka software editor Anda dan ketikkan script code berdasarkan disamping ini. Untuk script code dibawah ini untuk menghubungkan bootstrap secara offline.

Anda tidak perlu menghubungkan seluruh file bootstrap yang ada. Anda hanya perlu menghubungkan file bootstrap.min.css (production) / bootstrap.css (development) dan bootstrap.js.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <title>Belajar Bootstrap 5</title>
7      <meta name="viewport" content="width=device-width, initial-scale=1">
8      <!-- Menvisipkan Bootstrap -->
9      <link rel="stylesheet" href="css/bootstrap.min.css" />
10     <!-- Menvisipkan JQuery dan Javascript Bootstrap -->
11     <script src="js/bootstrap.min.js"></script>
12 </head>
13 <body>
14     <!-- Membuat komponen jumbotron sebagai header -->
15     <header>
16         <div class="jumbotron jumbotron-fluid">
17             <div class="container">
18                 <div class="row">
19                     <div class="col">
20                         <h1>Welcome to My Web</h1>
21                         <div class="lead">Aku sedang belajar Bootstrap 5</div>
22                     </div>
23                 </div>
24             </div>
25         </div>
26     </header>
27 </body>
28 </html>
```


Use of Bootstrap: Type Files

Terdapat perbedaan jenis bootstrap, antara lain sebagai berikut:

- `bootstrap.min.css`: file bootstrap yang sudah dikompres, yang dimana dari kapasitas lebih kecil. Secara umum digunakan untuk produksi (production).
- `bootstrap.css`: file yang digunakan untuk proses pengembangan (development).

Use of Bootstrap: Bootstrap CDN (Online)

Cara menghubungkan bootstrap (online) melalui CDN dengan menuliskan link berdasarkan gambar dibawah ini.

Bootstrap CDN

The folks over at [jsDelivr](#) graciously provide CDN support for Bootstrap's CSS and JavaScript. Just use these [Bootstrap CDN](#) links.

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css" integrity="sha384-
BVYiiSiFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

<!-- Optional theme -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap-
theme.min.css" integrity="sha384-rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwL/Sp"
crossorigin="anonymous">

<!-- Latest compiled and minified JavaScript -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHj0MaLkfuWVxZxUPnCJA7l2mCWNIPg9mGCD8wGNIcPD7Txa"
crossorigin="anonymous"></script>
```

Use of Bootstrap: Tag Responsive

Line code di bawah ini agar desain web menjadi lebih responsive. Perlu di perhatikan pemanggilan bootstrap.js harus sesudah pemanggilan jquery.js , jadi jangan pernah memanggil file bootstrap.js terlebih dahulu sebelum jquery.js kalau javascript bootstrap ingin berjalan.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Container in Bootstrap



Container in Bootstrap

- Pertama “.container” menjadikan lebar menjadi tidak full.
- Kedua “.container-fluid” menjadikan memenuhi lebar browser.



Column in Bootstrap



Column in Bootstrap

Bootstrap menggunakan sampai dengan 12 column, yang dimana column standar lebar yang umum digunakan terdiri dari column 3 , column 4, column 6, column 8, column 12 .

```
<div class="col-xs-4"> column extra small </div>
```

```
<div class="col-sm-4"> column small </div>
```

```
<div class="col-md-4"> column medium </div>
```

```
<div class="col-lg-4"> column large </div>
```


Class in Bootstrap



Class in Bootstrap

Ada beberapa warna background di bootstrap yang tersedia, pemanggilan class sama seperti warna teks, hanya saja berbeda dengan nama awalnya. Jika untuk text menggunakan .text-success, tapi jika warna background menggunakan .bg-primary, .bg-success, bg-info, bg-warning, dan .bg-danger

`.text-muted, .text-primary, .text-success, .text-info, .text-warning, and .text-danger`

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Exercises



Exercises: Create Folder

Buatlah folder dengan nama “Pert7” dan hierarki berikut ini.

- Pert7
 - folder images
 - folder css (source from bootstrap)
 - Folder js (source from bootstrap)
 - Files html: index.html, form.html, dll

Exercises: Form Input with Bootstrap

Buatlah form input menggunakan dengan framework bootstrap berdasarkan gambar dibawah ini.

Email address

name@example.com

Example textarea

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!-- basic -->
5     <meta charset="utf-8">
6     <!-- Compatible with edge browser -->
7     <meta http-equiv="X-UA-Compatible" content="IE=edge">
8     <!-- mobile metas -->
9     <meta name="viewport" content="width=device-width, initial-scale=1">
10    <!-- site metas -->
11    <meta name="viewport" content="initial-scale=1, maximum-scale=1">
12    <title>Form</title>
13    <!-- Insert Bootstrap -->
14    <link rel="stylesheet" href="css/bootstrap.css"/>
15    <!-- Insert JQuery & Javascript Bootstrap -->
16    <script src="js/bootstrap.js"></script>
17  </head>
18  <body>
19
20    <div class="container-fluid">
21      <div class="mb-3">
22        <label for="exampleFormControlInput1" class="form-label">Email address</label>
23        <input type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">
24      </div>
25      <div class="mb-3">
26        <label for="exampleFormControlTextarea1" class="form-label">Example textarea</label>
27        <textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
28      </div>
29    </div>
30  </body>
31 </html>
```

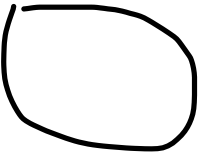
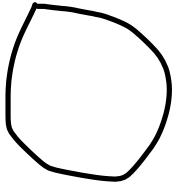
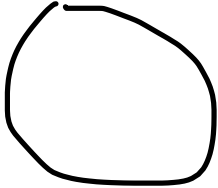
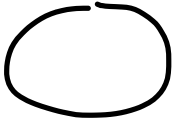
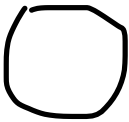
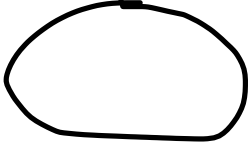
Exercises: Navigation Bar

Buatlah navigation bar/menu yang terdiri dari sebagai berikut:

- Home
- About
 - Profile
 - History
 - Services
- Portfolio
- Contact

Exercises: Column

Buatlah column berdasarkan format dibawah ini dan masukan gambar.

References

1. Lynch. (2011). Site File Structure Web Style Guide 3. <https://webstyleguide.com/wsg3/5-site-structure/3-site-file-structure.html>
2. Patrick J. Lynch, S. H. (2011). Web Style Guide, Third edition. In Web Style Guide. Sarah Horton, Patrick Lynch. <http://webstyleguide.com/wsg3/4-interface-design/>
3. <https://getbootstrap.com/>
4. <https://www.w3schools.com/bootstrap5/index.php>

THANK YOU



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web

Pertemuan 9

Grid Bootstrap

Zuhri Halim, S.Kom., M.Kom.

Outline

1. Introduction
2. Reflection
3. Grid Bootstrap
4. Grid Example



Introduction: Learning Outcomes

Mahasiswa menganalisis layout dan estetika sebuah web yang menggunakan grid bootstrap

Reflection

خَلَقَ اللَّهُ السَّمَاوَاتِ وَالْأَرْضَ بِالْحَقِّ ۚ إِنَّ فِي ذَلِكَ لَآيَةً لِّلْمُؤْمِنِينَ

﴿ ٤٤ ﴾

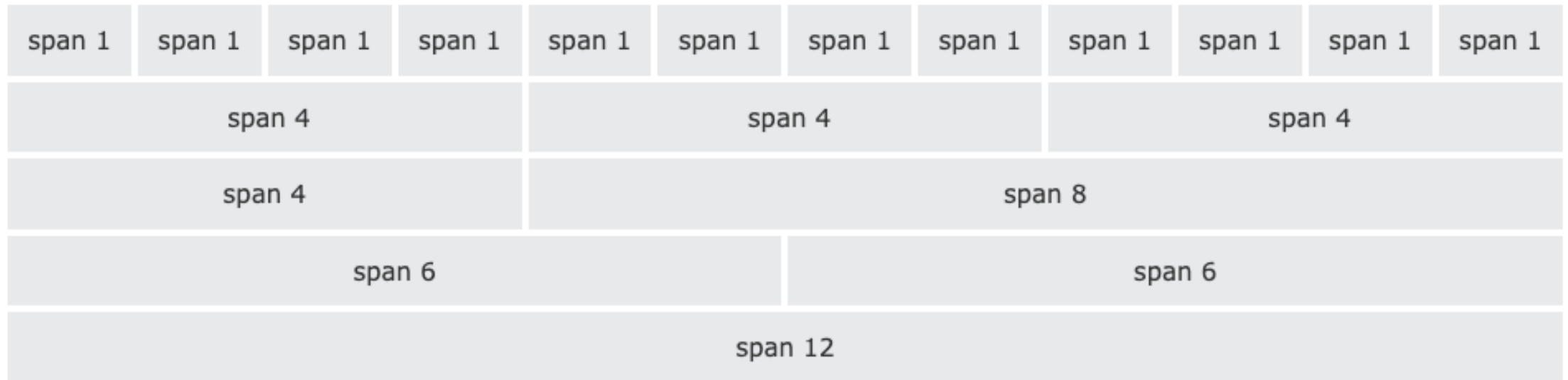
(QS. Al-Ankabut : 44) Allah menciptakan langit dan bumi dengan hak. Sesungguhnya pada yang demikian itu terdapat tanda-tanda kekuasaan Allah bagi orang-orang mukmin.

Grid Bootstrap



Grid Bootstrap

Grid pada Bootstrap merupakan pengaturan ukuran yang di tampilkan pada monitor. Grid Bootstrap berfungsi untuk membuat pengaturan untuk lebar dari masing-masing komponen web sehingga kita dapat dengan bebas mengatur ke-responsivan halaman website yang kita buat dengan Bootstrap. Bootstrap memiliki 12 class grid.



Grid Bootstrap: Columns & Class Row

Grid pada Bootstrap merupakan pengaturan ukuran yang di tampilkan pada monitor. Grid Bootstrap berfungsi untuk membuat pengaturan untuk lebar dari masing-masing komponen web sehingga kita dapat dengan bebas mengatur ke-responsivan halaman website yang kita buat dengan Bootstrap. Bootstrap memiliki 12 class grid.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Grid Bootstrap: Classes

The Bootstrap grid system has four classes:

- xs (for phones - screens less than 768px wide)
- sm (for tablets - screens equal to or greater than 768px wide)
- md (for small laptops - screens equal to or greater than 992px wide)
- lg (for laptops and desktops - screens equal to or greater than 1200px wide)

The classes above can be combined to create more dynamic and flexible layouts.

Tip: Each class scales up, so if you wish to set the same widths for xs and sm, you only need to specify xs.

Grid Bootstrap: Rules

- Tiap baris span harus diletakkan dalam sebuah **.container** (fixed-width) atau **.container-fluid** (full-width) untuk alignment dan padding.
- Gunakan **.row** untuk membuat grup horizontal dari setiap column (**.col-xxx**).
- Konten harus diletakkan diantar column dan hanya column yang memungkinkan untuk membuat turunan/sub dari **.row**.
- Class grid yang telah ditentukan seperti **.row** dan **.col-xs-x** tersedia untuk mempercepat pembuatan grid layouts.
- Class column dibuat dengan menentukan nomor dari 12 column yang tersedia.
- Jika dalam **.row** memiliki lebih dari 12 column, maka akan diletakkan pada baris baru yang sebenarnya merupakan masih satu-kesatuan.
- Jika ada class yang secara resolusi dibesarkan, tetapi secara lebar memiliki kemiripan seperti **xs** dan **sm**, maka Anda hanya perlu menentukan **xs**.

Grid Bootstrap: Basic Structure

```
<div class="container">
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    ...
  </div>
</div>
```

Secara struktur seperti berikut ini:

- Membuat (<div class="container">)
- Menambahkan row dengan jumlah span yang diinginkan .col-sm-4.
- Dalam menentukan column harus mengacu ke jenis span.

Grid Example



Grid Example: Stacked-to-Horizontal

col-sm-6

col-sm-6

Jenis grid yang memiliki 2 column layout yang dimana di split menjadi 50%-50%.

```
<div class="container">
  <h1>Hello World!</h1>
  <div class="row">
    <div class="col-sm-6" style="background-color:yellow;">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-6" style="background-color:pink;">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```


Grid Example: Small Devices

col-sm-3

col-sm-9

Jenis grid yang memiliki 2 column layout (tablet screen).

```
<div class="container-fluid">
  <h1>Hello World!</h1>
  <div class="row">
    <div class="col-sm-3" style="background-color:yellow;">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-9" style="background-color:pink;">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

Grid Example: Medium Devices

col-sm-3

col-sm-9

Jenis grid yang memiliki 2 column layout (small laptop/notebook screen).

```
<div class="container-fluid">
  <h1>Hello World!</h1>
  <div class="row">
    <div class="col-sm-3" style="background-color:yellow;">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-9" style="background-color:pink;">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

Grid Example: Large Devices

col-sm-3

col-sm-9

Jenis grid yang memiliki 2 column layout (laptop/desktop screen).

```
<div class="container-fluid">
  <h1>Hello World!</h1>
  <div class="row">
    <div class="col-sm-3 col-md-6 col-lg-4" style="background-
color:yellow;">
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-9 col-md-6 col-lg-8" style="background-
color:pink;">
      <p>Sed ut perspiciatis...</p>
    </div>
  </div>
</div>
```

Grid Example: No Gutters

No gutters

Use the `.row-no-gutters` class to remove the gutters from a row and its columns:



And here's a row with gutters to demonstrate the differences:



Apa yang menjadi pembeda?

Grid Example: No Gutters (Continue)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
  <h1>No gutters</h1>
  <p>Use the .row-no-gutters class to remove the gutters from a row and its columns:</p>
  <div class="row row-no-gutters">
    <div class="col-sm-4" style="background-color:lavender;">.col-sm-4</div>
    <div class="col-sm-8" style="background-color:lavenderblush;">.col-sm-8</div>
  </div>
  <br>
  <p>And here's a row with gutters to demonstrate the differences:</p>
  <div class="row">
    <div class="col-sm-4" style="background-color:lavender;">.col-sm-4</div>
    <div class="col-sm-8" style="background-color:lavenderblush;">.col-sm-8</div>
  </div>
</div>

</body>
</html>
```

References

1. Lynch. (2011). Site File Structure Web Style Guide 3. <https://webstyleguide.com/wsg3/5-site-structure/3-site-file-structure.html>
2. Patrick J. Lynch, S. H. (2011). Web Style Guide, Third edition. In Web Style Guide. Sarah Horton, Patrick Lynch. <http://webstyleguide.com/wsg3/4-interface-design/>
3. <https://getbootstrap.com/>
4. <https://www.w3schools.com/bootstrap5/index.php>

THANK YOU



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web Pertemuan 10 Single Page Web Application

Zuhri Halim, S.Kom., M.Kom.

Outline

1. JQuery Introduction
2. Single Page Application (SPA)
3. Modules in SPA
4. Client-Side Routing
5. View Composition & Layout
6. Examples



Introduction: Learning Outcomes

Mahasiswa mampu menjelaskan konsep Single Page Apps (SPA) dan metode untuk membangunnya

JQuery Introduction



JQuery Introduction: Definition

- JQuery adalah Javascript Library / Kumpulan kode/fungsi Javascript siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode Javascript.
- Secara standar, Apabila kita membuat kode Javascript, maka diperlukan kode yang cukup panjang, bahkan terkadang sangat sulit dipahami,

jQuery Introduction: Benefits?

- Mempermudah akses dan manipulasi elemen tertentu pada dokumen.
- Mempermudah modifikasi/perubahan tampilan halaman web.
- Mampu merespon interaktif antara user dengan halaman web dengan lebih cepat.
- Menyediakan fasilitas untuk membuat animasi sekelas Flash dengan mudah.

JQuery Introduction: How to Use?

- Kita mengunduh terlebih dahulu, library jQuery di <http://jquery.com>.
- Selanjutnya Selanjutnya panggil library jQuery diantara tag <head>...</head>

```
<html>
<head>
<script type = "text/javascript" src="jquery 1.4.js">
</script>
</head>
<body>
</body>
```


How to U

Selanjutnya kita bisa membuat script jQuery pada bagian head, semisal kita ingin menggunakan fungsi untuk menyembunyikan dan menampilkan sebuah gambar. Selain itu, bisa menggunakan CDN.

JQuery Introduction: How to Use? (Continue)

Setelah itu, kita bisa menggunakan id atau class di yang telah kita definisikan sebelumnya pada bagian body website.

```
13 <body>
14 <button class="sembunyi">sembunyikan</button> <button class="tampil">
    tampilkan</button> <p></p> </body>
15 </html>
```

JQuery Introduction: How to Work? (Continue)

JQuery akan memastikan bahwa semua elemen atau elemen yang dibagikan sudah ditampilkan semua dihalaman web, fungsi yang digunakan adalah:

```
$document.ready(function() {  
//baris kode jquery akan dijalankan  
//apabila semua elemen sudah ditampilkan semua  
})
```

JQuery Introduction: How to Work? (Continue)

Setelah semua elemen ditampilkan, tahap berikutnya adalah memilih elemen berdasarkan class atau id yang telah didefinisikan. Dalam hal ini, JQuery menggunakan fungsi **Selector**. Contoh skripnya:

```
$("#foto")  
$(".Sembu  
nyi")  
$(".tampi  
l")
```

JQuery Introduction: How to Work? (Continue)

Setelah elemen dipilih, tahap berikutnya adalah memberikan aksi/operasi/event terhadap elemen yang sudah dipilih. Misalnya, operasi klik dan hide, dimana apabila tombol diklik, maka akan menyembunyikan elemen foto.

```
$ (".sembunyi").click(function() {  
  $ ("#foto").hide("slow")  });
```

JQuery Introduction: Selector

Selectors berfungsi untuk memilih dan mengambil elemen – elemen tertentu yang ingin dilakukan operasi terhadap elemen tersebut. Atau manipulasi terhadap elemen – elemen tersebut.

JQuery Introduction: Selector (Continue)

- Selector dengan nama Tag

```
$(document).ready(function(){  
$("div").addClass("biru");});
```

- Selector Class. Dengan menambahkan # di awal nama selector

```
$("#daftar").addClass("merahkuning");});
```

- Selector id. Dengan menambahkan diawal selector

```
$(".subdaftar").addClass("kuninghitam");});</script>
```


JQuery Introduction: Event

Event adalah bagian yang memutuskan kapan dilakukan operasi tersebut, misalnya elemen yang telah dipilih (selectors) akan dilakukan operasi terhadapnya setelah diklik (klik) atau setelah klik 2x (double click), setelah mouse berada diatas objek (hover) dsb. Contoh event:

```
$("a").click(function(){  
  alert("selamat datang di website facebook");  
});  
});  
</script>  
</head>  
<body>  
<a href="http://facebook.com">  
klik disini untuk membuka website facebook</a> </body>  
</html>
```

JQuery Introduction: Event (Continue)

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

jQuery Introduction: Effect

Effects mengenai tentang menampilkan dan menyembunyikan suatu elemen, menggerakkan elemen, dan menganimasi elemen. Pada sub bab sebelumnya kita telah menggunakan event klik pada tombol, dimana apabila tombol sembunyikan diklik, maka elemen berupa foto akan hilang dan apabila diklik tombol tampilkan maka foto tersebut akan ditampilkan kembali. Pada contoh tersebut kita telah menggunakan dua buah effect yaitu show dan hide.

```
1 <html>
2 <head>
3 <script type="text/javascript" src="jquery-1.4.4.min.js"></script>
4 <script type="text/javascript">
5 $(document).ready(function(){
6   setTimeout(function(){
7     $("#kotak").fadeOut("slow",function(){
8       $("#kotak").remove();
9     });
10  }, 3000);
11 });
12 </script>
13 <style type="text/css">
```

jQuery Effects

- jQuery Hide/Show
- jQuery Fade
- jQuery Slide
- jQuery Animate
- jQuery stop()
- jQuery Callback
- jQuery Chaining

JQuery Introduction: Document Object Model (DOM)

- Salah satu penggunaan JQuery yang paling umum adalah untuk mengubah dan manipulasi dokumen HTML beserta attribute-nya.
- JQuery menyediakan 3 fungsi utama untuk mengambil konten, antara lain sebagai berikut:
 - `text()`: berfungsi membaca atau mengambil isi teks dari elemen terpilih.
 - `html()`: berfungsi membaca atau mengembalikan konten dari elemen terpilih, termasuk elemen HTML-nya.
 - `val()`: berfungsi membaca atau mengembalikan nilai dari kotak isian pada form.

DOM (Cont

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Mengambil Konek</title>
  <script src="jquery-3.1.1.min.js"></script>
  <script>(function($) {
    $.fn.click(function() {
      $("#btn1").click(function() {
        alert("Tekan = " + $("#kalimat").val());
      });
    });
    $.fn.click(function() {
      $("#btn2").click(function() {
        alert("HTML nya " + $("#kalimat").val());
      });
    });
  });
</script>
```

DOM (Cont

```
<body>
```

```
  <p id="ka" style="text-align: center;">Ini adalah teks <b>tebal</b> dan <i>miring</i>  
  </p>
```

```
  <button id="btn1" type="button">Tampilkan Teks</button>
```

```
  <button id="btn2" type="button">Kode HTML</button>
```

```
  </button>
```

```
</body>
```

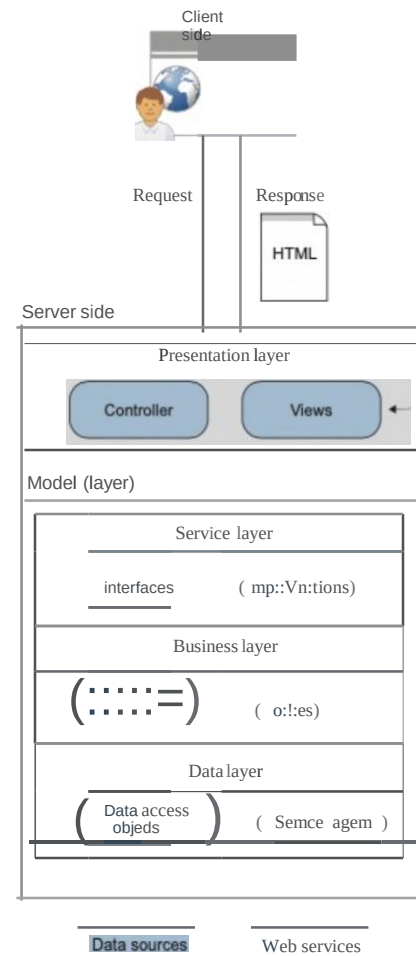
```
</html>
```

Single Page Application (SPA)



n (SPA): D

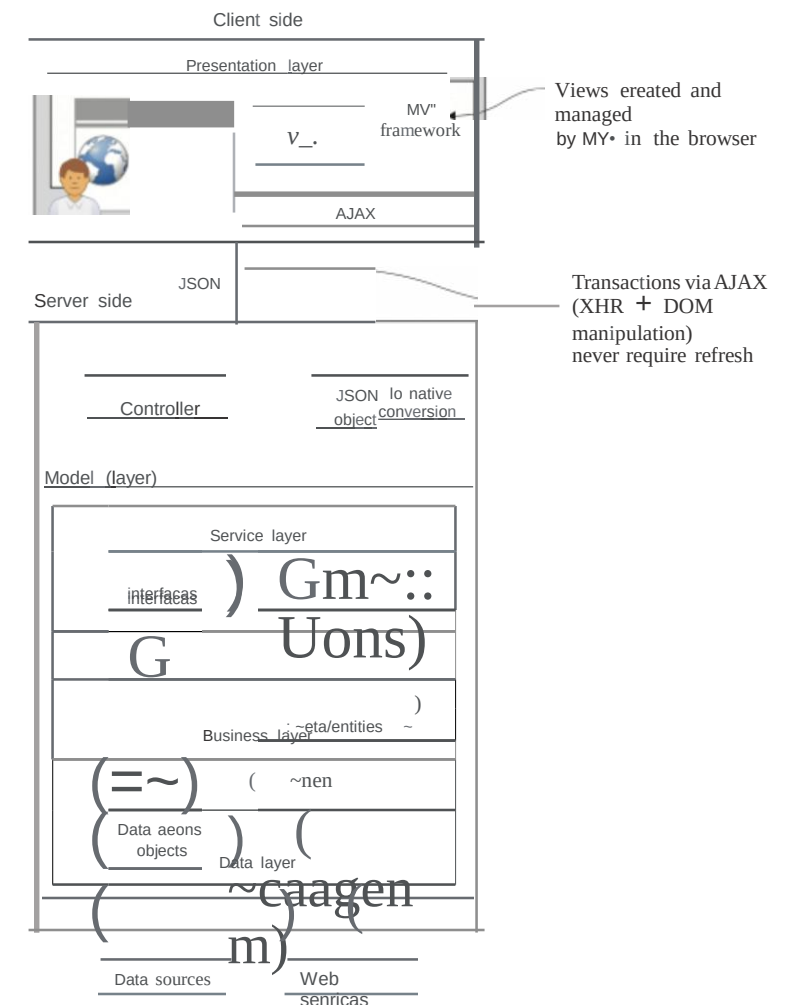
Single Page Application (SPA) merupakan sebuah website yang dapat menjalankan fungsi/komponen dalam single web page. Selain itu, SPA dapat membuat konten dalam sebuah web page, yang dimana tidak memerlukan refresh browser.



HTML views returned, causing full-page refresh

Views created and managed on the server

Traditional Web Architecture



SPA Architecture

n (SPA): H

Dalam SPA merujuk ke HTML file/shell. Cara kerja SPA hanya cukup sekali me-load single HTML file. Berdasarkan dari gambar dibawah ini, bahwa untuk container DIV dapat mempunyai subsections/subcontainer. Subsections/subcontainer dapat disebut juga sebagai regions content, yang dimana dapat di isi oleh konten yang dinamis.

```
<!DOCTYPE html>
<html>
<head>
  <title>Shell Example</title>
  <link rel="stylesheet"
        type="text/css"
        href="app/css/default.css">
</head>
<body>
  <div id="container"></div>
</body>
</html>
```

Load the application's style sheets

Initial container DIV



HTML File/Shell

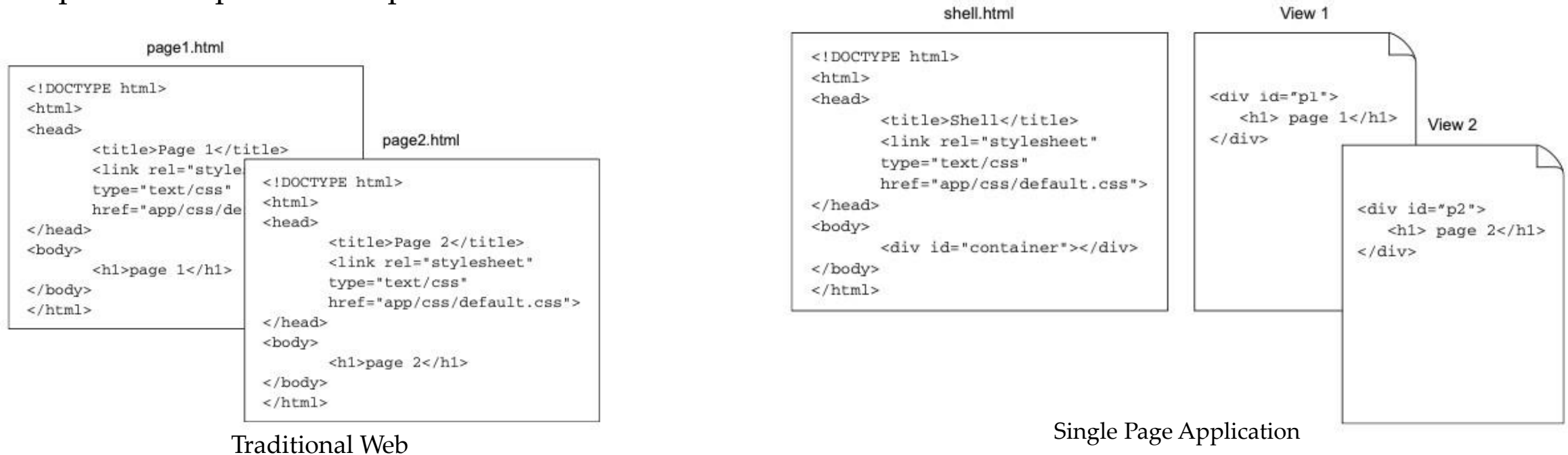


Content Region/View 1

Single Page Application (SPA): What's Different?

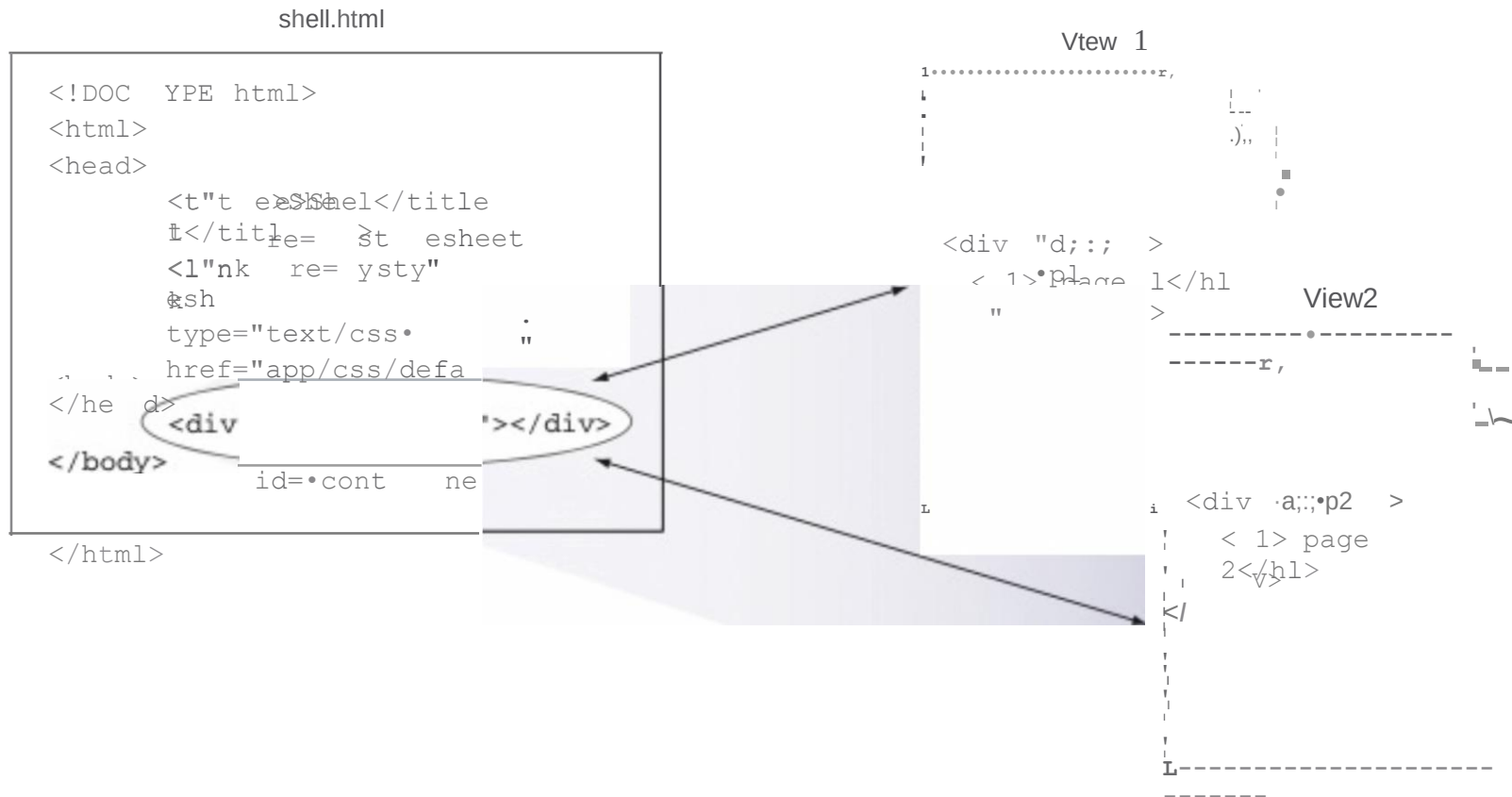
Perbedaan SPA dengan traditional web antara lain sebagai berikut:

- Dalam traditional web untuk membuat page harus terpisah.
- Struktur dari navigasi/menu dalam traditional web harus dibuat baru.
- Dalam SPA untuk struktur navigasi/menu hanya dapat dibuat sekali, dan untuk konten yang bersifat dinamis dapat dibuat pada file terpisah.



SPA: Zero

Contoh SPA dibawah ini merupakan struktur navigasi/menu yang hanya dapat dibuat sekali.

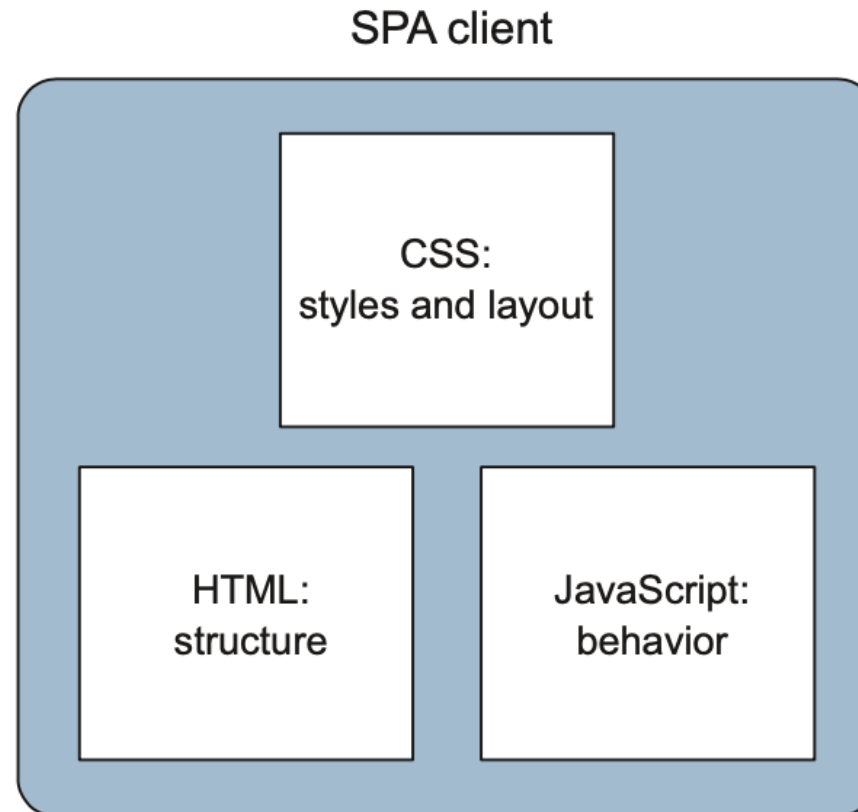


Single Page Application (SPA): Benefits of SPAs than Traditional

- **Renders like a desktop application:** tanpa harus me-refresh untuk membuka page baru.
- **Decoupled presentation layer:** proses update konten dalam regions/view dapat dilakukan secara terpisah.
- **Faster:** proses akses menjadi lebih cepat karena data/konten yang dikirim dan terima dari langsung dari server.
- **Less user wait time:** proses render konten lebih cepat dalam sebuah web page.
- **Easier code maintenance:** proses maintenance lebih cepat karena setiap konten dibuat terpisah.

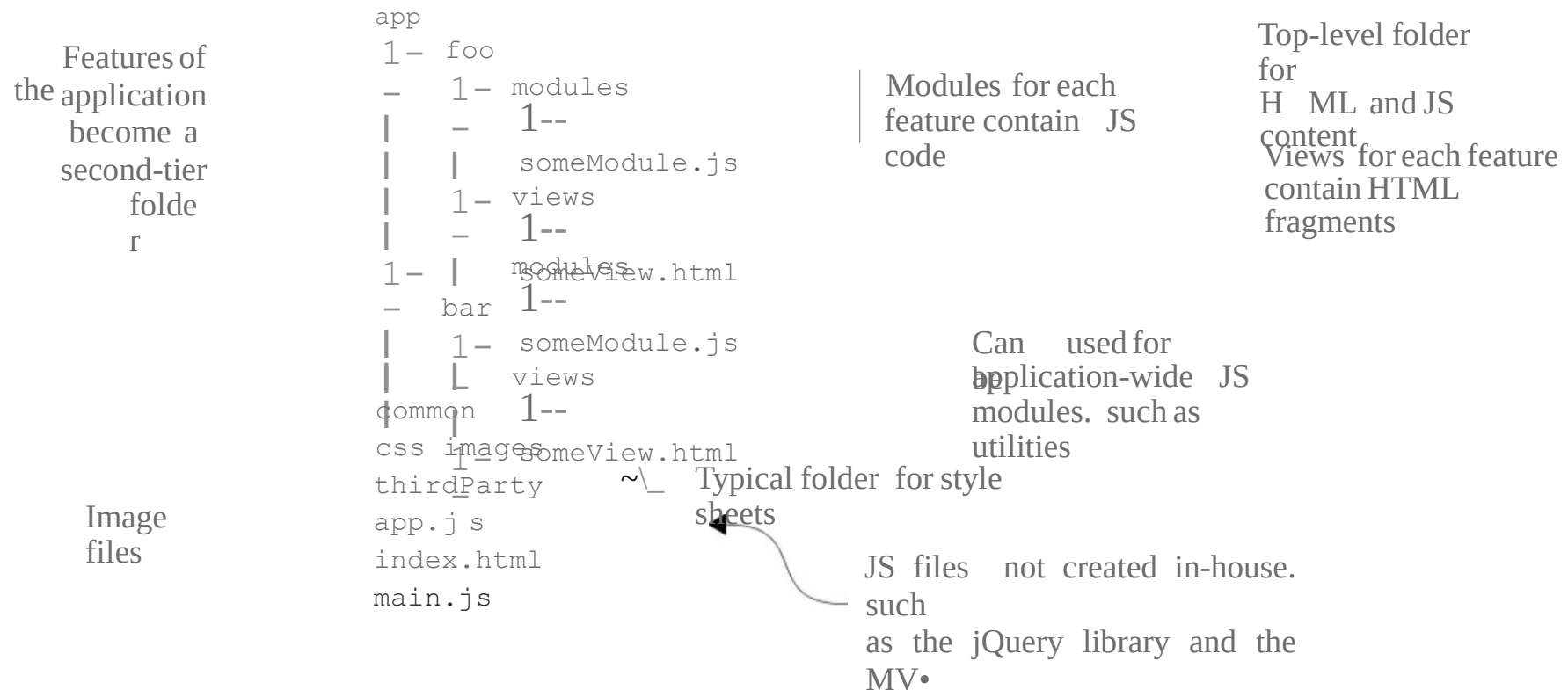
Single Page Application (SPA): Client-Side

Dalam membangun SPA teragi menjadi 3 bagian seperti styles and layout, structure, & behavior. Selain itu, tidak memerlukan browser plugin dan belajar lebih mendalam terkait dengan SPA language programming.



(SPA): Orga

Gambar dibawah ini merupakan contoh dari struktur folders & files dalam membuat SPA berdasarkan **fitur-fitur** dalam sebuah website. Setiap fitur-fitur dalam website dibuat terpisah. Berdasarkan gambar dibawah ini, terdapat 2 fitur **foo** dan **bar**.



Single Page Application (SPA): Organizing Project (Continue)

Gambar dibawah ini merupakan contoh dari struktur folders & files dalam membuat SPA berdasarkan **fitur-fitur** dalam sebuah website, dan untuk file yang berkaitan dengan konten/view dijadikan satu dalam folder fitur **foo**.



Single Page Application (SPA): Organizing Project (Continue)

Gambar dibawah ini merupakan contoh dari struktur folders & files dalam membuat SPA berdasarkan **functional** dalam sebuah website, dan untuk modules berkaitan dengan js scripts, sedangkan untuk views terkait dengan tampilan/konten.

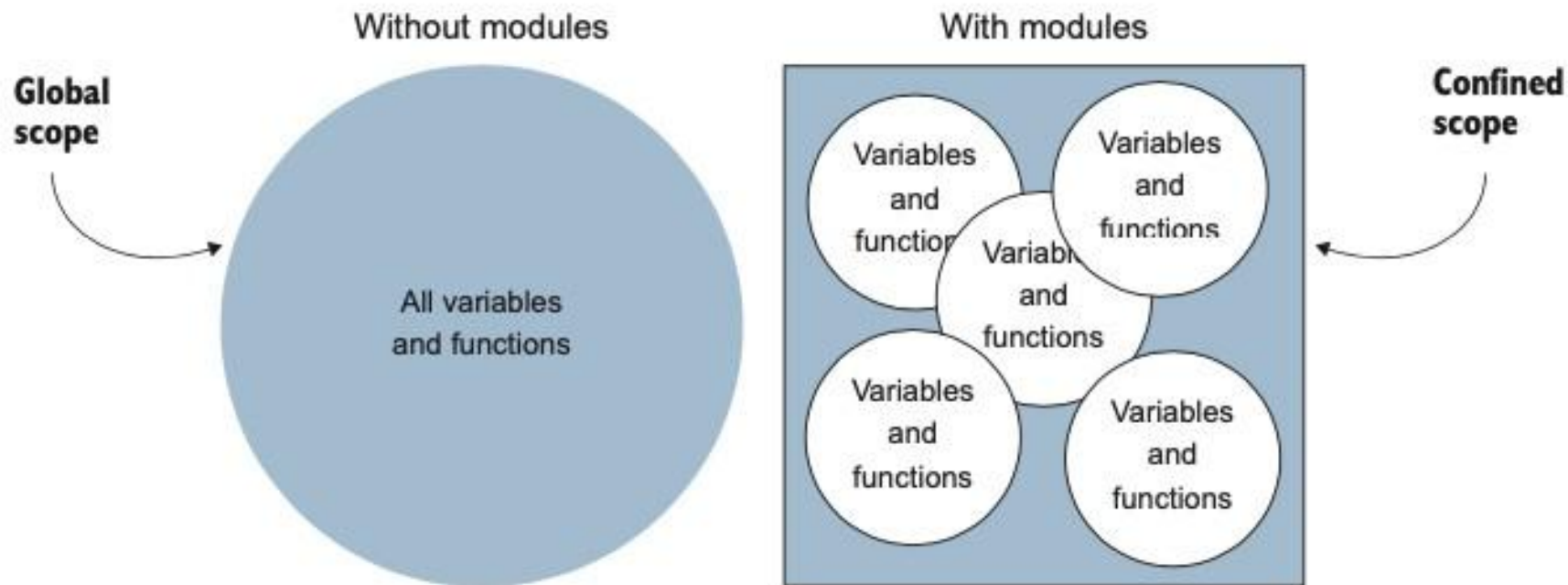


Modules

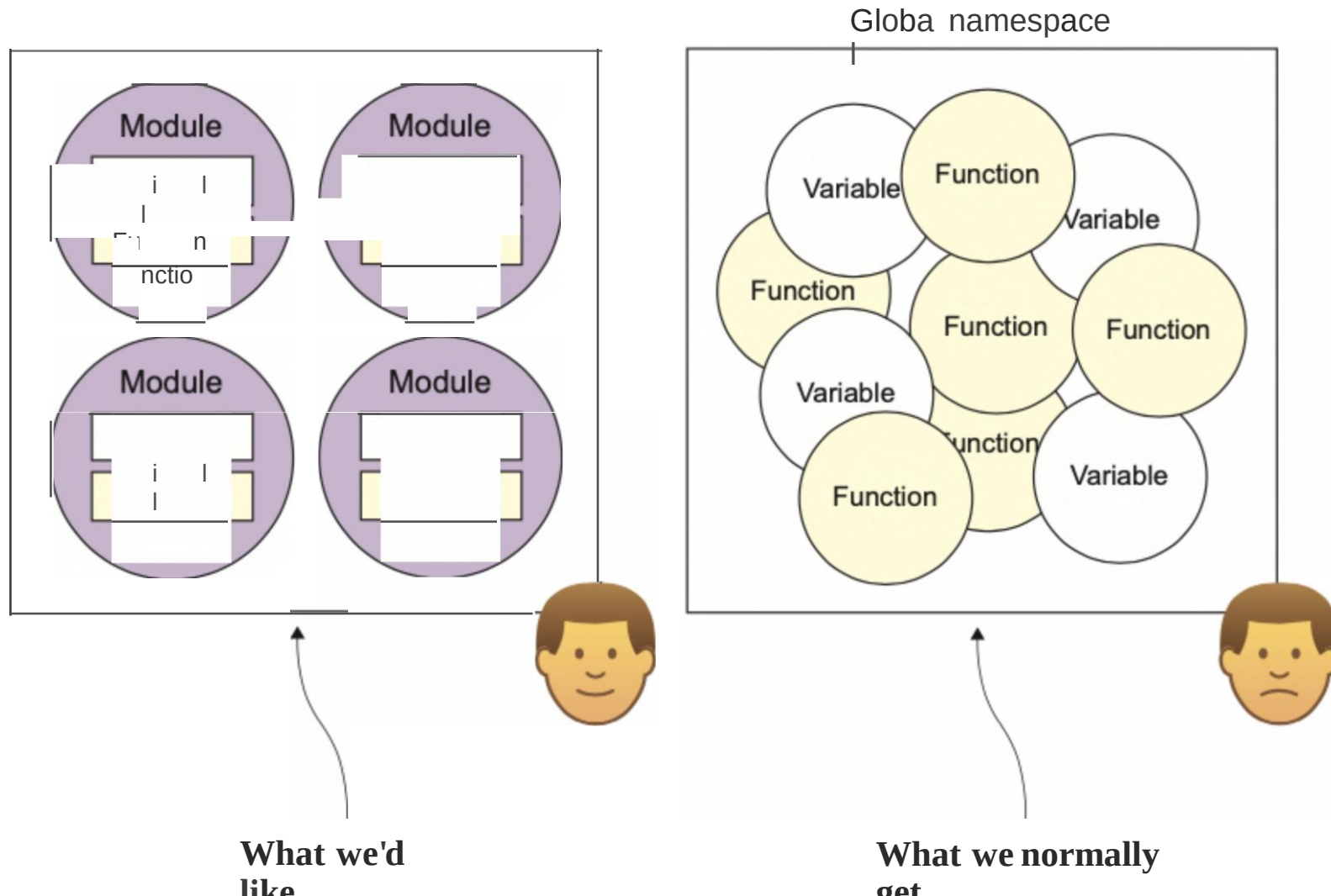


Modules: Definition

Modules merupakan bagian dari component/larger structure. Modules tidak dapat dikatakan semacam fitur dalam web application, tetapi modules dapat dikatakan sebagai fungsi.



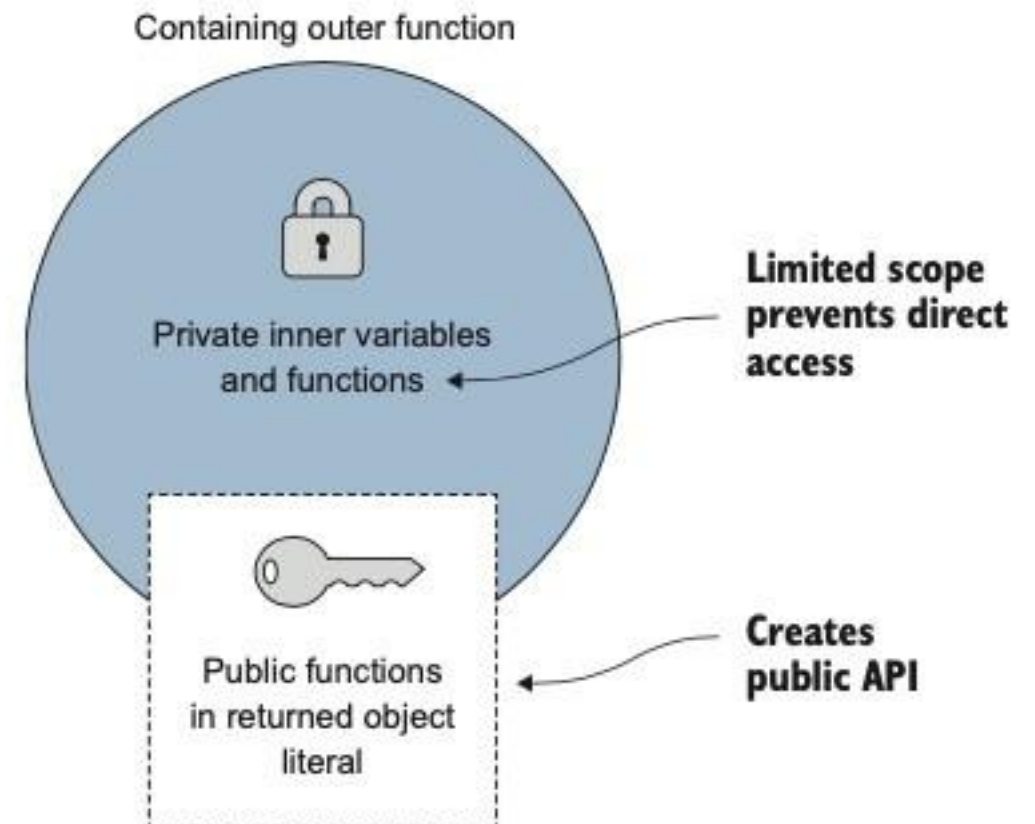
Continue)



Modules: The Structure

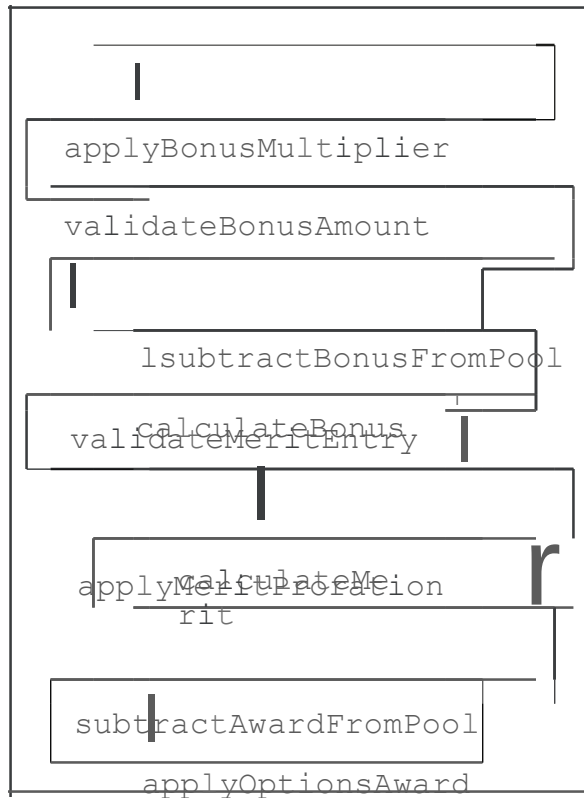
Modules dapat juga merangkum bagian logic/functions menjadi satu. Berdasarkan gambar tersebut tidak semua function bisa diakses (limited scope).

Akses diatur melalui fungsi dalam literal objek yang dikembalikannya.

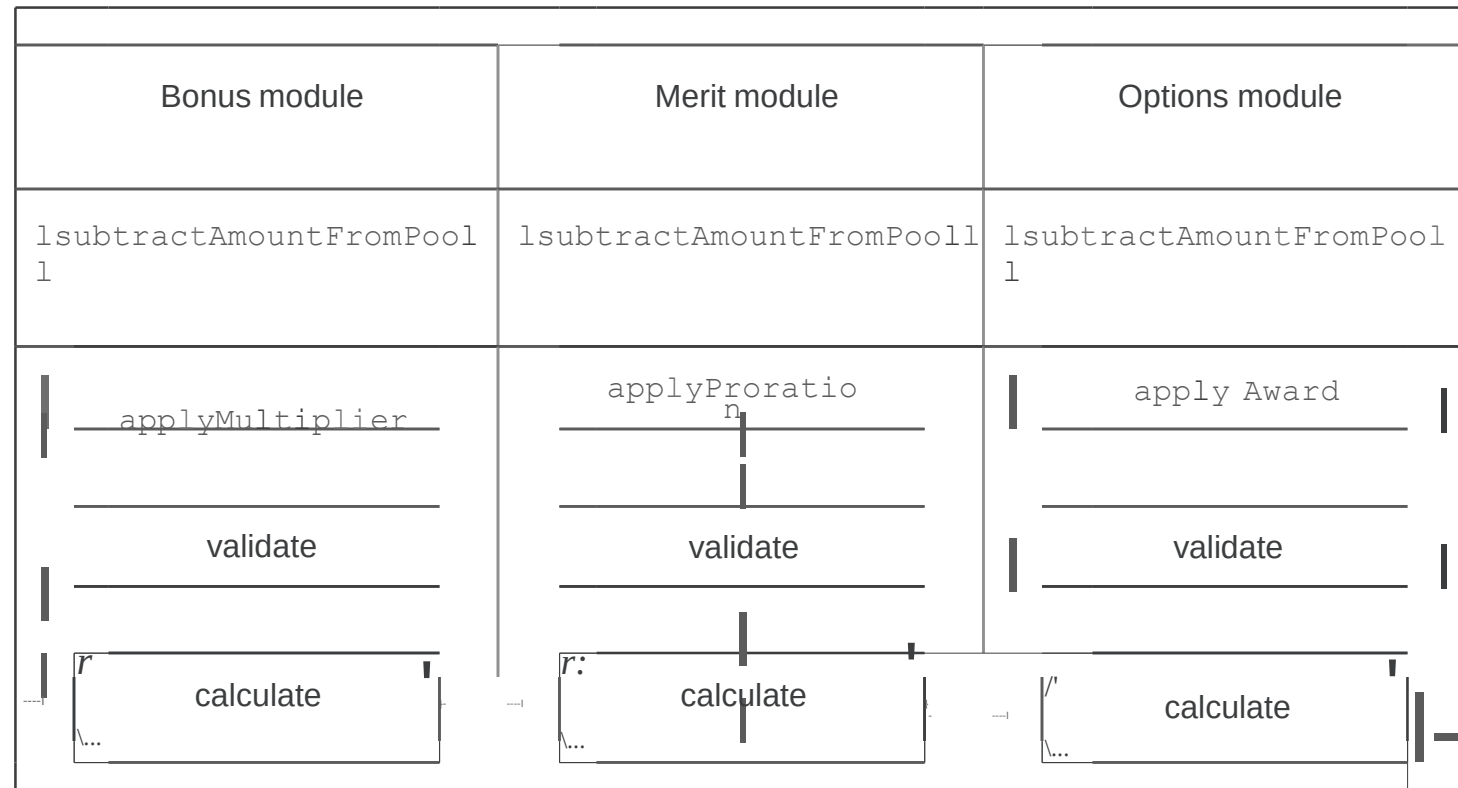


Modules dapat juga merangkum bagian logic/functions menjadi satu. Berdasarkan gambar tersebut tidak semua function bisa diakses (limited scope).

Global functions



Modules



**Private
variable
initialization
contains**

```
var numberModule = (function()  
    var    = {  
        num1    2;  
        function addNumbersinternally(num2)  
            {  
            }  
            return num1 + num2;  
        }  
        return {  
            addTwoNumbers : function(num2) {  
                alert(addNumbersinternally(num2));  
            }  
        }  
    }  
);  
numberModule.addTwoNumbers(2);
```

**Private function
addNumbersl
nternally()
does the work**

**Alerts the
result from the
call to
addTwoNumbersIntema
ly()**

**Call the module's
public
functions like this**

Submodules

Berdasarkan gambar disamping menunjukkan contoh dari submodules



Module Scope

```
STGourmet.customer = (function() {
  var customerLoggedIn = true;
  var customerName = "Emmit";
  function isNewCustomer() {
    return customerLoggedIn;
  }

  function getName() {
    return customerName;
  }

  function isLoggedin() {
    return isNewCustomer();
  }

  return {
    isLoggedin: isLoggedin,
    getName: getName,
    setName: setName
  };
})();
```

Adding customer to STGourmet creates a new submodule

Code not in the return statement is private and can't be accessed outside the module

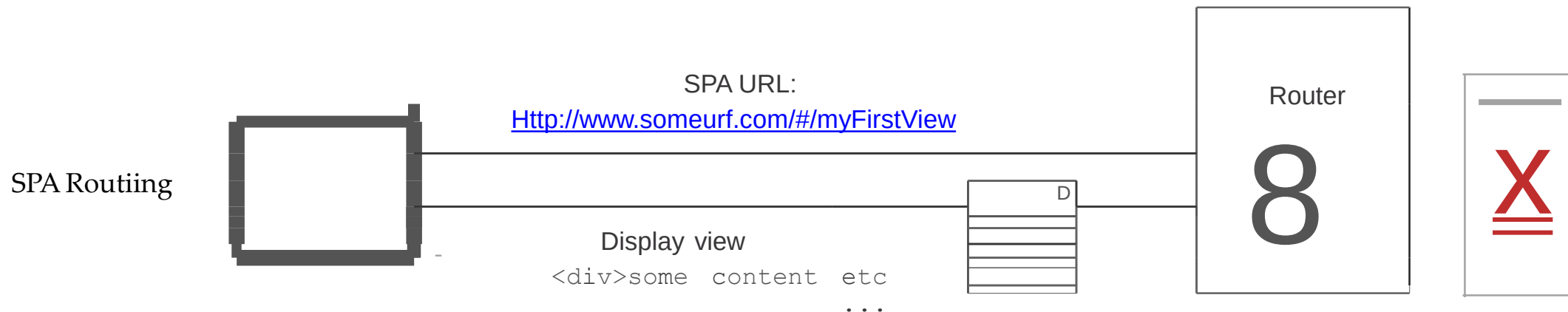
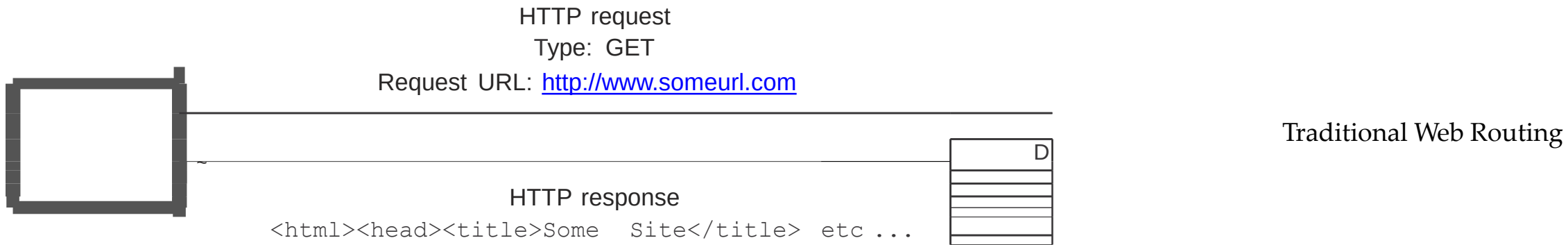
Anything declared in the return statement is public

Client-Side Routing



What's Diff

Gambar dibawah ini merupakan dari traditional web routing.



Client-Side Routing: Route Details

1. User requests the office hours view for faculty ID "rmanderson"

http:1/someuniv.edu/#/officehrsJm1a n d e r s

2. Router attempts to match the URL with one of the routes from the router configuration

Router

```
getOfficeHours(){
}
```

3. Run associated code for matched route or for default route if no match is found

4. Display view for Dr. Mary Anderson

View

Routes

Path list

/faculty

~ /contact

View:

r

Functionality:

getOfficeHours()

t

/officeHrs.html

Office hours route

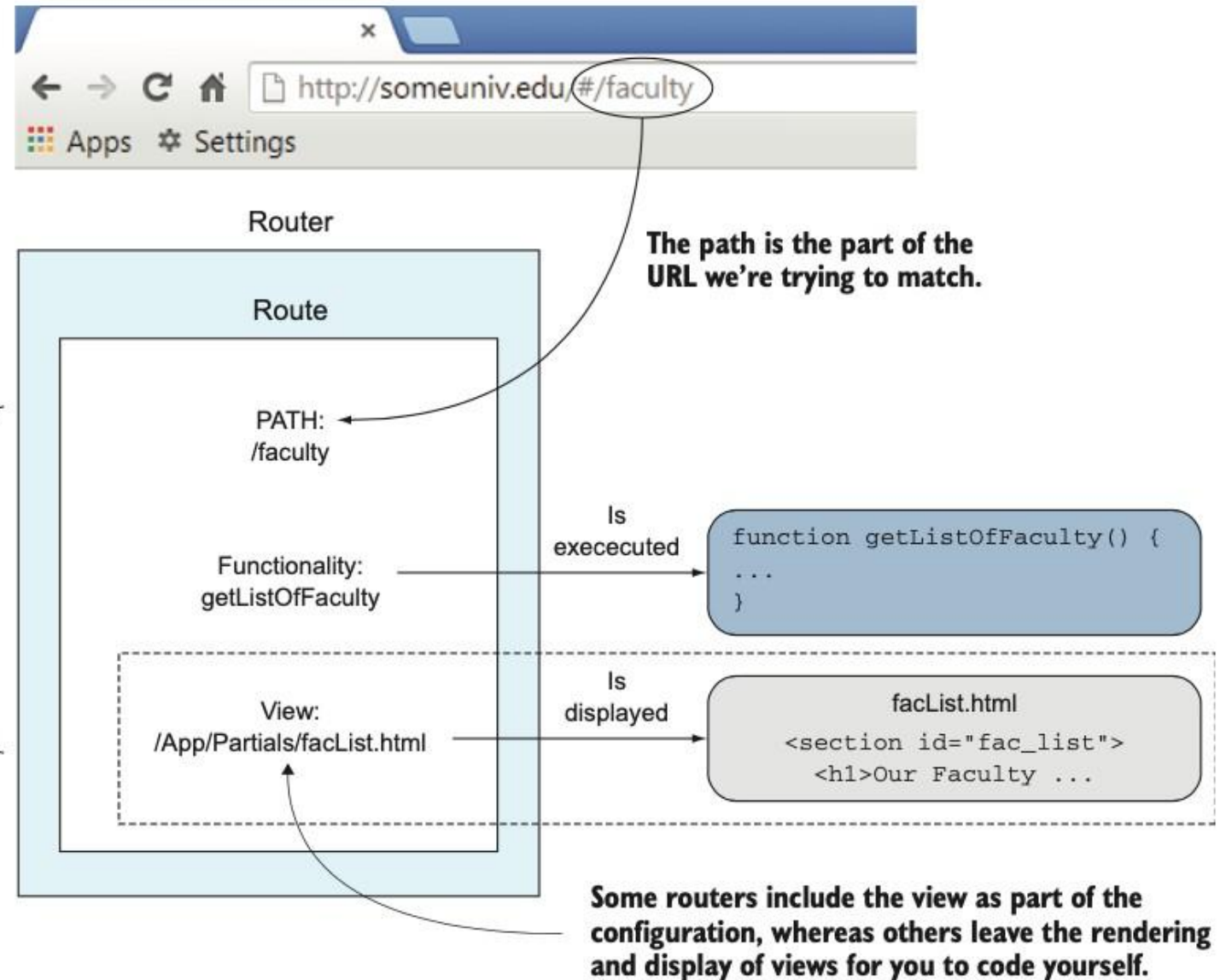
When there is no match

Routes can contain parameters (marked by special characters that vary by vendor).

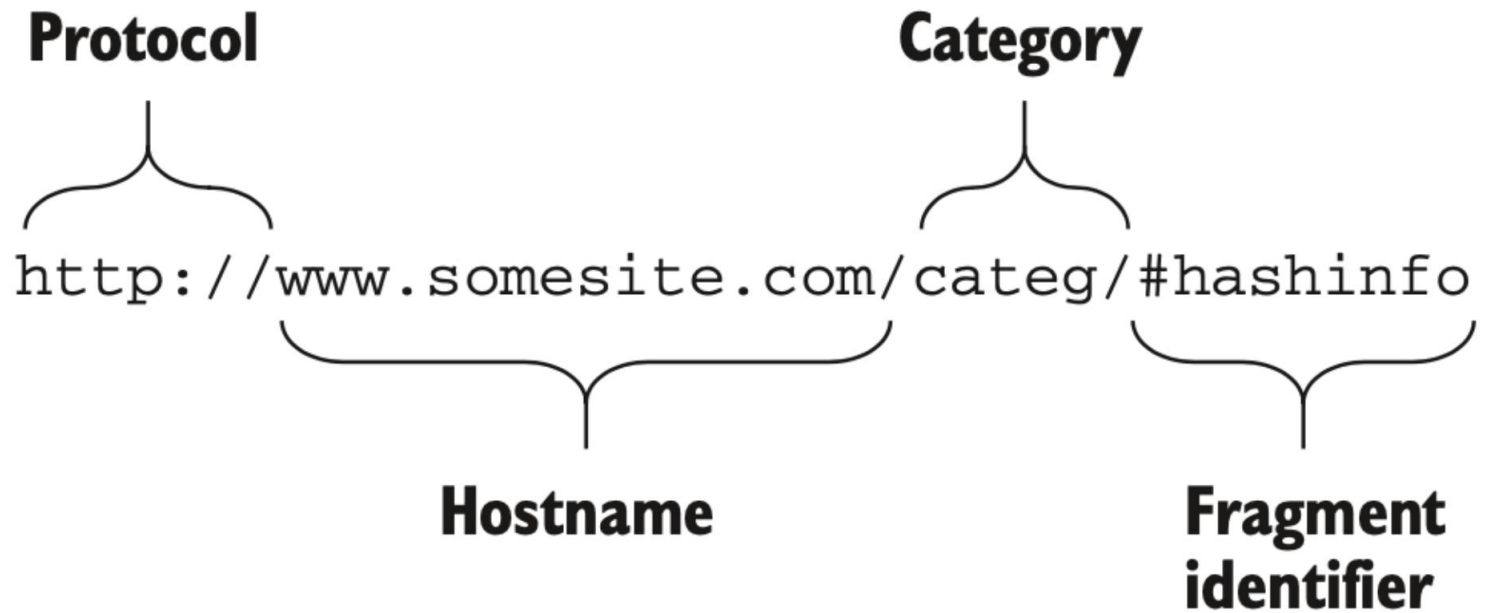
This example results in a view change, but routing doesn't always result in an update to the UI.

Client-Side Routing: Example Route Details

Each entry
in a router's
configuration
is called a
route.



Client-Side
Routing:
Identifier
Method



View Composition & Layout



```
... = modifier_ob.  
mirror object to mirror.  
mirror_mod.mirror_object
```

```
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

--- OPERATOR CLASSES ---

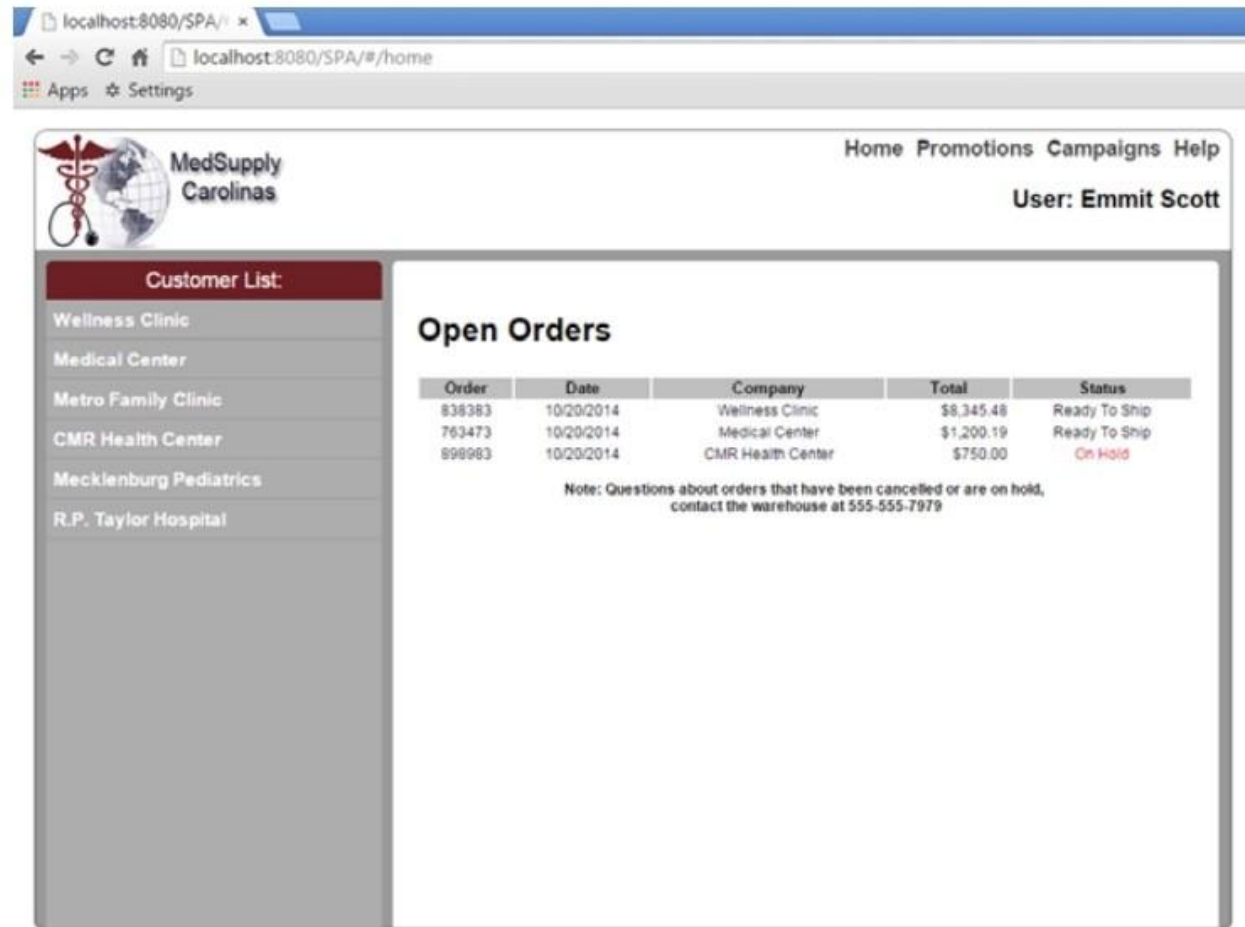
```
types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

```
context):  
context.active_object is not
```

View Composition & Layout

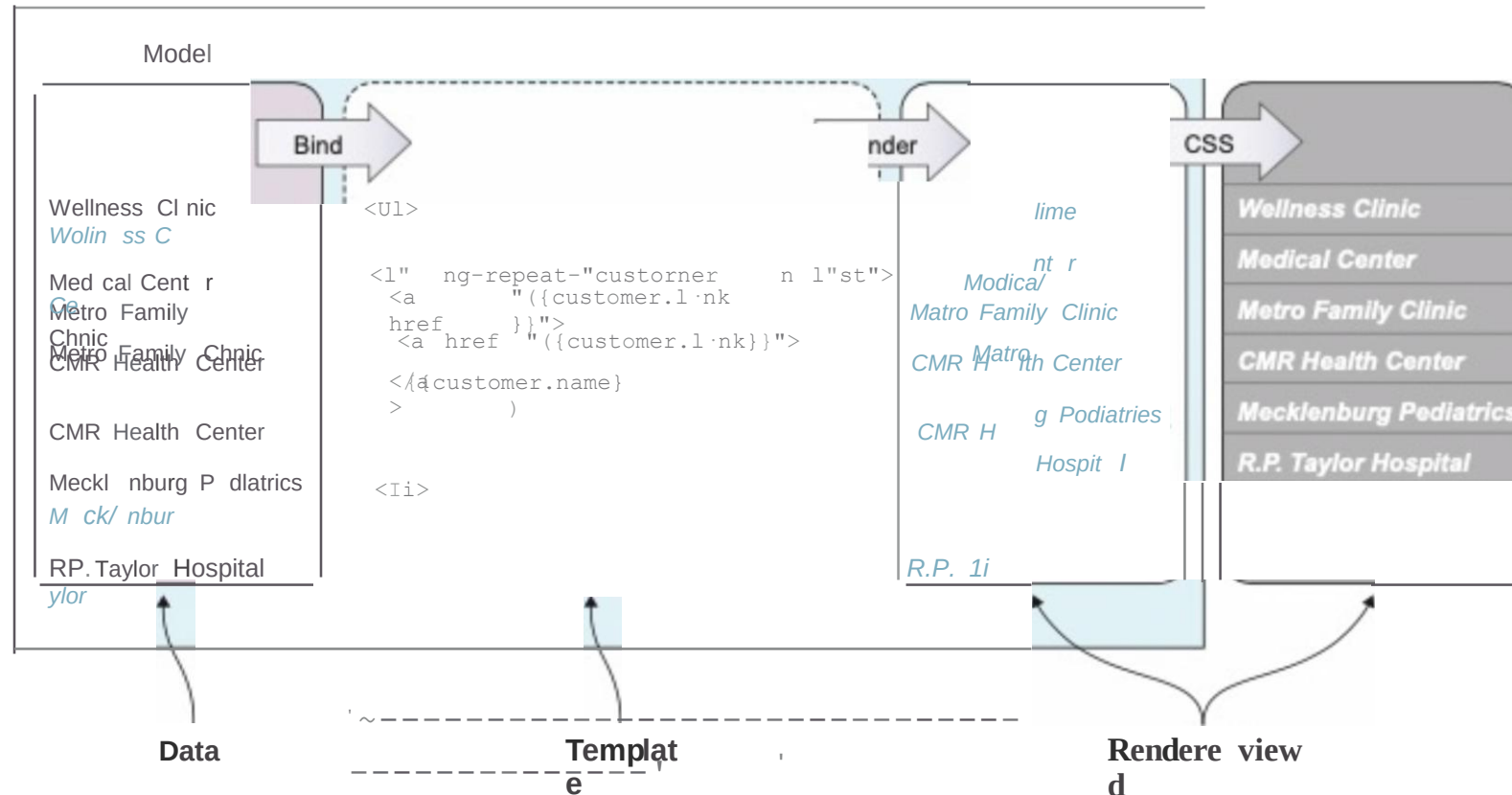
Layout design dalam SPA terdiri dari 5 bagian, antara lain sebagai berikut:

- Views
- Regions
- View Composition
- Nested Views
- Routes



Layout: Vi

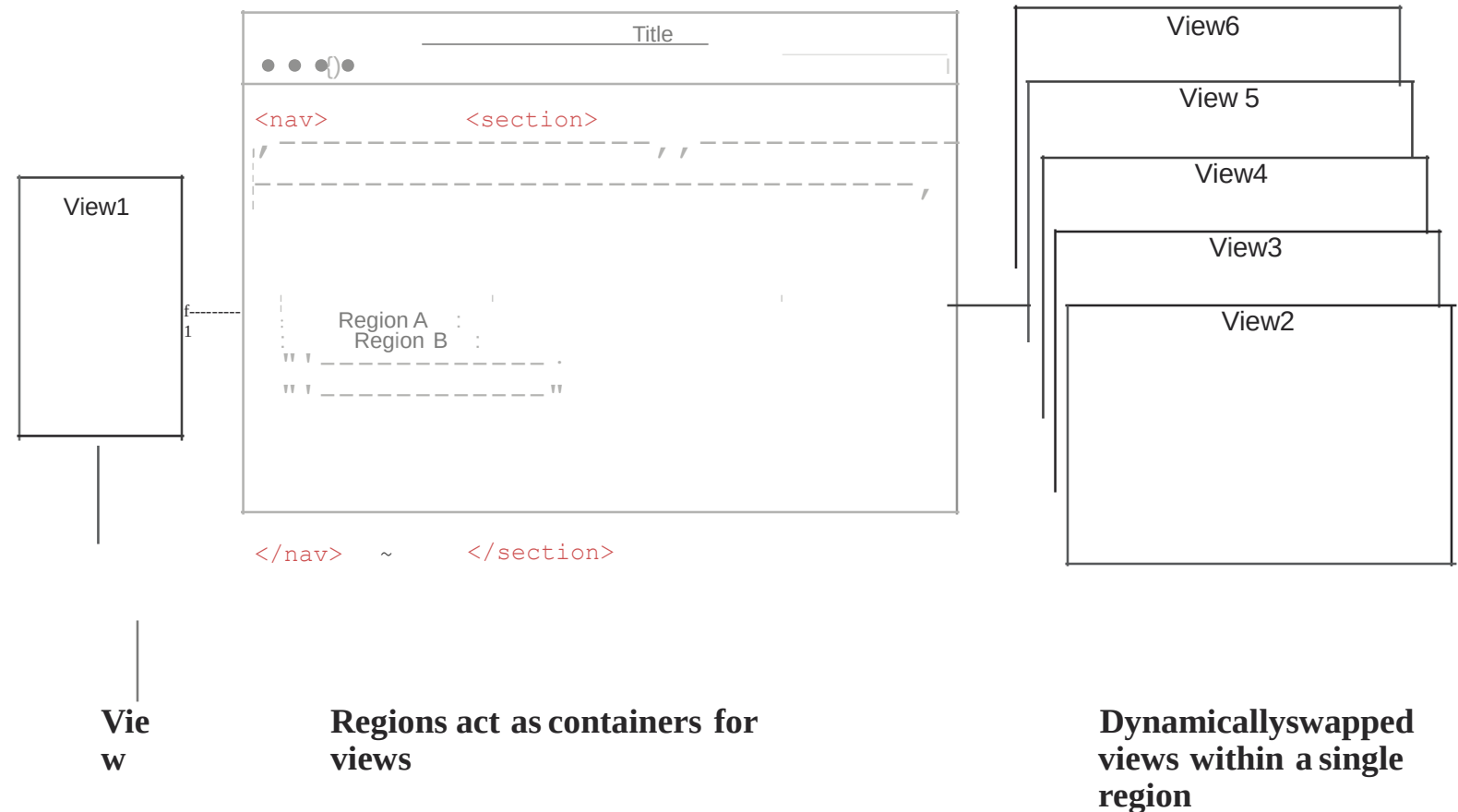
View dalam SPA merupakan bagian dari tampilan web. Gambar dibawah ini merupakan struktur dari sebuah web yang menerapkan konsep SPA.



Layout: Re

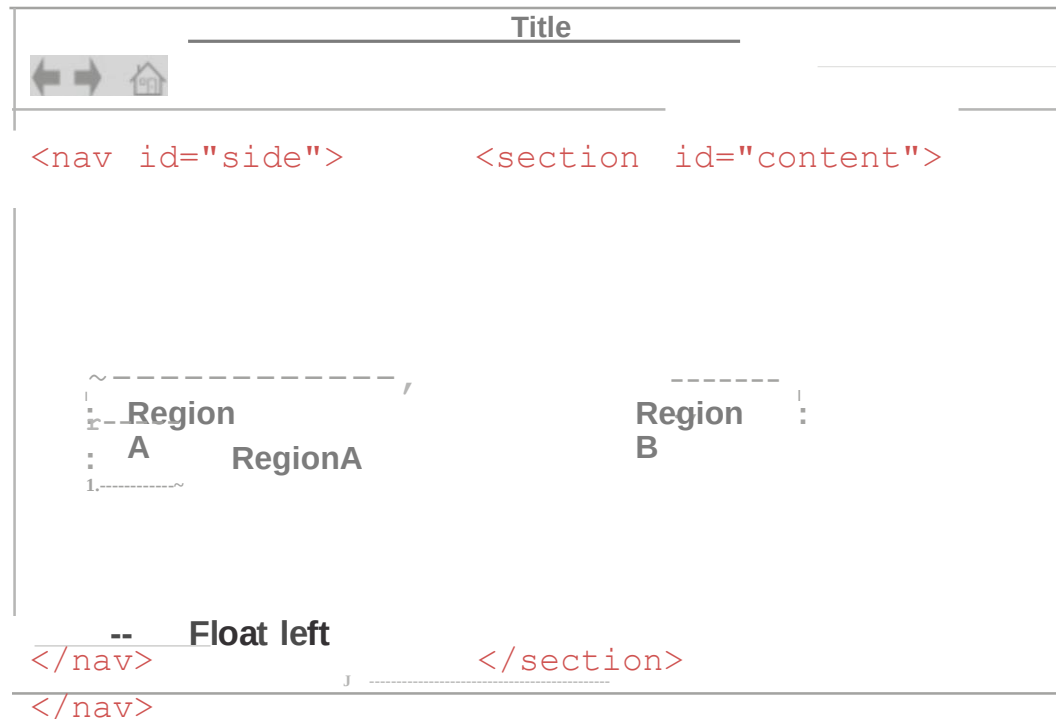
Regions dalam SPA merupakan bagian dari semantic elements jika menggunakan HTML5 atau sebuah element seperti **DIV** yang berguna sebagai containers untuk views.

Nama regions dapat dijadikan sebagai physical attributes dalam sebuah layout, dan isi dari sebuah regions adalah CSS script.



Output: Region

Gambar dibawah ini merupakan contoh bahwa regions sebagai physical attributes/kerangka dari CSS script yang ada.

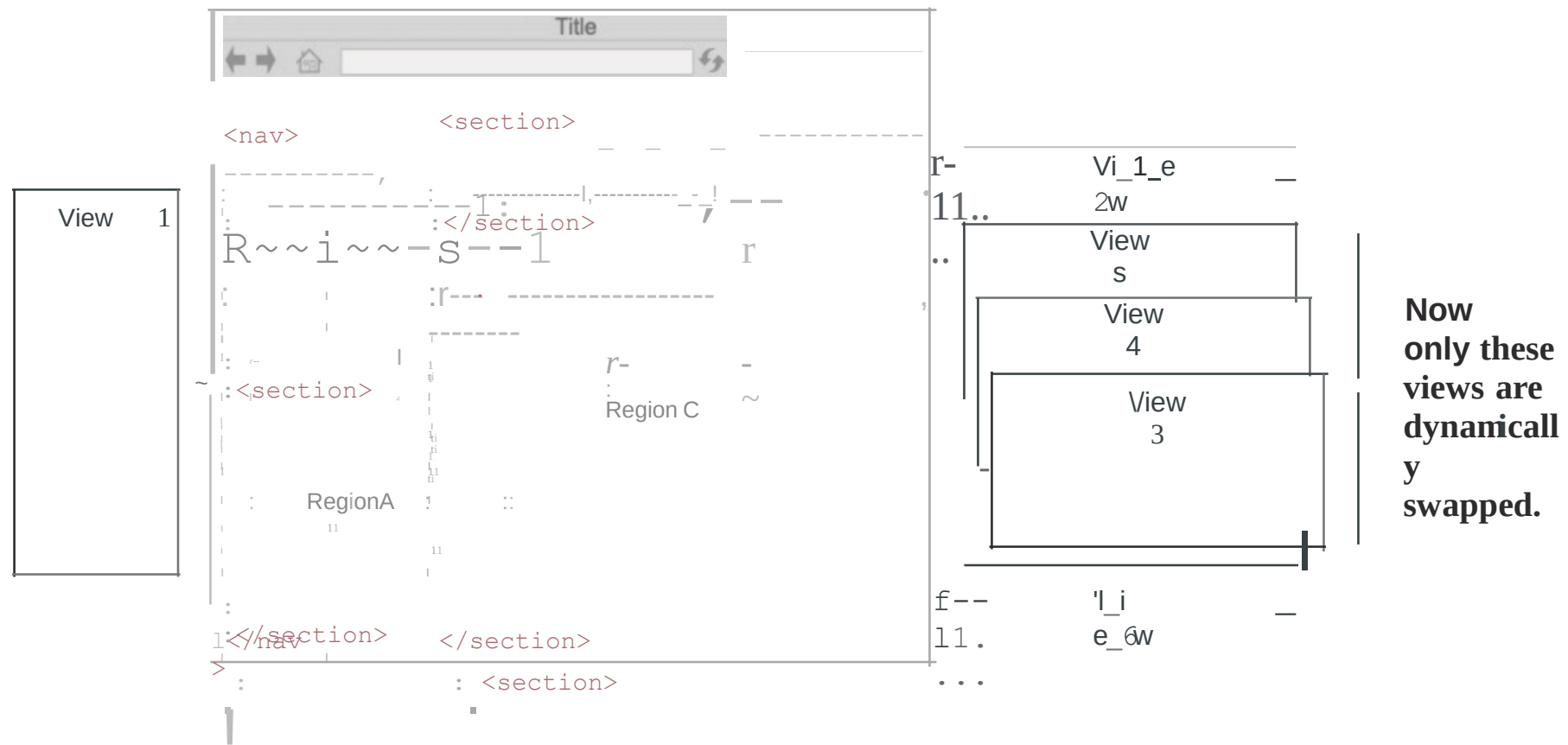


```
nav#side {
  background-color: #ADADAD;
  float: left;
  width: 28%;
  margin: 1%;
}
```

```
#content {
margin: 1%;
padding: 2%;
height 100%;
overflow-      auto;
y:              top;
text-align:
background-color: #FFFFFF;
border-              Spx;
radius:
} color: black;
```

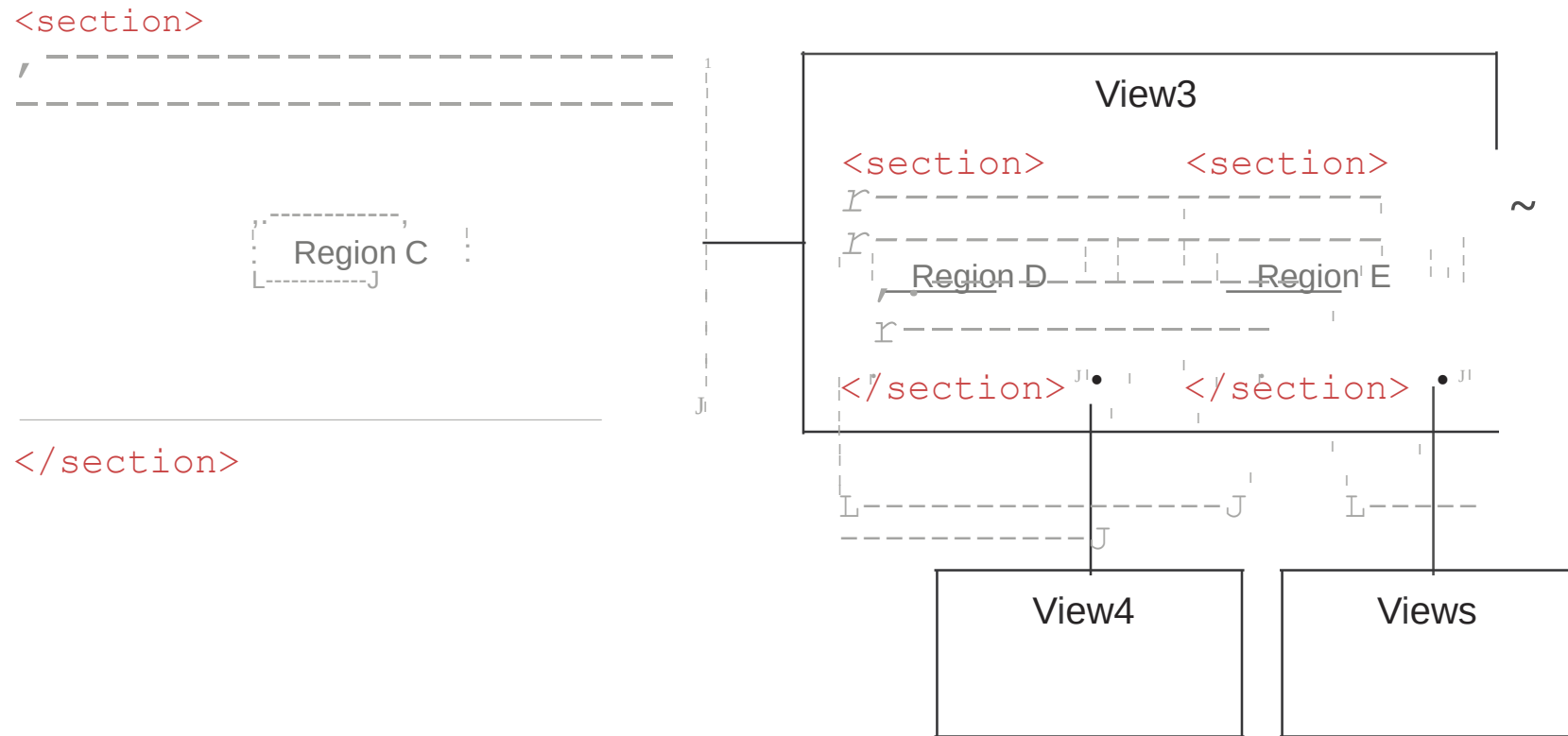
Layout: Vi

View composition dalam SPA merupakan bagian dari bounding boxes/kotak pembatas dalam layout.



Layout: Ne

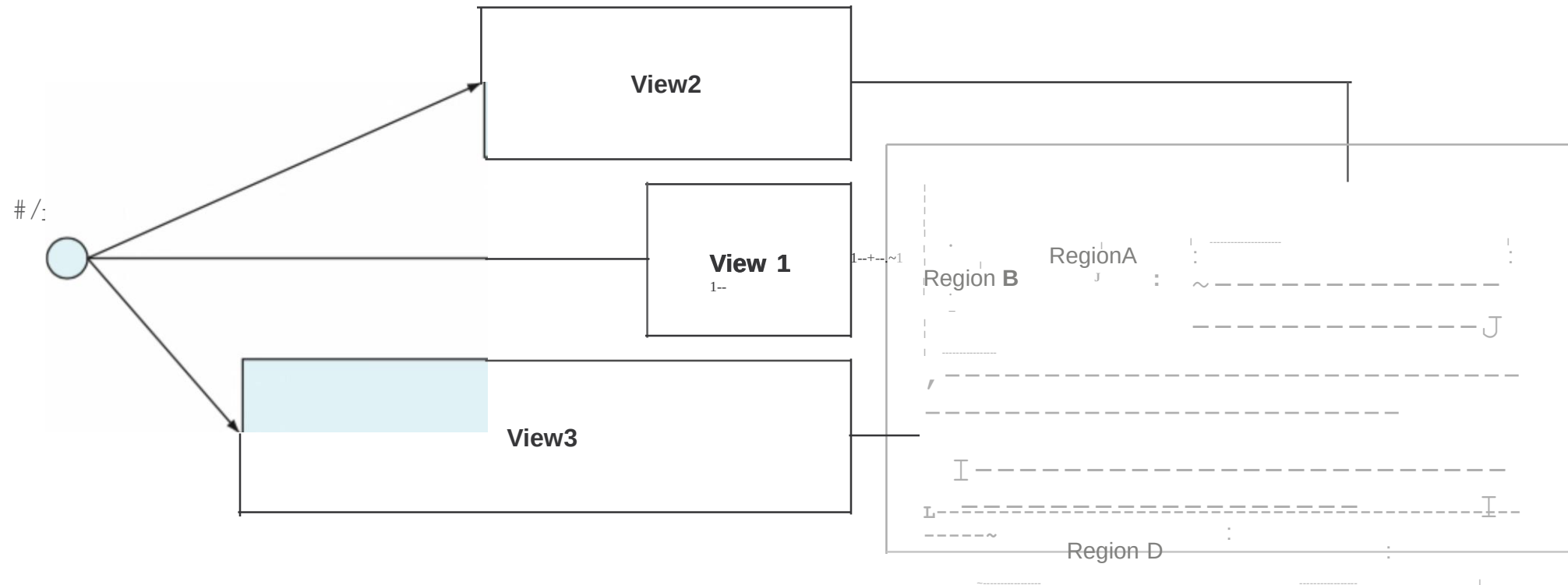
Nested views merupakan bagian turunan dari regions.



**Regions
within
views create
nested
views.**

Layout: Ro

Routes dalam SPA merupakan cara untuk mengatur rute/alur aplikasi antara views dengan setiap regions yang ada dalam sebuah layout.

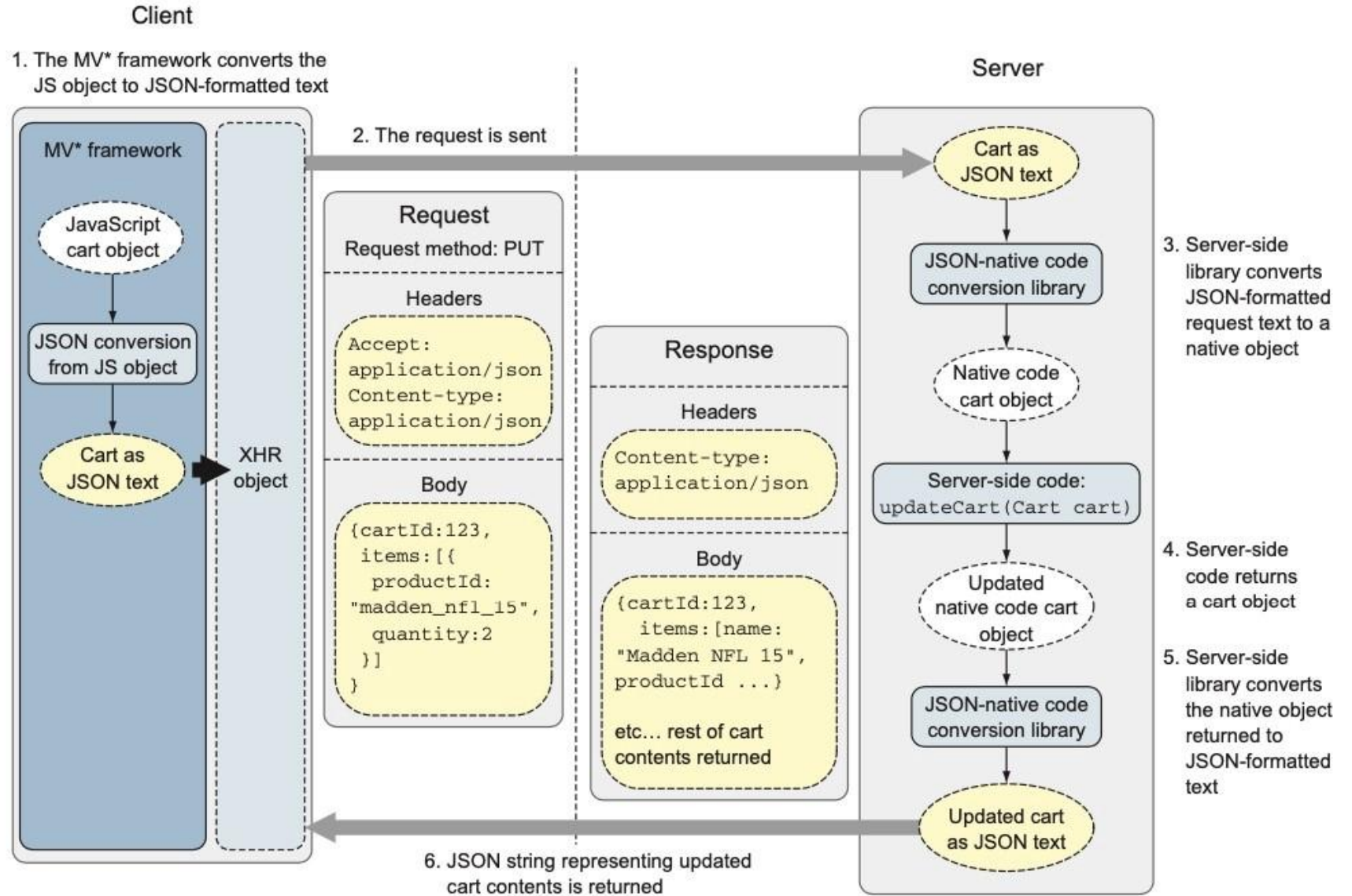


Method in SPA



Method	Descrpt on	Example	Safe?	Idempotent?
GET	Typically, GET is used to fetch data. This method is most commonly used for creating a resource or adding an item to a resource. Typically, PUT is used like an update-or-create action, updating the existing resource or optionally adding it if it doesn't exist. This is used to remove a resource.	View the shopping cart	Yes	Yes
POST		Add an item to the cart	No	No
PUT		Update the quantity of an item in the cart	No	Yes
DELETE		Remove an item from the cart	No	Yes

Method in SPA (Continue)

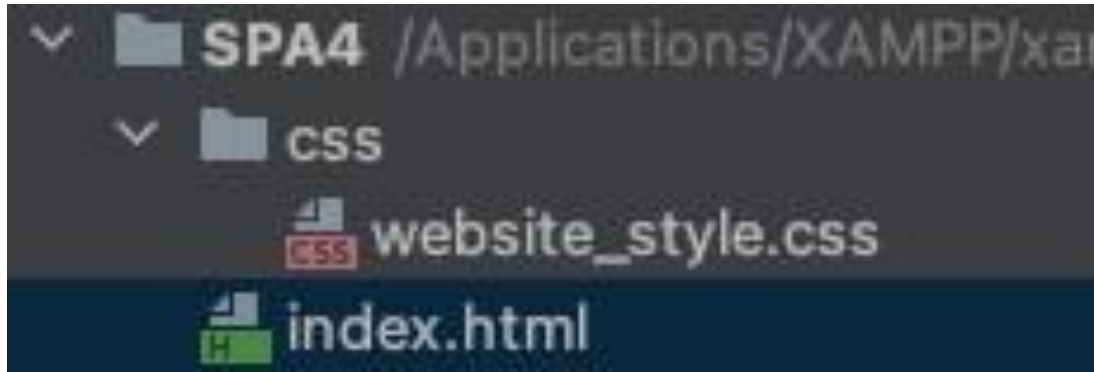


Examples



Examples: How to Create SPA?

- Buatlah struktur folder dan file seperti gambar dibawah ini:



- Setelah itu, buatlah file **index.html** dan file css dengan nama **website_style.css**

Gambar
script
index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <link rel="stylesheet" type="text/css" href="css/website_style.css">
6   <script src="https://code.jquery.com/jquery-3.7.0.js" integrity="sha256-JLqSTELeR4TLqP00G9dxM7yDPqX1ox/HfqiSLBj8+kM=" crossorigin="
7   <title>Example Single Page Application</title>
8
9   <script type="text/javascript">
10     no usages
11     function scrollTo(div)
12     {
13       $('html, body').animate(
14         {
15           scrollTop: $("#"+div).offset().top
16         }, 'slow');
17     }
18   </script>
19 </head>
20 <body>
21   <div id="header">
22     <li id="site_name">Example One Page Website</li>
23     <li onclick="scrollTo('home');">HOME</li>
24     <li onclick="scrollTo('about');">ABOUT</li>
25     <li onclick="scrollTo('contact');">CONTACT</li>
26   </div>
27
28   <div id="home">
29     <h1>Demo Of One Page Website Template Using HTML, CSS And jQuery<p>UHAMKA University</p></h1>
30     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer luctus quam quis nibh fringilla sit amet consectetur lectus
31
html > body > div#about > p
```



```

20 <div id="header">
21   <li id="site_name">Example One Page Website</li>
22   <li onclick="scrollto('home');">HOME</li>
23   <li onclick="scrollto('about');">ABOUT</li>
24   <li onclick="scrollto('contact');">CONTACT</li>
25 </div>
26
27 <div id="home">
28   <h1>Demo Of One Page Website Template Using HTML, CSS And jQuery<p>UHAMKA University</p></h1>
29   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer luctus quam quis nibh fringilla sit amet consectetur lectus ma
30 </div>
31
32 <div id="about">
33   <p>In this tutorial we will show you how to create One Page Website Template Using HTML, CSS And jQuery. Now you can build simple
34   <p>Web Programming Club UHAMKA is a complete destination to learn Web Development with Best, Fast and Easy Methods for those who
35   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer luctus quam quis nibh fringilla sit amet consectetur lectus ma
36   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer luctus quam quis nibh fringilla sit amet consectetur lectus ma
37 </div>
38
39 <div id="contact">
40   <h1>If you have any question or query you can contact us at <br>uhamka@ac.id <br><br>or you can fill the form below</h1>
41   <input type="text" placeholder="Name">
42   <textarea placeholder="Message"></textarea>
43   <input type="submit" value="SEND">
44 </div>
45 </body>
46 </html>

```

Gambar lanjutan script index.html

```
1  body
2  {
3      margin:0px auto;
4      padding:0px;
5      text-align:center;
6      width:100%;
7      font-family: "Myriad Pro","Helvetica Neue",Helvetica,Arial,Sans-Serif;
8      background-color:#A9D0F5;
9  }
10 #header
11 {
12     width:100%;
13     margin:0px auto;
14     padding:0px;
15     text-align:center;
16     height:70px;
17     line-height:70px;
18 }
19 #header li
20 {
21     display:inline-block;
22     margin-right:25px;
23     font-size:17px;
24     color:#084B8A;
25     font-weight:bold;
26     cursor:pointer;
27 }
```

Gambar script website_style.css

```
28     #header #site_name
29     {
30         text-align:left;
31         width:680px;
32         font-size:35px;
33         margin-left:20px;
34     }
35     #home,#about,#contact
36     {
37         margin-top:100px;
38         width:90%;
39         height:400px;
40         margin-left:5%;
41         padding:50px;
42         box-sizing:border-box;
43         background-color:#084B8A;
44         border-radius:10px;
45         color:white;
46     }
47     #home h1
48     {
49         font-size:40px;
50     }
51     #home p
52     {
53         font-size:20px;
54     }
55     #about p
56     {
57         font-size:20px;
58     }
```

Gambar lanjutan script website_style.css

```
59     #contact input[type="text"]
60     {
61         width:250px;
62         height:35px;
63         padding-left:10px;
64         float:left;
65         margin-left:80px;
66         border:none;
67     }
68     #contact textarea
69     {
70         float:left;
71         width:250px;
72         height:35px;
73         margin-left:20px;
74         border:none;
75         padding-left:10px;
76         padding-top:10px;
77     }
78     #contact input[type="submit"] {
79         float: left;
80         margin-left: 20px;
81         height: 35px;
82         width: 100px;
83         background: none;
84         color: white;
85         border: 1px solid white;
86     }
87
```

Gambar lanjutan script website_style.css

References

1. Mikowski, M. S., Powell, J. C., & Benson, G. D. (2014). Single Page Web Application - JavaScript end-to-end (K. Miller, J. III, J. Ryan, J. Vail, B. Berg, T. Mulligan, G. Salinovic, & M. Tudor (eds.)). Manning Publications Co. www.it-ebooks.info
2. Scott, E. (2016). SPA Design and Architecture (D. Maharry, J. Kotarski, A. Gibson, J.-F. Morin, S. Wilkey, L. Recktenwald, & M. Tudor (eds.)). Manning Publications Co.
3. <https://jquery.com/>
4. <https://talkerscode.com/webtricks/one-page-website-template-using-html-css-and-jquery.php>



THANK YOU



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web

Pertemuan 11

PHP & MySql Dasar

Zuhri Halim, S.Kom., M.Kom.

Outline

1. PHP Introduction
2. Function
3. PHP & MySQL



Introduction: Learning Outcomes

Mahasiswa mampu menjelaskan fungsi-fungsi PHP dan MySQL.

Mahasiswa mampu menganalisis script PHP dan MySQL dasar.

PHP Introduction



PHP Introduction

- Singkatan dari PHP adalah **Hypertext Preprocessor**.
- Versi terbaru: PHP 8.2.6
- Menyediakan fasilitas pemrograman sekuensial & berorientasi objek di sisi server (server-side scripting).
- Kode PHP dikompilasi dan diproses oleh web server, kemudian hasilnya ditampilkan oleh browser (kode PHP-nya sendiri tidak terlihat jika source page dibuka dari browser).

PHP Introduction: Web Server

- Lingkungan kompilasi dan implementasi untuk dokumen web server-side scripting
- Contoh web server:
 - PHP: Apache HTTP Server
 - Java Server Pages (JSP): Apache Tomcat
 - Active Server Pages (ASP): IIS

PHP Introduction: How to Use?

- Buat folder “praktikum_php_mysql” di path “../xampp/htdocs/”.
- Buat file berekstensi .php atau .html berisi:

```
<?php  
    echo "Hello PHP!";  
?>
```

- Akses melalui browser dengan alamat:
 - localhost/praktikum
 - 127.0.0.1/praktikum

PHP Introduction: Comment in PHP

Sama dengan Java

- `//`
- `/* .... */`

PHP Introduction: Variables & Data Types

- Aturan penamaan
 - Case sensitive
 - Harus diawali huruf atau underscore (_)
 - Bukan keyword khusus PHP
 - Tidak mengandung karakter tanda baca atau karakter untuk operasi pada kode
- Deklarasi
 - `$namavariabel = nilai;`
- Tipe data:
 - Standar (boolean, int, float/double, string)
 - Array
 - NULL
 - Resource (pre-defined variables)

PHP Introduction: Variables & Data Types (Continue)

Berikut merupakan contoh dari tipe data:

- `$flag = true;` (boolean)
- `$i = 2;` (integer)
- `$j = 2.3;` (float / double)
- `$nama1 = "double";` (string dq)
- `$nama2 = 'single';` (string sq)

PHP Introduction: Data Array

Berikut merupakan contoh dari tipe data array:

- Inisialisasi
 - `$arr1 = array();`
 - `$arr2 = array(301,"MI");` (=> indeks numerik)
 - `$arr3 = array("kode"=>301, "nama"=>"MI");` (=> indeks asosiatif)
- Pengaksesan
 - `echo $arr2[1];`
 - `echo $arr3["kode"];`
- Assignment
 - `$arr2[1] = "KA";`
 - `$arr3["kode"] = 303;`

PHP Introduction: Data NULL

- NULL berbeda dengan 0
- Satu-satunya nilai yg terdefinisi yaitu nilai tak terdefinisi (NULL)
 - `$namavar = NULL;`

PHP Introduction: Data Resource Types

- Variabel khusus untuk informasi dan operasi PHP spesifik, biasanya berlaku global
- Contoh:
 - `$_SERVER["HTTP_HOST"];`
 - `$_SESSION["user"];`
 - `$_GET["id"];`
- Session dan Get akan dibahas nanti

PHP Introduction: Operator

- Aritmatika: + - * / % ++ --
- Assignment : = += -= *= /= %=
- Perbandingan : == != > < >= <=
- Logika: && || ! AND OR XOR
- String (konkatenasi): . .=

PHP Introduction: Sequential/Sequence

- Kode dieksekusi secara urutan, dari atas ke bawah
- Contoh:

```
<?php
    $x = 15;
    $x += 6;
    echo "hasil: ".$x;
?>
```


PHP Introduction: Conditional (if)

- Kode dieksekusi secara urutan, dari atas ke bawah
- Contoh:

```
<?php
    $x = 15;
    $x += 6;
    echo "hasil: ".$x;
?>
```

PHP Introduction: Conditional (if-else)

- Contoh:

```
<?php
    $i = -8;
    if ($i>0) {
        echo "positif";
    } else if ($i==0) {
        echo "nol";
    } else {
        echo "negatif";
    }
?>
```

PHP Introduction: Conditional (switch-case)

```
contoh_switch_case.php x
1  <?php
2  $bil=5;
3  switch ($bil) {
4      case 0 :
5          echo "nol";
6          break;
7      case 2 :
8          echo "dua";
9          break;
10     default :
11         echo "lainnya";
12     }
13  ?>
```

PHP Introduction: Looping (while)

```
1  <?php
2  $i=0;
3  while ($i<4) {
4      echo $i. "<br />";
5      $i++;
6  }
7  ?>
```

PHP Introduction: Looping (do-while)

- Contoh:

```
<?php
    $i=0;
    do {
        echo $i. "<br />";
        $i++;
    } while ($i<8);
?>
```

PHP Introduction: Looping (for)

- Contoh:

```
<?php
    for ($i=1;$i<8;$i++) {
        echo    "<h".$i.">H".$i."</h".$i.">";
    }
?>
```

PHP Introduction: Looping (foreach)

- Contoh:

```
<?php
    $arr = array("MI","TK","KA");
    foreach ($arr as $x) {
        echo  "jurusan ".$x."<br />";
    }
?>
```


Function



Function

- Sekumpulan instruksi yang dilakukan dan mungkin menghasilkan sesuatu
- Dapat dianggap modul/sub program
- Boleh mempunyai parameter masukan atau tidak, dan boleh tidak mengembalikan hasil

Function Examples (Continue)

- Contoh:

```
<?php
    function bilanghalo($str) {
        echo "halo ".$str;
    }
    function tambah($a,$b) {
        return $a+$b;
    }
    bilanghalo("halo bro! <br />");
    $x = tambah(5,6);
    echo $x."+ 3 = ".tambah($x,3);
?>
```

Function: Data Process in Form

- Metode GET
 - Data diletakkan di belakang URL
 - Cara pengambilan: `$_GET["nama_name"]`
- Metode POST
 - Data dikirim secara terpisah, tidak diletakkan di belakang URL
 - Cara pengambilan: `$_POST["nama_name"]`

Function: Use of GET & POST

- Bila data bersifat rahasia, maka gunakan POST (misalnya login account / password)
- Bila data bersifat memperjelas, gunakan GET (misalnya pencarian atau link)
- Bila data yg dikirim banyak, gunakan POST (agar URL tidak terlalu panjang dan spesifik)

Function: Use of GET & POST (Continue)

Contoh penggunaan GET:

- proses.php

```
<?php
    echo "User= ".$_GET["user"].", Pass= ".$_GET["pass"];
?>
```

- form.php

```
<form method="get" action="proses.php">
    User:<input type="text" name="user" /><br />
    Pass:<input type="password" name="pass" /><br />
    <input type="submit" value="Login" />
</form>
```

Function: Use of GET & POST (Continue)

Contoh penggunaan POST:

- proses.php

```
<?php  
    echo "User= ".$_POST["user"].",Pass= ".$_POST["pass"];  
?>
```

- form.php

```
<form method="post" action="proses.php">  
    User:<input type="text" name="user" /><br />  
    Pass:<input type="password" name="pass" /><br />  
    <input type="submit" value="Login" />  
</form>
```


PHP & MySQL



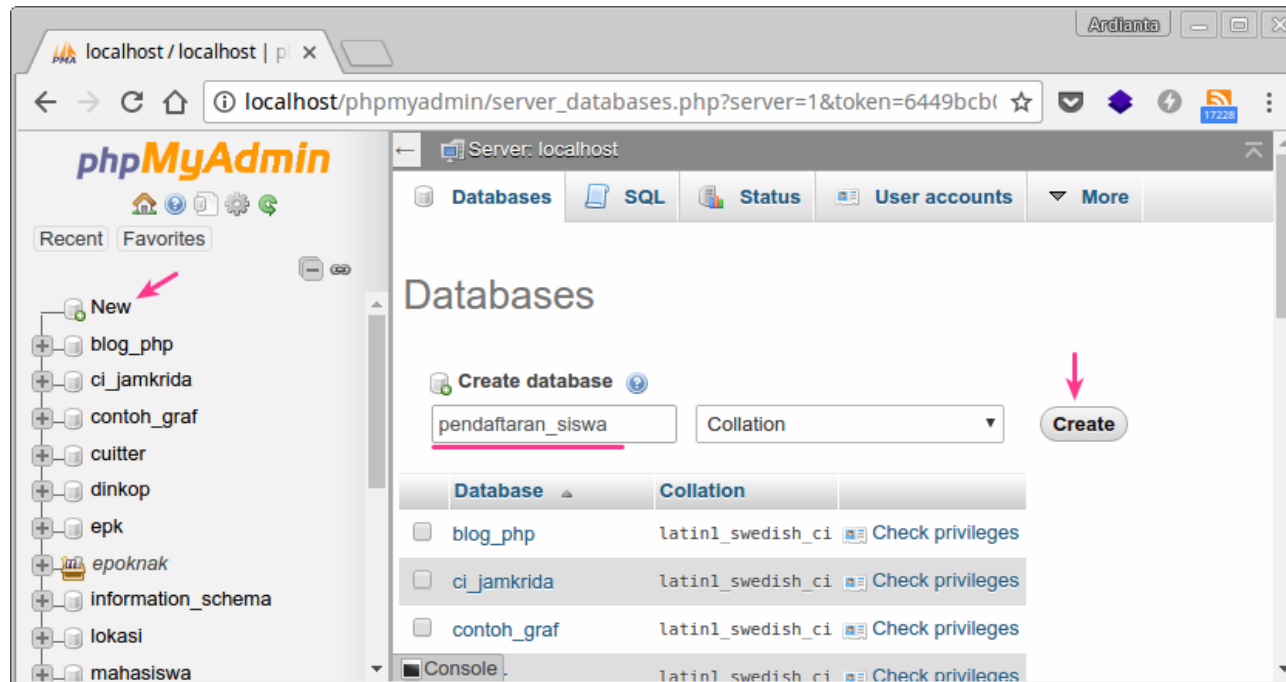
PHP & MySQL: Preparation Database

- PHP dan MySQL dapat saling berhubungan melalui perantara API (Application Programming Interface).
- Ada tiga jenis API MySQL yang disediakan di PHP:
 - mysql API ini sudah tidak didukung/dikembangkan lagi (kadalwarsa).
 - mysqli (mysql improvement) peningkatan dari API mysql
 - PDO (PHP Data Object)
- API yang pertama sudah tidak didukung lagi sejak PHP versi 5.5.x dan digantikan dengan mysqli yang lebih canggih. Karena mysqli dapat digunakan secara OOP dan prosedural.
- Sedangkan API PDO digunakan untuk menghubungkan PHP dengan vendor database yang lainnya. Jadi tidak hanya untuk MySQL saja.

PHP & MySQL: Preparation Database (Continue)

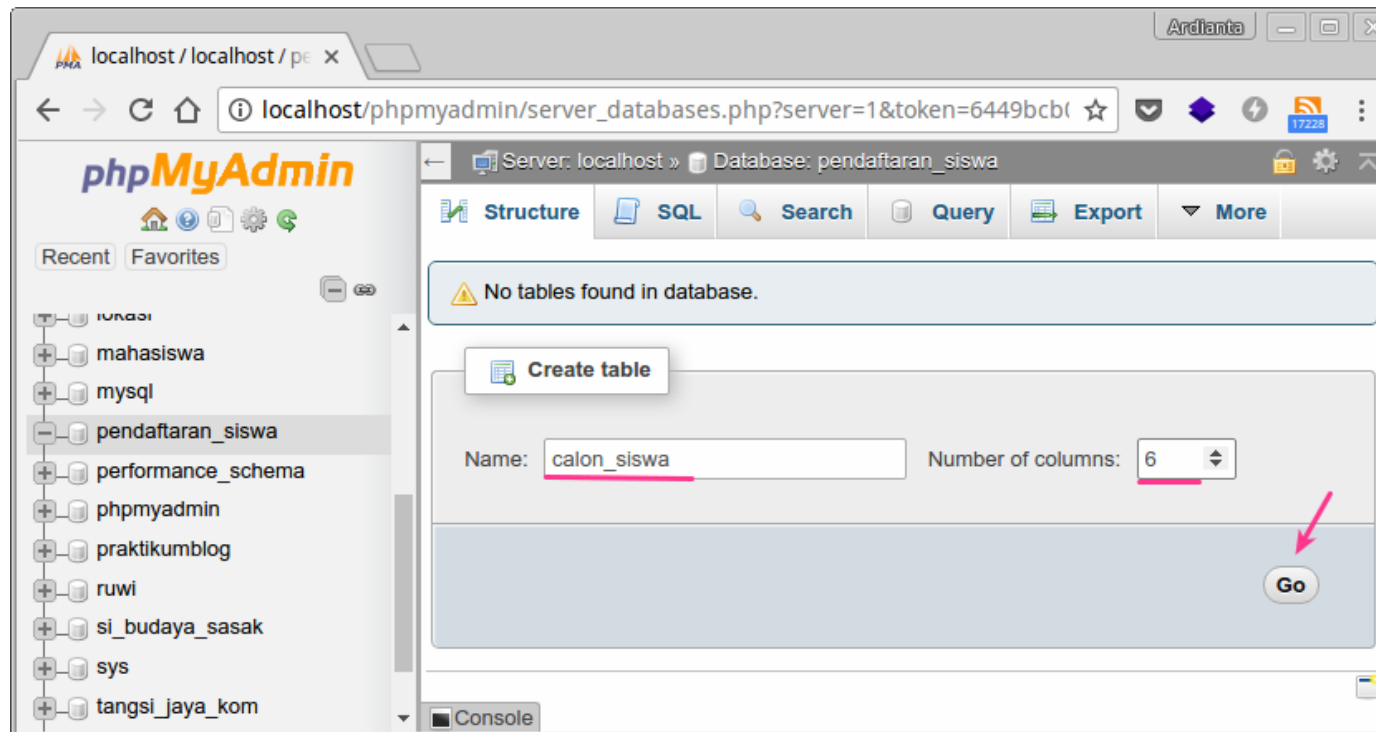
Pertama, buatlah database bernama pendaftaran_siswa. Atau dapat juga dengan menulis script sebagai berikut:

```
CREATE DATABASE `pendaftaran_siswa`;
```



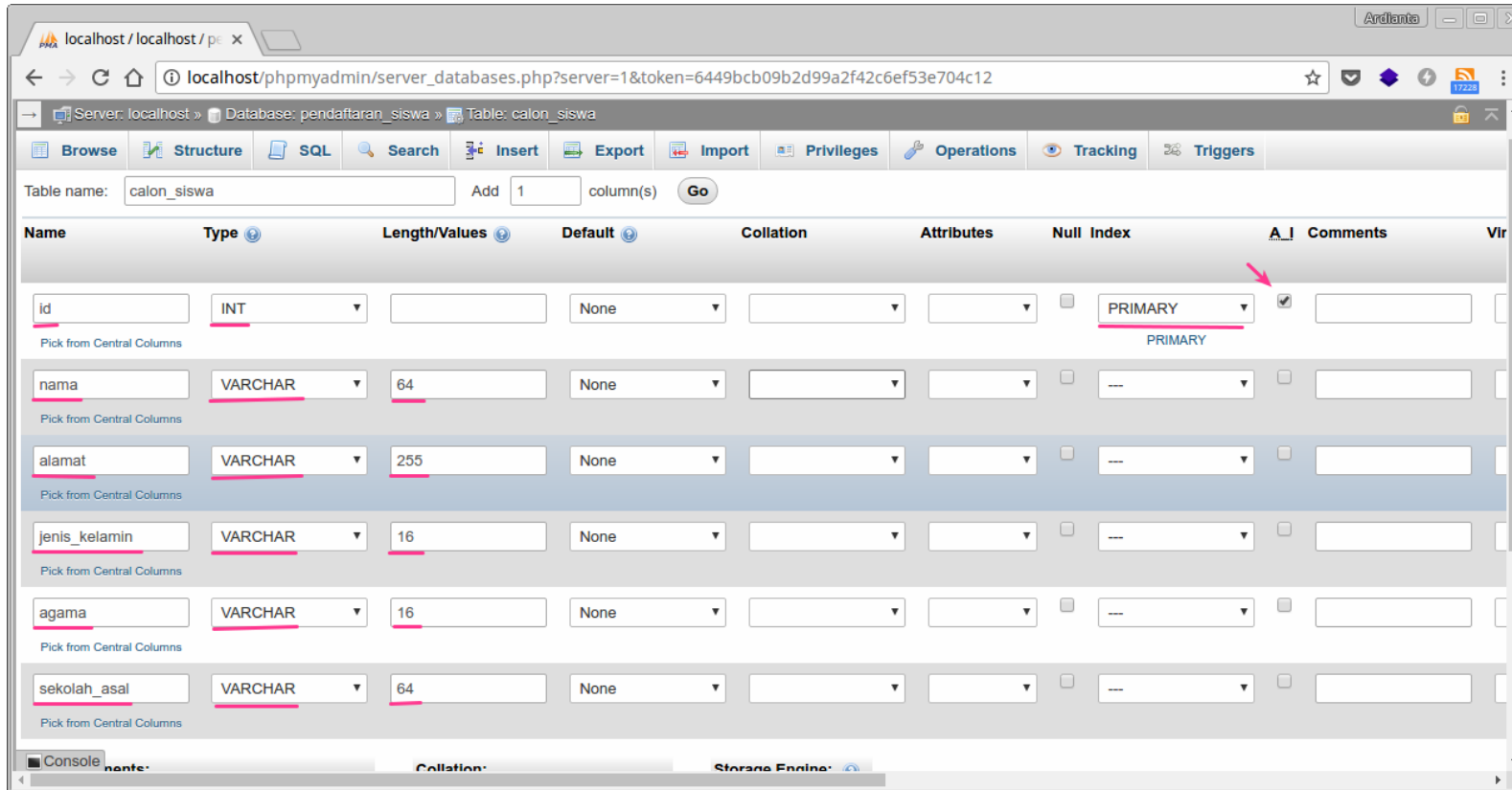
PHP & MySQL: Preparation Database (Continue)

Setelah itu, buatlah tabel di dalam database tersebut dengan nama **calon_siswa**, serta berikan tipe data kolomnya.



PHP & MySQL: Preparation Database (Continue)

Berikan tipe data berdasarkan gambar dibawah ini. Setelah itu klik save.



The screenshot shows the phpMyAdmin interface for editing the structure of a table named 'calon_siswa' in the 'pendaftaran_siswa' database. The interface includes a top navigation bar with tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers. Below the navigation bar, the table name 'calon_siswa' is entered, and the 'Add' button is set to '1 column(s)'. The main area displays a table structure with columns: id, nama, alamat, jenis_kelamin, agama, and sekolah_asal. Each column has a data type dropdown, a length/precision field, a default value dropdown, a collation dropdown, an attributes dropdown, a null checkbox, an index dropdown, and a comments field. The 'id' column is highlighted in blue, indicating it is the primary key. A red arrow points to the 'PRIMARY' index dropdown for the 'id' column. The 'Storage Engine' is set to 'InnoDB'.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	Comments
id	INT		None			☒	PRIMARY	
nama	VARCHAR	64	None			☐		
alamat	VARCHAR	255	None			☐		
jenis_kelamin	VARCHAR	16	None			☐		
agama	VARCHAR	16	None			☐		
sekolah_asal	VARCHAR	64	None			☐		

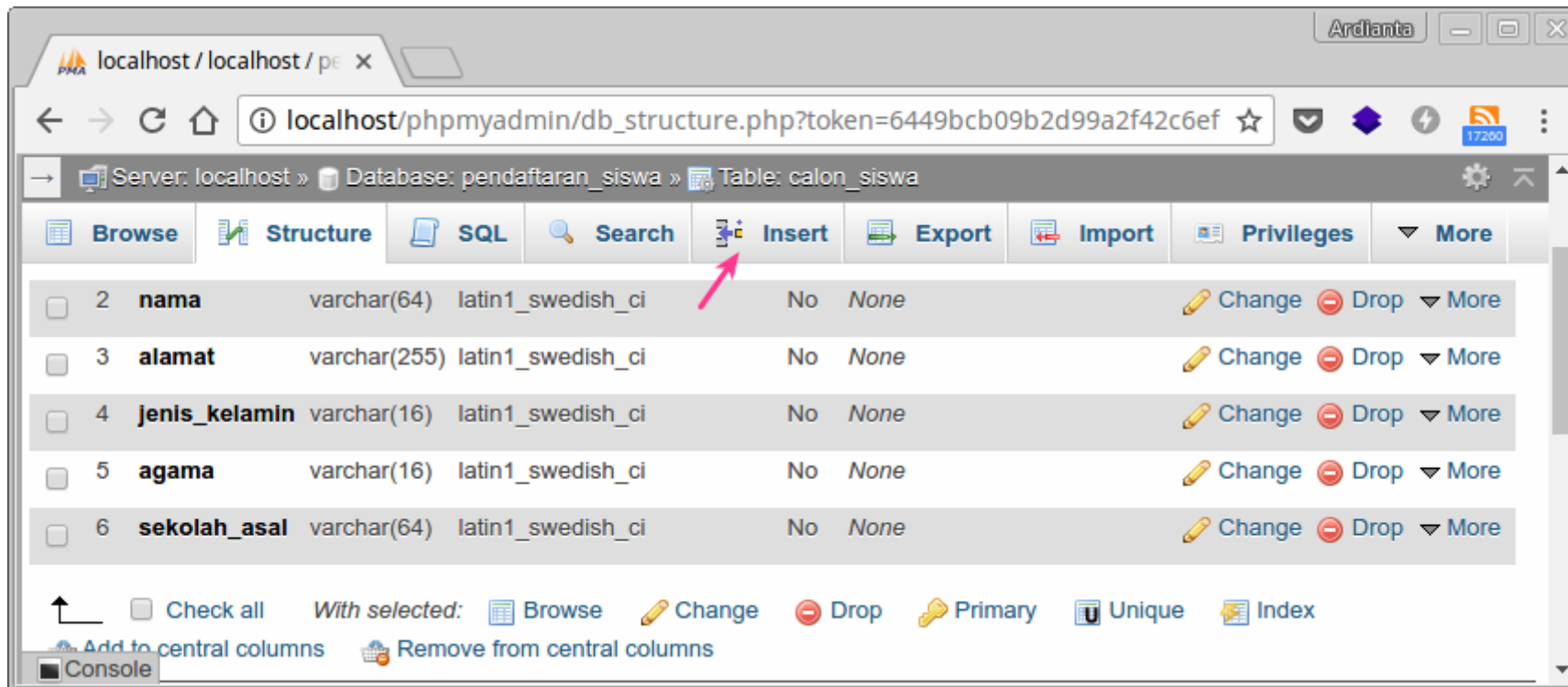
PHP & MySQL: Preparation Database (Continue)

Atau Anda dapat membuat table calon_siswa pada pendaftaran_siswa dengan script berikut.

```
CREATE TABLE `pendaftaran_siswa`.`calon_siswa` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `nama` VARCHAR(64) NOT NULL ,  
  `alamat` VARCHAR(255) NOT NULL ,  
  `jenis_kelamin` VARCHAR(16) NOT NULL ,  
  `agama` VARCHAR(16) NOT NULL ,  
  `sekolah_asal` VARCHAR(64) NOT NULL ,  
  PRIMARY KEY (`id`)  
) ENGINE = InnoDB;
```

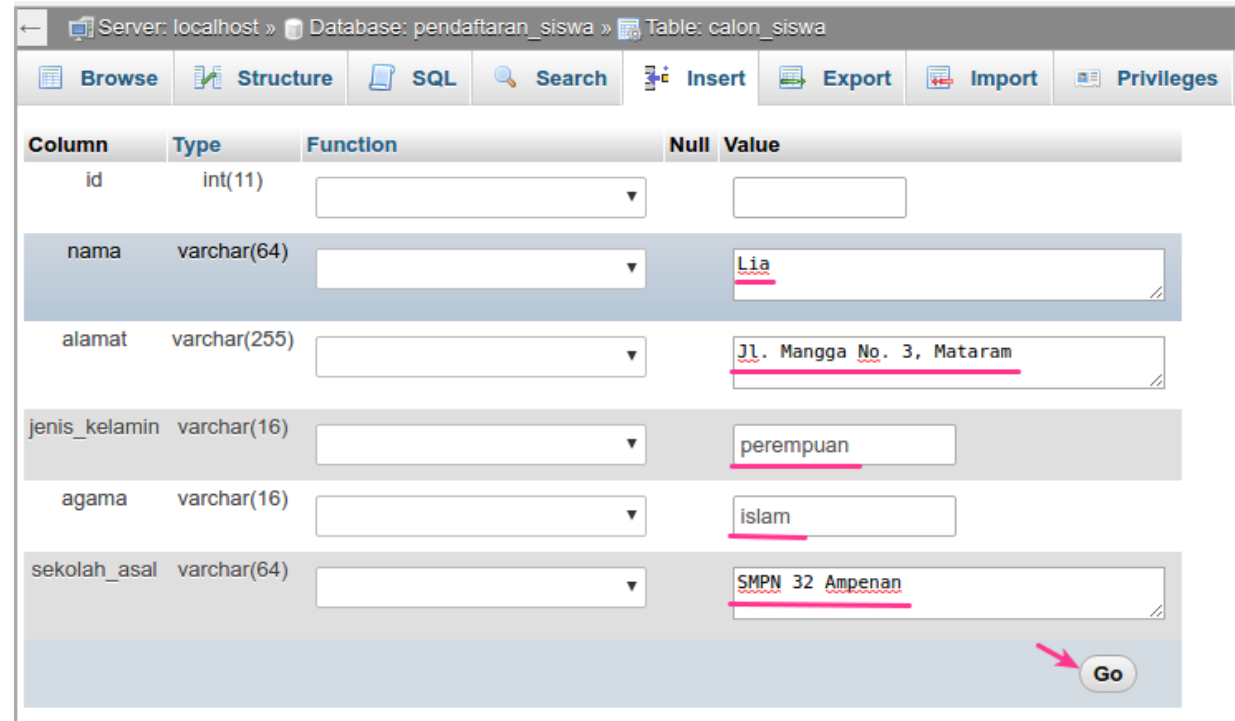

PHP & MySQL: Insert into Table

Setelah database dan table dibuat maka dapat coba diisi dengan klik insert.



PHP & MySQL: Insert into Table (Continue)

Masukan nilai secara bebas pada table calon_siswa atau bisa melalui script dibawah ini.

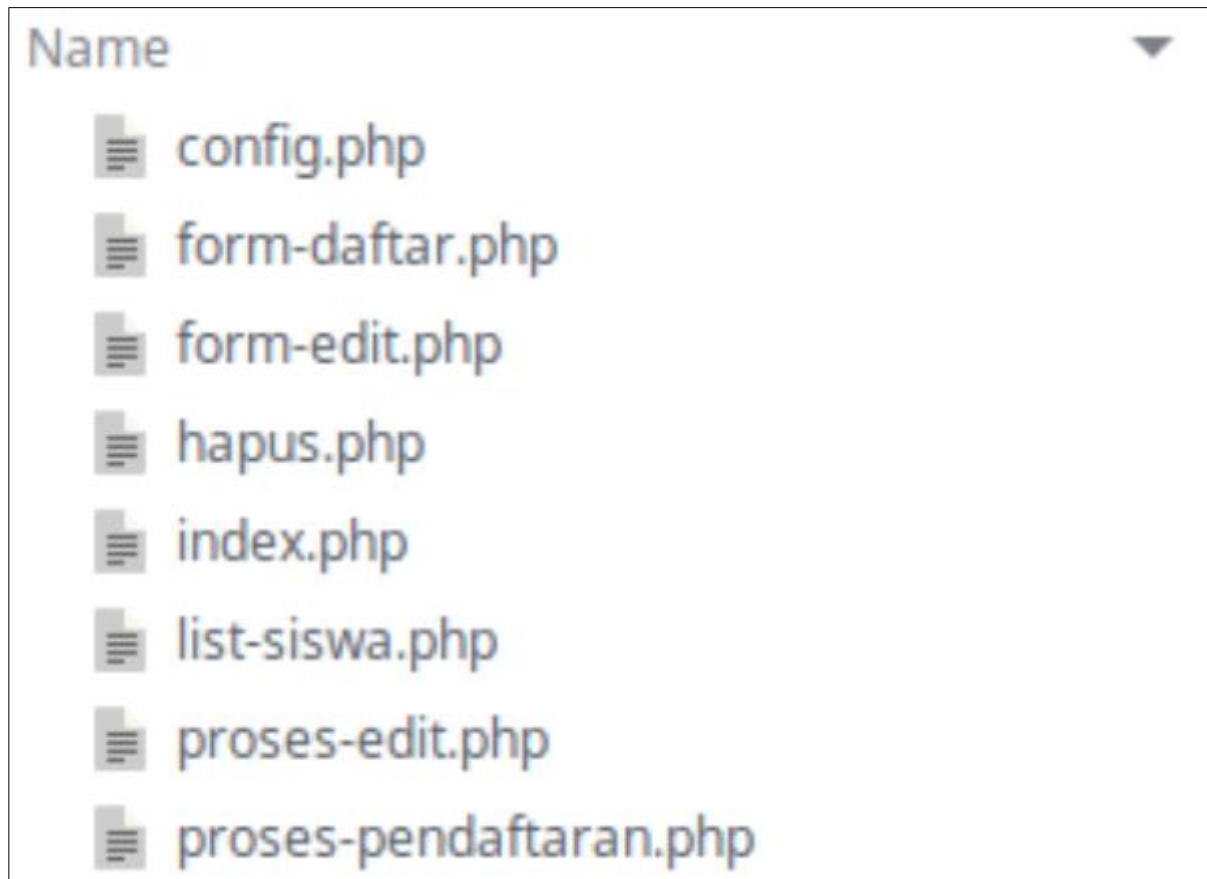


Column	Type	Function	Null	Value
id	int(11)			
nama	varchar(64)			Lia
alamat	varchar(255)			Jl. Mangga No. 3, Mataram
jenis_kelamin	varchar(16)			perempuan
agama	varchar(16)			islam
sekolah_asal	varchar(64)			SMPN 32 Ampenan

```
INSERT INTO `calon_siswa` (`id`, `nama`, `alamat`, `jenis_kelamin`, `agama`, `sekolah_asal`) VALUES (NULL, 'Lia', 'Jl. Mangga No. 3, Mataram', 'perempuan', 'islam', 'SMPN 32 Ampenan');
```

PHP & MySQL: Create the Project

Berikut merupakan struktur file untuk project yang akan dibuat.



Keterangan:

- config.php — untuk menyimpan konfigurasi database;
- index.php — halaman utama;
- list-siswa.php — halaman untuk menampilkan data siswa;
- form-daftar.php — halaman formulir pendaftaran;
- proses-pendaftaran.php — skrip yang memproses pendaftaran;
- form-edit.php — form untuk edit data siswa;
- proses-edit.php — skrip untuk memproses edit/update;
- hapus.php — skrip untuk menghapus data dari database;

PHP & MySQL: Connecting Database?

Buka file config.php, silahkan isi dengan kode berikut:

```
1  <?php
2  $server = "localhost";
3  $user = "root";
4  $password = "";
5  $nama_database = "pendaftaran_siswa";
6  $db = mysqli_connect($server, $user, $password, $nama_database);
7  if( !$db ){
8      die("Gagal terhubung dengan database: " . mysqli_connect_error());
9  }
10 ?>
```

PHP & MySQL: Connecting Database? (Continue)

- Untuk menghubungkan PHP dengan MySQL, kita menggunakan fungsi `mysqli_connect()` dengan parameter alamat server, user, password, dan nama database.
- Silahkan isi password sesuai dengan yang dibuat di servernya. Jika tidak menggunakan password, maka dikosongkan saja.

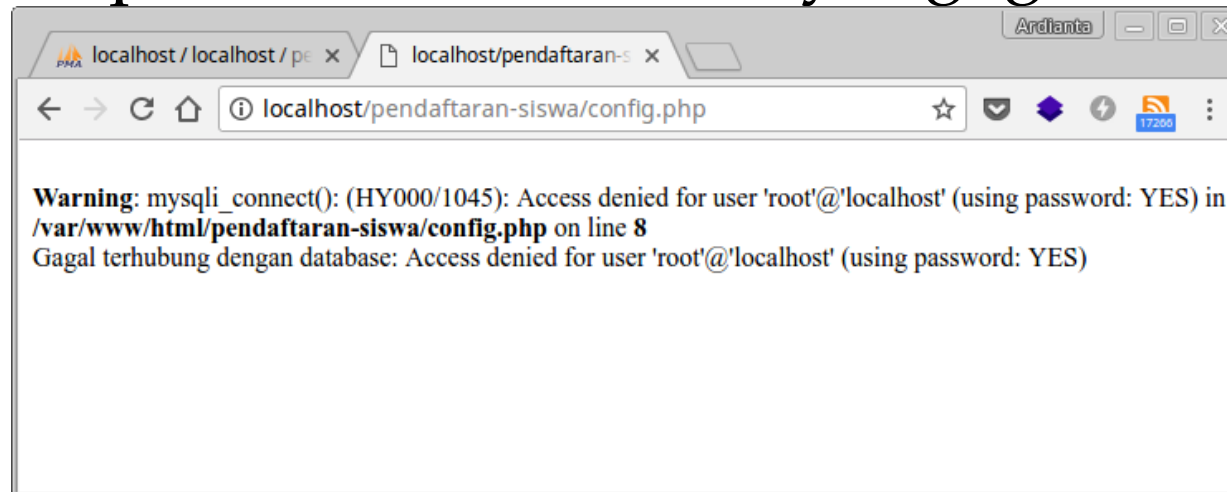
```
$password = "";
```

PHP & MySQL: Connecting Database? (Continue)

- Jika koneksi gagal, variabel `$db` akan bernilai false. Oleh Karena itu, kita bisa mengeceknya dengan ekspresi if.

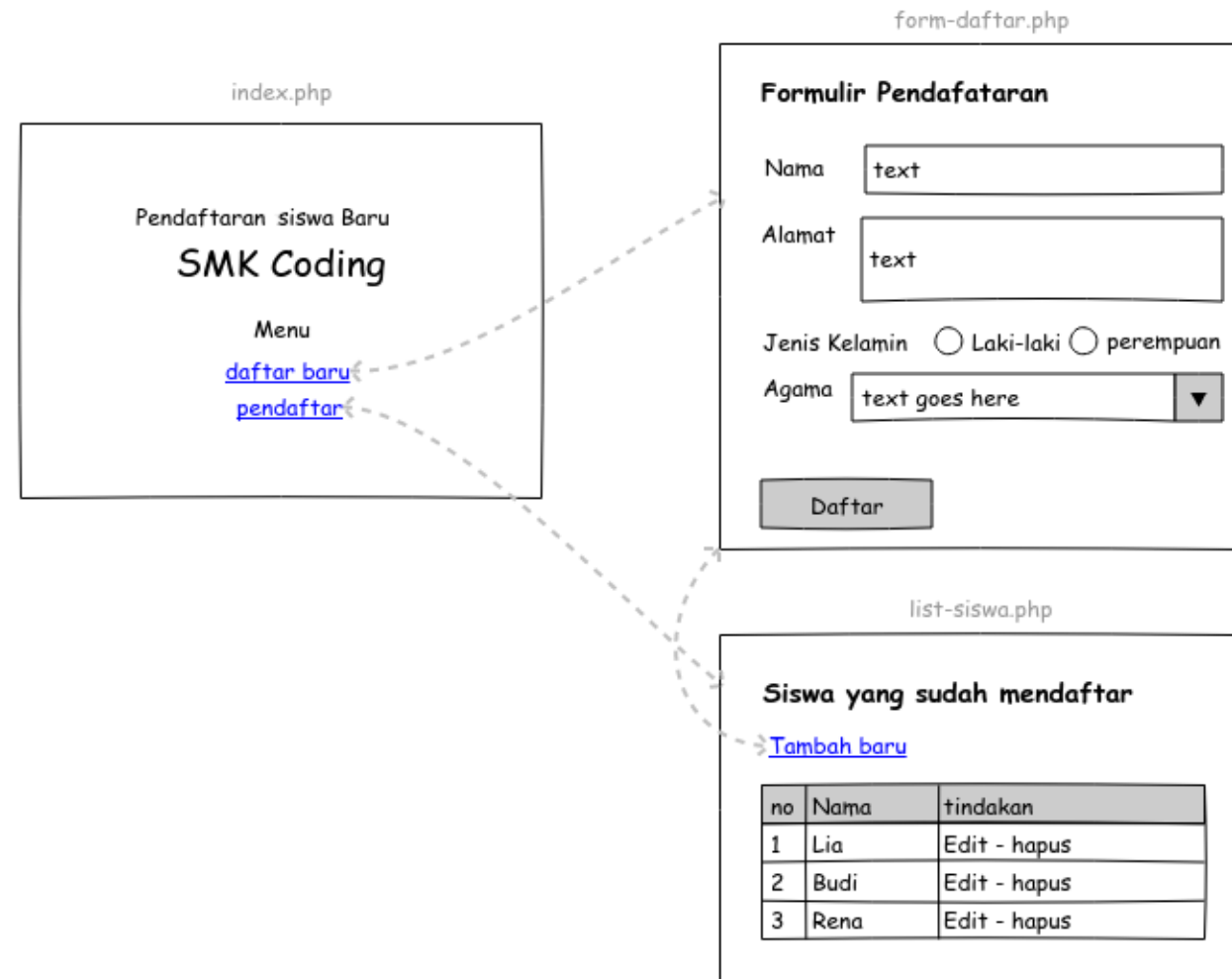
```
if( !$db ){  
    die("Gagal terhubung dengan database: " . mysqli_connect_error());  
}
```

- Berikut ini tampilan saat koneksinya gagal, karena password-nya salah.



PHP & MySQL: Create Index Page

Buatlah halaman index.php yang terdiri dari dua link. Link pertama untuk menuju ke formulir pendaftaran dan link yang kedua untuk menuju ke halaman data siswa yang sudah mendaftar.



PHP & MySQL: Create Index Page (Continue)

Isi dari file index.php.

```
<!DOCTYPE html>
<html>
<head>
    <title>Pendaftaran Siswa Baru | SMK Coding</title>
</head>

<body>
    <header>
        <h3>Pendaftaran Siswa Baru</h3>
        <h1>SMK Coding</h1>
    </header>

    <h4>Menu</h4>
    <nav>
        <ul>
            <li><a href="form-daftar.php">Daftar Baru</a></li>
            <li><a href="list-siswa.php">Pendaftar</a></li>
        </ul>
    </nav>

</body>
</html>
```


PHP & MySQL: Create View Page

Buatlah file list-siswa.php, dan berikut isi script tersebut.

```
1  <?php include("config.php"); ?>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>Pendaftaran Siswa Baru | SMK Coding</title>
6  </head>
7  <body>
8  <header>
9      <h3>Siswa yang sudah mendaftar</h3>
10 </header>
11 <nav>
12     <a href="form-daftar.php">[+] Tambah Baru</a>
13 </nav>
14 <br>
15 <table border="1">
16     <thead>
17     <tr>
18         <th>No</th>
19         <th>Nama</th>
20         <th>Alamat</th>
21         <th>Jenis Kelamin</th>
22         <th>Agama</th>
23         <th>Sekolah Asal</th>
24         <th>Tindakan</th>
25     </tr>
26     </thead>
27     <tbody>
```

```
27     <tbody>
28     <?php
29     $sql = "SELECT * FROM calon_siswa";
30     $query = mysqli_query($db, $sql);
31     while($siswa = mysqli_fetch_array($query)){
32         echo "<tr>";
33         echo "<td>".$siswa['id']. "</td>";
34         echo "<td>".$siswa['nama']. "</td>";
35         echo "<td>".$siswa['alamat']. "</td>";
36         echo "<td>".$siswa['jenis_kelamin']. "</td>";
37         echo "<td>".$siswa['agama']. "</td>";
38         echo "<td>".$siswa['sekolah_asal']. "</td>";
39         echo "<td>";
40         echo "<a href='form-edit.php?id=".$siswa['id']."'>Edit</a> | ";
41         echo "<a href='hapus.php?id=".$siswa['id']."'>Hapus</a>";
42         echo "</td>";
43         echo "</tr>";
44     }
45     ?>
46     </tbody>
47 </table>
48 <p>Total: <?php echo mysqli_num_rows($query) ?></p>
49 </body>
50 </html>
```

PHP & MySQL: Create View Page (Continue)

Penjelasan dari file list-siswa.php sebagai berikut:

- Pertama kita membutuhkan koneksi ke database, maka kita harus mengimpor file config.php agar variabel \$db dapat dibaca.

```
<?php include("config.php"); ?>
```

- Setelah itu, kita melakukan query untuk mengambil data siswa pada tabel calon_siswa.

```
$sql = "SELECT * FROM calon_siswa";  
$query = mysqli_query($db, $sql);
```

PHP & MySQL: Create View Page (Continue)

- Hasil dari query tersebut, kemudian kita ubah menjadi array dengan fungsi `mysqli_fetch_array()` dan hasilnya disimpan dalam variabel `$siswa`.

```
$siswa = mysqli_fetch_array($query)
```

- Perulangan `while` akan mengulang selama masih ada isi dalam tabel. Kemudian, selama proses perulangan, maka data akan terus tampil.

PHP & MySQL: Create View Page (Continue)

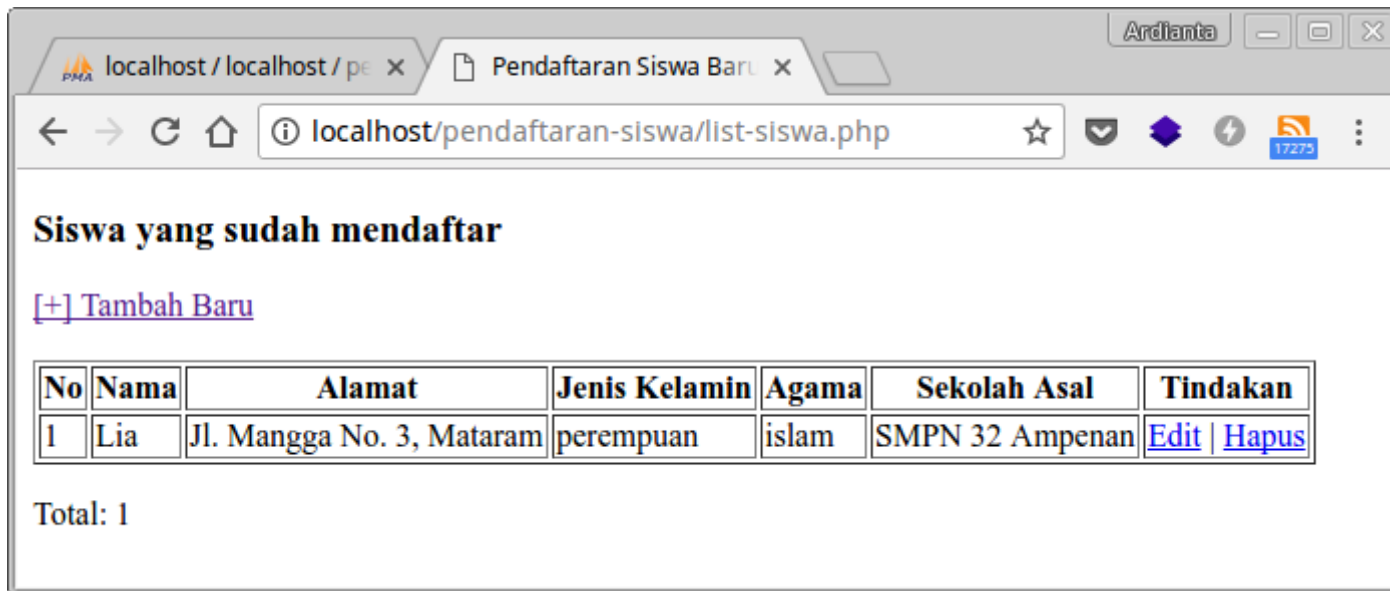
- Perhatikan array \$siswa, indeksnya sesuai dengan nama kolom pada database dan bersifat case sensitive.

```
31 while($siswa = mysqli_fetch_array($query)){
32     echo "<tr>";
33     echo "<td>".$siswa['id']. "</td>";
34     echo "<td>".$siswa['nama']. "</td>";
35     echo "<td>".$siswa['alamat']. "</td>";
36     echo "<td>".$siswa['jenis_kelamin']. "</td>";
37     echo "<td>".$siswa['agama']. "</td>";
38     echo "<td>".$siswa['sekolah_asal']. "</td>";
39     echo "<td>";
40     echo "<a href='form-edit.php?id=".$siswa['id']."'>Edit</a> | ";
41     echo "<a href='hapus.php?id=".$siswa['id']."'>Hapus</a>";
42     echo "</td>";
43     echo "</tr>";
44 }
```

PHP & MySQL: Create View Page (Continue)

Terakhir, kita menampilkan total data yang ada di dalam tabel dengan fungsi `mysqli_num_rows()`. Selain itu, terdapat tampilan `list-siswa.php`

```
echo mysqli_num_rows($query);
```



Siswa yang sudah mendaftar

[\[+\] Tambah Baru](#)

No	Nama	Alamat	Jenis Kelamin	Agama	Sekolah Asal	Tindakan
1	Lia	Jl. Mangga No. 3, Mataram	perempuan	islam	SMPN 32 Ampenan	Edit Hapus

Total: 1

PHP & MySQL: Insert Data Page

Ada dua file yang kita butuhkan untuk menambahkan data ke MySQL.

- **form-daftar.php** halaman untuk form input;
- **proses-pendaftaran.php** skrip untuk memproses data yang diinputkan.
- Agar lebih dipahami, dibuat menjadi terpisah.

PHP & MySQL: Insert Data Page (Continue)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Formulir Pendaftaran Siswa Baru | SMK Coding</title>
5 </head>
6 <body>
7 <header>
8 <h3>Formulir Pendaftaran Siswa Baru</h3>
9 </header>
10 <form action="proses-pendaftaran.php" method="POST">
11 <fieldset>
12 <p>
13 <label for="nama">Nama: </label>
14 <input type="text" name="nama" placeholder="nama lengkap" />
15 </p>
16 <p>
17 <label for="alamat">Alamat: </label>
18 <textarea name="alamat"></textarea>
19 </p>
20 <p>
21 <label for="jenis_kelamin">Jenis Kelamin: </label>
22 <label><input type="radio" name="jenis_kelamin" value="laki-laki"> Laki-laki</label>
23 <label><input type="radio" name="jenis_kelamin" value="perempuan"> Perempuan</label>
24 </p>
25 <p>
26 <label for="agama">Agama: </label>
27 <select name="agama">
28 <option>Islam</option>
29 <option>Kristen</option>
30 <option>Hindu</option>
31 <option>Budha</option>
32 <option>Atheis</option>
33 </select>
```

```
34 </p>
35 <p>
36 <label for="sekolah_asal">Sekolah Asal: </label>
37 <input type="text" name="sekolah_asal" placeholder="nama sekolah" />
38 </p>
39 <p>
40 <input type="submit" value="Daftar" name="daftar" />
41 </p>
42 </fieldset>
43 </form>
44 </body>
45 </html>
```

Script form-daftar.php

PHP & MySQL: Insert Data Page (Continue)

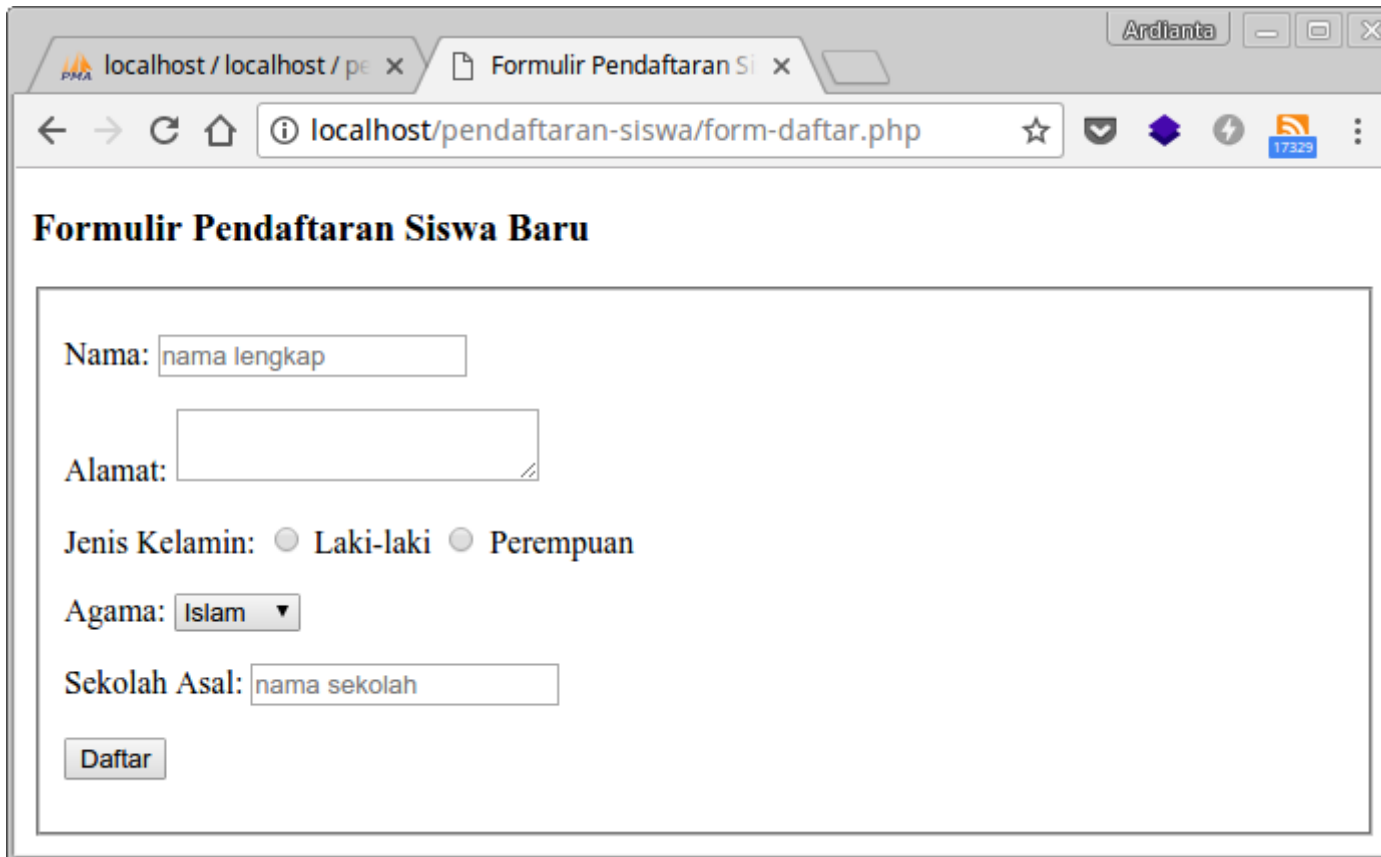
- Perhatikan pada tag pembuka <form>, terdapat atribut action dan method POST.

```
<form action="proses-pendaftaran.php" method="POST">  
...
```

- Atribut action digunakan untuk menentukan skrip mana yang akan memproses data inputan. Sedangkan atribut method untuk menentukan metode pengiriman datanya.

PHP & MySQL: Insert Data Page (Continue)

- Tampilan form-daftar.php



The screenshot shows a web browser window with the title 'Formulir Pendaftaran Siswa Baru'. The browser's address bar displays 'localhost/pendaftaran-siswa/form-daftar.php'. The form contains the following fields and controls:

- Nama:** A text input field with the placeholder text 'nama lengkap'.
- Alamat:** A text input field.
- Jenis Kelamin:** Two radio buttons labeled 'Laki-laki' and 'Perempuan'.
- Agama:** A dropdown menu with 'Islam' selected.
- Sekolah Asal:** A text input field with the placeholder text 'nama sekolah'.
- Daftar:** A button to submit the form.

PHP & MySQL: Data Input Process

- Script proses-pendaftaran.php
- Pertama mengambil data yang dikirim dari formulir, semuanya tersimpan pada variabel `$_POST`. Karena sebelumnya menggunakan method POST pada form-daftar.php

```
1 <?php
2 include("config.php");
3
4 // cek apakah tombol daftar sudah diklik atau blum?
5 if(isset($_POST['daftar'])){
6     // ambil data dari formulir
7     $nama = $_POST['nama'];
8     $alamat = $_POST['alamat'];
9     $jk = $_POST['jenis_kelamin'];
10    $agama = $_POST['agama'];
11    $sekolah = $_POST['sekolah_asal'];
12
13    // buat query
14    $sql = "INSERT INTO calon_siswa (nama, alamat, jenis_kelamin, agama, sekolah_asal) VALUE ($nama, '$alamat', '$jk', '$agama', '$sekolah)";
15    $query = mysqli_query($db, $sql);
16
17    // apakah query simpan berhasil?
18    if( $query ) {
19        // kalau berhasil alihkan ke halaman index.php dengan status=sukses
20        header( header: 'Location: index.php?status=sukses');
21    } else {
22        // kalau gagal alihkan ke halaman indek.php dengan status=gagal
23        header( header: 'Location: index.php?status=gagal');
24    }
25 } else {
26     die("Akses dilarang...");
27 }
28
29 >?
```

PHP & MySQL: Data Input Process (Continue)

- Setelah itu, harus dilakukan pengecekan, apakah query-nya berhasil atau gagal.
- Untuk menampilkan pesan berhasil dan gagal pada halaman index.php, silahkan tambahkan kode berikut di bawah tag <nav> pada file **index.php**:

```
<?php if(isset($_GET['status'])): ?>
    <p>
        <?php
            if($_GET['status'] == 'sukses'){
                echo "Pendaftaran siswa baru berhasil!";
            } else {
                echo "Pendaftaran gagal!";
            }
        ?>
    </p>
<?php endif; ?>
```

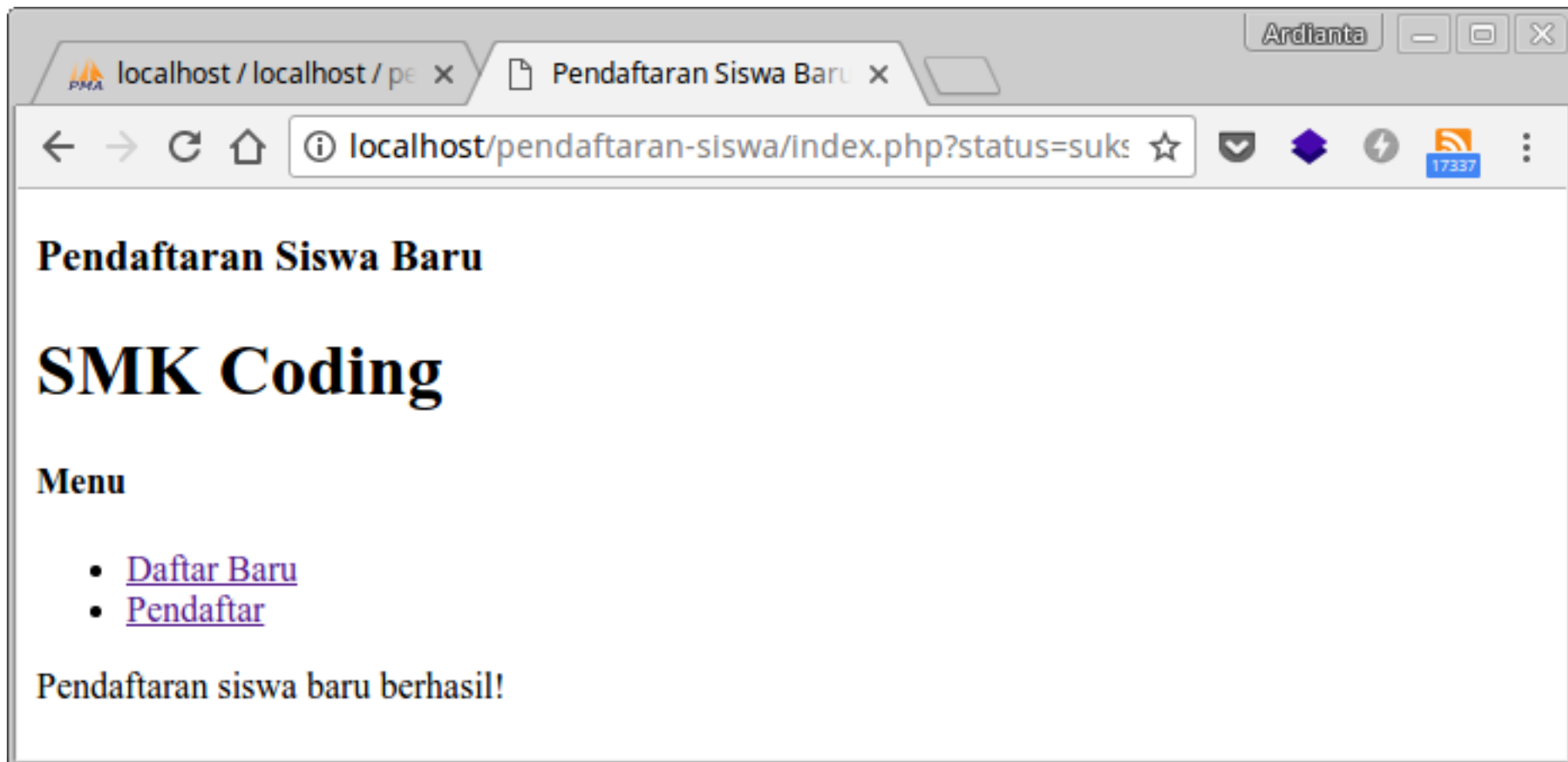
PHP & MySQL: Data Input Process (Continue)

Updated script terbaru index.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Pendaftaran Siswa Baru | SMK Coding</title>
5  </head>
6  <body>
7  <header>
8      <h3>Pendaftaran Siswa Baru</h3>
9      <h1>SMK Coding</h1>
10 </header>
11 <h4>Menu</h4>
12 <nav>
13     <ul>
14         <li><a href="form-daftar.php">Daftar Baru</a></li>
15         <li><a href="list-siswa.php">Pendaftar</a></li>
16     </ul>
17 </nav>
18 <?php if(isset($_GET['status'])): ?>
19     <p>
20         <?php
21         if($_GET['status'] == 'sukses'){
22             echo "Pendaftaran siswa baru berhasil!";
23         } else {
24             echo "Pendaftaran gagal!";
25         }
26         ?>
27     </p>
28 <?php endif; ?>
29 </body>
30 </html>
```

PHP & MySQL: Data Input Process (Continue)

Tampilan index.html setelah data diinputkan.



PHP & MySQL: Delete Data

- Pada saat menghapus, maka membutuhkan **id** untuk menentukan data yang akan dihapus. Karena itu, harus mengirimkan **id** melalui query string.

- Contoh:

```
http://localhost/pendaftaran-siswa/hapus.php?id=1
```

- Query string-nya adalah ?id=1. Query string akan tersimpan dalam variabel `$_GET`.

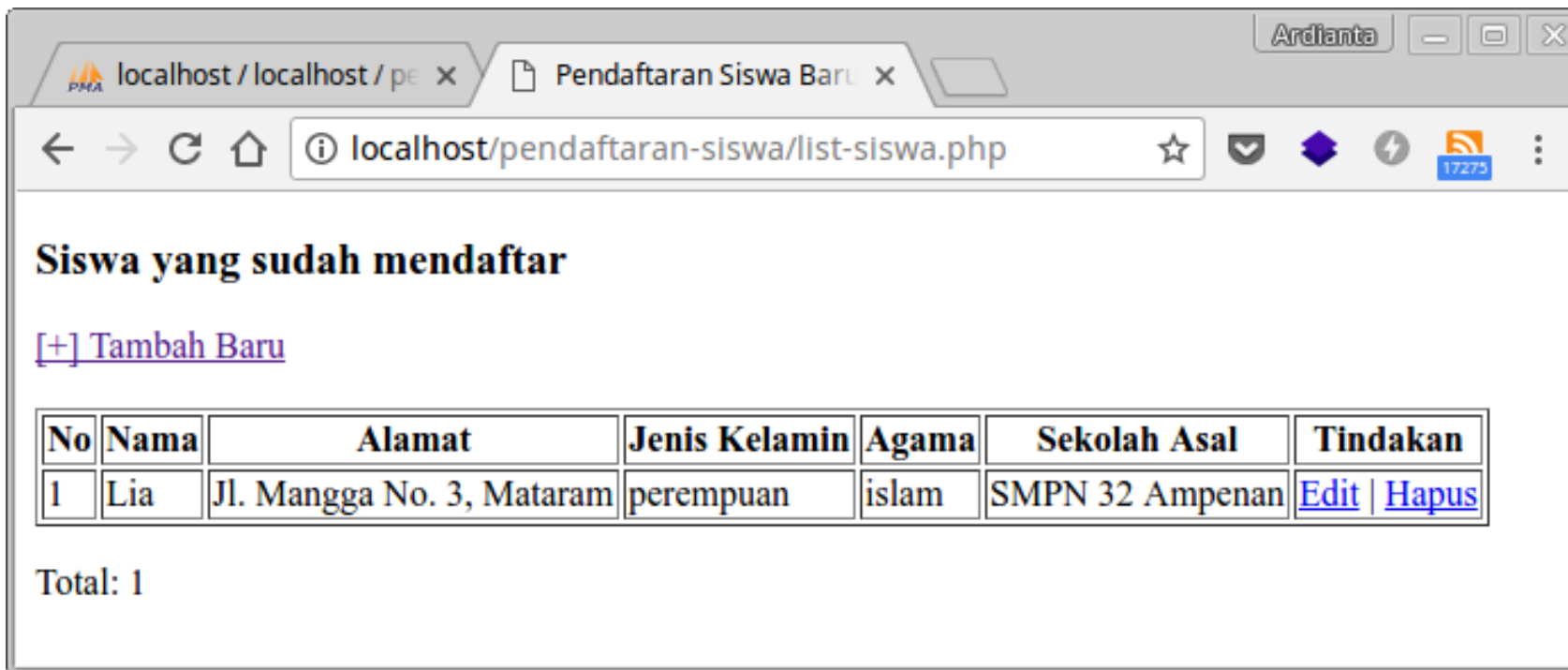
PHP & MySQL: Delete Data (Continue)

Untuk lebih jelasnya, silahkan buatlah file hapus.php dan isi dengan kode berikut:

```
1  <?php
2
3  include("config.php");
4
5  if( isset($_GET['id']) ){
6
7      // ambil id dari query string
8      $id = $_GET['id'];
9
10     // buat query hapus
11     $sql = "DELETE FROM calon_siswa WHERE id=$id";
12     $query = mysqli_query($db, $sql);
13
14     // apakah query hapus berhasil?
15     if( $query ){
16         header( header: 'Location: list-siswa.php' );
17     } else {
18         die("Gagal menghapus...");
19     }
20
21 } else {
22     die("Akses dilarang...");
23 }
24
25 ?>
```

PHP & MySQL: Delete Data (Continue)

Sekarang coba klik link hapus, maka datanya akan langsung terhapus. Catatan, sebetulnya cara ini kurang aman, karena tidak memberikan dialog konfirmasi kepada user. Misalnya seperti menanyakan 'Apakah yakin akan menghapus?'



localhost / localhost / pe x Pendaftaran Siswa Baru x

localhost/pendaftaran-siswa/list-siswa.php

Siswa yang sudah mendaftar

[\[+\] Tambah Baru](#)

No	Nama	Alamat	Jenis Kelamin	Agama	Sekolah Asal	Tindakan
1	Lia	Jl. Mangga No. 3, Mataram	perempuan	islam	SMPN 32 Ampenan	Edit Hapus

Total: 1

PHP & MySQL: Update Data

- Konsep update data hampir sama seperti menghapus.
- Pertama mengambil **id** data yang akan di-edit. Setelah itu kita melakukan query untuk mengambil data dari database dan menampilkannya ke dalam form.
- Setelah di-edit, selanjutnya melakukan query update untuk menyimpan perubahan.

PHP & MySQL: Update Data (Continue)

Untuk lebih jelasnya, silahkan buatlah file form-edit.php dan isi dengan kode berikut:

```
1 <?php
2 include("config.php");
3 // kalau tidak ada id di query string
4 if( !isset($_GET['id']) ){
5     header( header: 'Location: list-siswa.php' );
6 }
7 //ambil id dari query string
8 $id = $_GET['id'];
9 // buat query untuk ambil data dari database
10 $sql = "SELECT * FROM calon_siswa WHERE id=$id";
11 $query = mysqli_query($db, $sql);
12 $siswa = mysqli_fetch_assoc($query);
13 // jika data yang di-edit tidak ditemukan
14 if( mysqli_num_rows($query) < 1 ){
15     die("data tidak ditemukan...");
16 }
17 ?>
```

PHP & MySQL: Update Data (Continue)

```
19 <!DOCTYPE html>
20 <html>
21 <head>
22 <title>Formulir Edit Siswa | SMK Coding</title>
23 </head>
24 <body>
25 <header>
26 <h3>Formulir Edit Siswa</h3>
27 </header>
28 <form action="proses-edit.php" method="POST">
29 <fieldset>
30 <input type="hidden" name="id" value="<?php echo $siswa['id'] ?>" />
31 <p>
32 <label for="nama">Nama: </label>
33 <input type="text" name="nama" placeholder="nama lengkap" value="<?php echo $siswa['nama'] ?>" />
34 </p>
35 <p>
36 <label for="alamat">Alamat: </label>
37 <textarea name="alamat"><?php echo $siswa['alamat'] ?></textarea>
38 </p>
39 <p>
40 <label for="jenis_kelamin">Jenis Kelamin: </label>
41 <?php $jk = $siswa['jenis_kelamin']; ?>
42 <label><input type="radio" name="jenis_kelamin" value="laki-laki" <?php echo ($jk == 'laki-laki') ? "checked": "" ?>> Laki-laki</label>
43 <label><input type="radio" name="jenis_kelamin" value="perempuan" <?php echo ($jk == 'perempuan') ? "checked": "" ?>> Perempuan</label>
44 </p>
45 <p>
46 <label for="agama">Agama: </label>
47 <?php $agama = $siswa['agama']; ?>
48 <select name="agama">
49 <option <?php echo ($agama == 'Islam') ? "selected": "" ?>>Islam</option>
50 <option <?php echo ($agama == 'Kristen') ? "selected": "" ?>>Kristen</option>
51 <option <?php echo ($agama == 'Hindu') ? "selected": "" ?>>Hindu</option>
52 <option <?php echo ($agama == 'Budha') ? "selected": "" ?>>Budha</option>
53 <option <?php echo ($agama == 'Atheis') ? "selected": "" ?>>Atheis</option>
54 </select>
55 </p>
56 <p>
57 <label for="sekolah_asal">Sekolah Asal: </label>
58 <input type="text" name="sekolah_asal" placeholder="nama sekolah" value="<?php echo $siswa['sekolah_asal'] ?>" />
59 </p>
60 <p>
61 <input type="submit" value="Simpan" name="simpan" />
62 </p>
63 </fieldset>
64 </form>
65 </body>
66 </html>
```

PHP & MySQL: Update Data (Continue)

Ada beberapa hal yang perlu diperhatikan dalam kode tersebut.

- Pertama kita mengubah data dari hasil query menjadi array dengan menggunakan fungsi `mysqli_fetch_assoc()`.
- Alasan menggunakan fungsi tersebut, karena data yang akan diambil adalah satu baris dan agar indeks array-nya menggunakan nama kolom.
- Contoh hasilnya seperti ini:

```
Array
(
    [id] => 1
    [nama] => Lia
    [alamat] => Jl. Mangga No. 3, Mataram
    [jenis_kelamin] => perempuan
    [agama] => Islam
    [sekolah_asal] => SMPN 32 Ampenan
)
```


PHP & MySQL: Update Data (Continue)

- Data tersebut yang tampilkan untuk menjadi nilai default pada form.
- Jangan lupa untuk membuat input **id** dengan tipe hidden untuk menyimpan **id** data yang akan di-update.

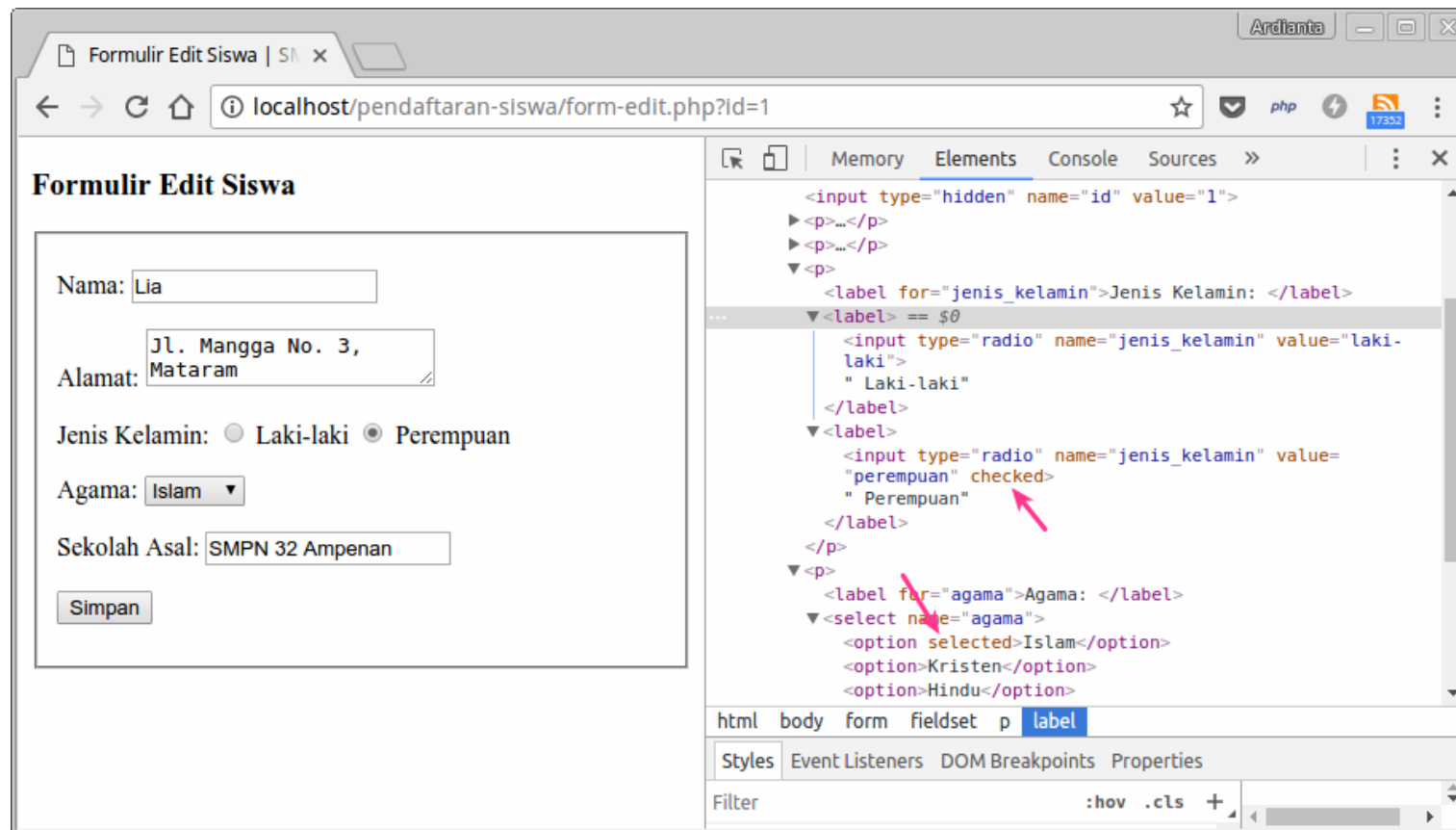
```
<input type="hidden" name="id" value="<?php echo $siswa['id'] ?>" />
```

- Untuk input radio dan select, dapat menggunakan operator ternary agar dapat membuat atribut checked dan selected.

```
39 <p>
40 <label for="jenis_kelamin">Jenis Kelamin: </label>
41 <?php $jk = $siswa['jenis_kelamin']; ?>
42 <label><input type="radio" name="jenis_kelamin" value="laki-laki" <?php echo ($jk == 'laki-laki') ? "checked": "" ?>> Laki-laki</label>
43 <label><input type="radio" name="jenis_kelamin" value="perempuan" <?php echo ($jk == 'perempuan') ? "checked": "" ?>> Perempuan</label>
44 </p>
45 <p>
46 <label for="agama">Agama: </label>
47 <?php $agama = $siswa['agama']; ?>
48 <select name="agama">
49 <option <?php echo ($agama == 'Islam') ? "selected": "" ?>>Islam</option>
50 <option <?php echo ($agama == 'Kristen') ? "selected": "" ?>>Kristen</option>
51 <option <?php echo ($agama == 'Hindu') ? "selected": "" ?>>Hindu</option>
52 <option <?php echo ($agama == 'Budha') ? "selected": "" ?>>Budha</option>
53 <option <?php echo ($agama == 'Atheis') ? "selected": "" ?>>Atheis</option>
54 </select>
55 </p>
```


PHP & MySQL: Update Data (Continue)

Ketika kita lihat dari inspect elements, maka akan tampil seperti ini:



PHP & MySQL: Update Data (Continue)

Setelah itu, buatlah file proses-edit.php dengan kode berikut:

```
1 <?php
2 include("config.php");
3 // cek apakah tombol simpan sudah diklik atau blum?
4 if(isset($_POST['simpan'])){
5     // ambil data dari formulir
6     $id = $_POST['id'];
7     $nama = $_POST['nama'];
8     $alamat = $_POST['alamat'];
9     $jk = $_POST['jenis_kelamin'];
10    $agama = $_POST['agama'];
11    $sekolah = $_POST['sekolah_asal'];
12    // buat query update
13    $sql = "UPDATE calon_siswa SET nama='$nama', alamat='$alamat', jenis_kelamin='$jk', agama='$agama', sekolah_asal='$sekolah' WHERE id=$id";
14    $query = mysqli_query($db, $sql);
15    // apakah query update berhasil?
16    if( $query ) {
17        // kalau berhasil alihkan ke halaman list-siswa.php
18        header( header: 'Location: list-siswa.php');
19    } else {
20        // kalau gagal tampilkan pesan
21        die("Gagal menyimpan perubahan...");
22    }
23 } else {
24     die("Akses dilarang...");
25 }
26 >>
```

PHP & MySQL: Update Data (Continue)

Silahkan simpan dan cobalah untuk mengedit data.



Siswa yang sudah mendaftar

[\[+\] Tambah Baru](#)

No	Nama	Alamat	Jenis Kelamin	Agama	Sekolah Asal	Tindakan
1	Priscilia Yuliani	Jl. Mangga No. 3, Mataram	perempuan	Islam	SMPN 32 Ampenan	Edit Hapus

Total: 1

References

1. <https://www.php.net/docs.php>
2. <https://www.petanikode.com/tutorial-php-mysql/>

THANK YOU



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web

Pertemuan 12

PHP & AJAX

Zuhri Halim, S.Kom., M.Kom.

Outline

1. AJAX Introduction
2. How to Work?
3. HTML DOM
4. JSON
5. AJAX & PHP



Introduction: Learning Outcomes

Mahasiswa mampu menganalisa script server side yaitu PHP

Mahasiswa mampu menjelaskan fungsi AJAX

Mahasiswa mampu menganalisis script yang menggunakan AJAX

AJAX Intorduction



AJAX Introduction

AJAX atau Asynchronous JavaScript and XML adalah teknik yang digunakan untuk membuat website yang dinamis. Artinya website mampu mengupdate dan menampilkan data baru dari server tanpa perlu melakukan reload.

Contoh penggunaan:



AJAX Introduction (Continue)

- AJAX terdiri dari JavaScript dan XML yang bekerja bersama. JavaScript adalah bahasa pemrograman untuk mengelola konten website yang dinamis. Sementara XML (eXtensible Markup Language) digunakan untuk memuat dan membawa data dari server ke browser.
- AJAX JavaScript dan XML ini bekerja secara asynchronous untuk berkomunikasi dengan server. Proses pertukaran informasi ini dilakukan di background. Artinya, saat AJAX JavaScript dan XML bekerja, halaman dapat tetap diakses oleh pengunjung website.

AJAX Introduction (Continue)

Teknologi yang ada dalam AJAX:

- HTML yang digunakan untuk membuat Web forms dan mengindentifikasikan field-field yang akan anda gunakan dalam aplikasi.
- JavaScript adalah kode inti untuk menjalankan aplikasi Ajax dan untuk membantu memfasilitasi komunikasi dengan aplikasi.
- DHTML, atau Dynamic HTML, membantu anda untuk membuat form atau web anda dinamis. Anda akan menggunakan `<div>`, `<span>` dan elemen HTML dinamis lainnya.
- DOM, Document Object Model, akan digunakan (melalui kode JavaScript) untuk bekerja dengan kedua struktur dari HTML dan XML anda yang dalam beberapa kasus berasal dari server.

AJAX Introduction (Continue)

Keuntungan:

- Tidak harus refresh seluruh halaman web karena hanya bagian tertentu saja yang diupdate.
- Traffic data antara server dan client lebih kecil.
- Lebih cepat
- Lebih interaktif

Kerugian:

- Halaman yang datanya diambil dengan AJAX tidak terindex di mesin pencari (tidak SEO-friendly)
- Sourcecode bisa diview (javascript bisa di view-source). Ini bisa memunculkan masalah di sisi keamanan.
- Tidak berjalan di browser yang tidak support javascript atau javascript di-disable.

AJAX Introduction: Synchronous vs Asynchronous

Synchronous:

Synchronous terjadi ketika sekumpulan statement dieksekusi secara berurutan dan dilakukan satu per satu. Artinya statement kedua dilaksanakan setelah statement pertama, dan statement ketiga dilaksanakan setelah statement kedua dan seterusnya.

Hasil eksekusinya pun akan sesuai dengan urutan eksekusi statement.

Aysnchronous:

Asynchronous terjadi ketika sekumpulan statement dieksekusi berurutan tetapi hasilnya bisa tidak sesuai urutan statementnya. Ada kemungkinan statement ketiga bisa selesai terlebih dahulu, baru kemudian hasil statement 2 selesai didapatkan

AJAX Introduction: Synchronous vs Asynchronous (Continue)

Synchronous	Asynchronous
Contoh Statement : statement_sync(1); statement_sync(2); statement_sync(3);	Contoh Statement : statement_async(1); statement_async(2); statement_async(3);
Hasil Run : 1 2 3	Hasil Run : 1 3 2

Ada kemungkinan hasil tidak sesuai urutan statement (menyesuaikan dengan statement mana yang selesai terlebih dahulu).

AJAX Introduction: What's the Prerequisite?

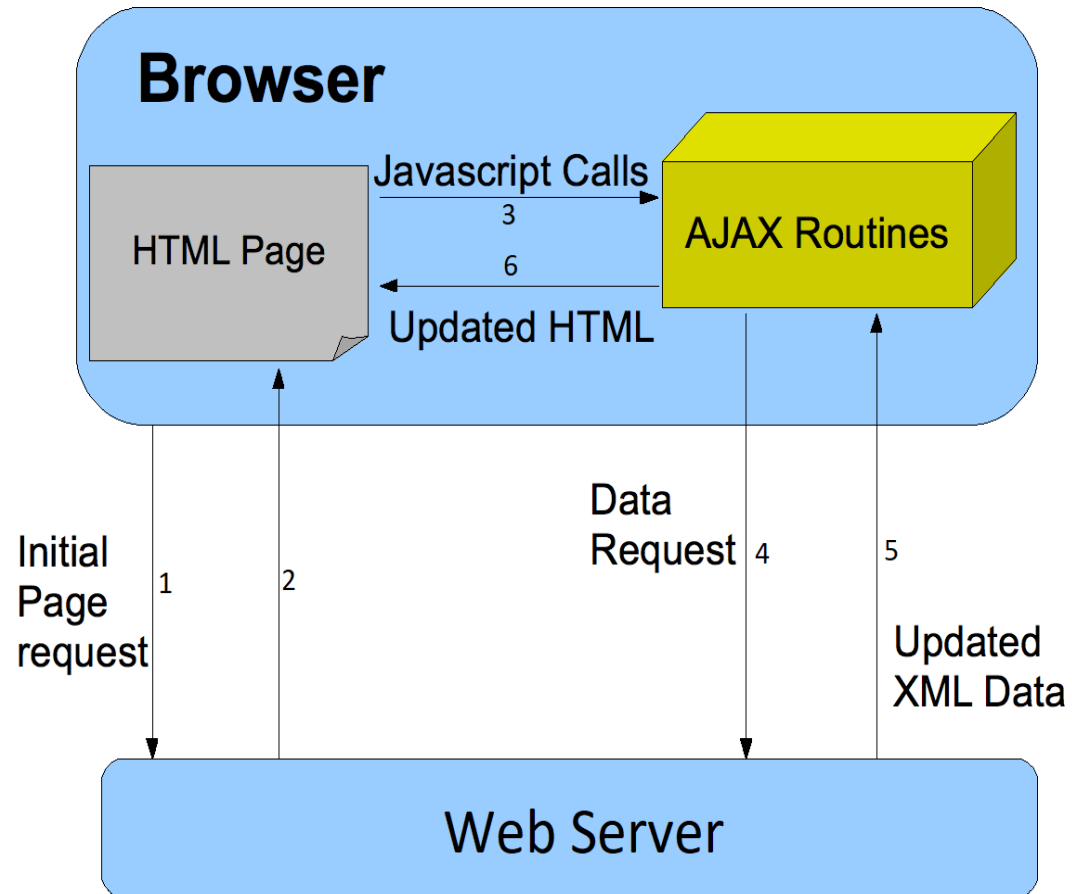
Synchronous:

- Javascript: script untuk melakukan proses.
- Hypertext Markup Language Document Object Model (HTML DOM): digunakan untuk memanipulasi halaman HTML.
- XML / JSON: digunakan sebagai format pertukaran data antara browser dan server. Kebanyakan sekarang menggunakan JSON.
- XMLHttpRequest sebuah class yang digunakan untuk melakukan request ke server.

How to Work?

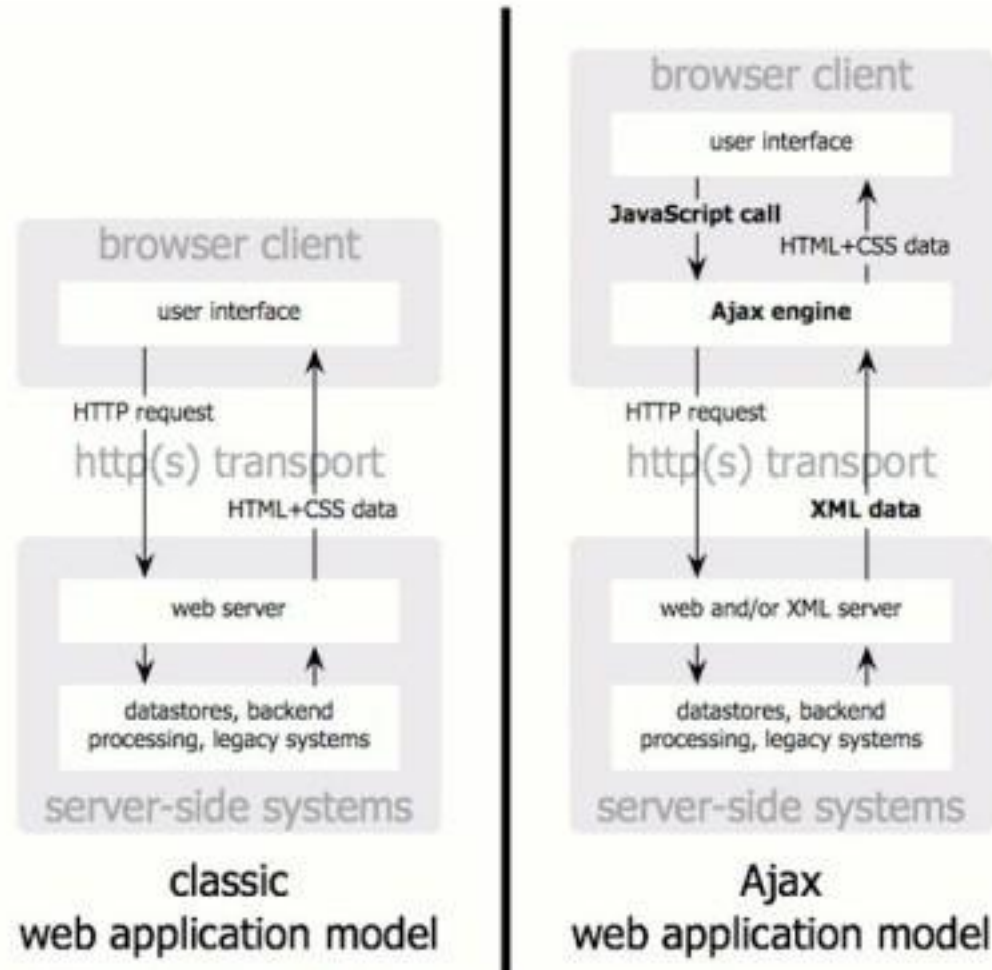


How to Work?



- Web server mengirimkan response berupa file HTML yang didalamnya mengandung rutin/statement AJAX.
- Jika sesuatu terjadi (click, mouseover, keypress) maka akan memanggil rutin AJAX.
- Rutin AJAX akan request ke web server untuk mendapatkan data yang diperlukan. Response dari request ini bisa berasal dari database
- Setelah response tersusun, maka dikirim kembali ke rutin AJAX.
- Setelah menerima response dari web server, maka rutin AJAX akan mengupdate halaman HTML tersebut tanpa harus reload semua.

How to Work? (Continue)



- Dengan XML, suatu halaman web dapat membuat request dan mendapatkan respon dari server web tanpa reload.
- User akan selalu tetap dihalaman sama.
- Tidak perlu menunggu server untuk merespon.

HTML DOM

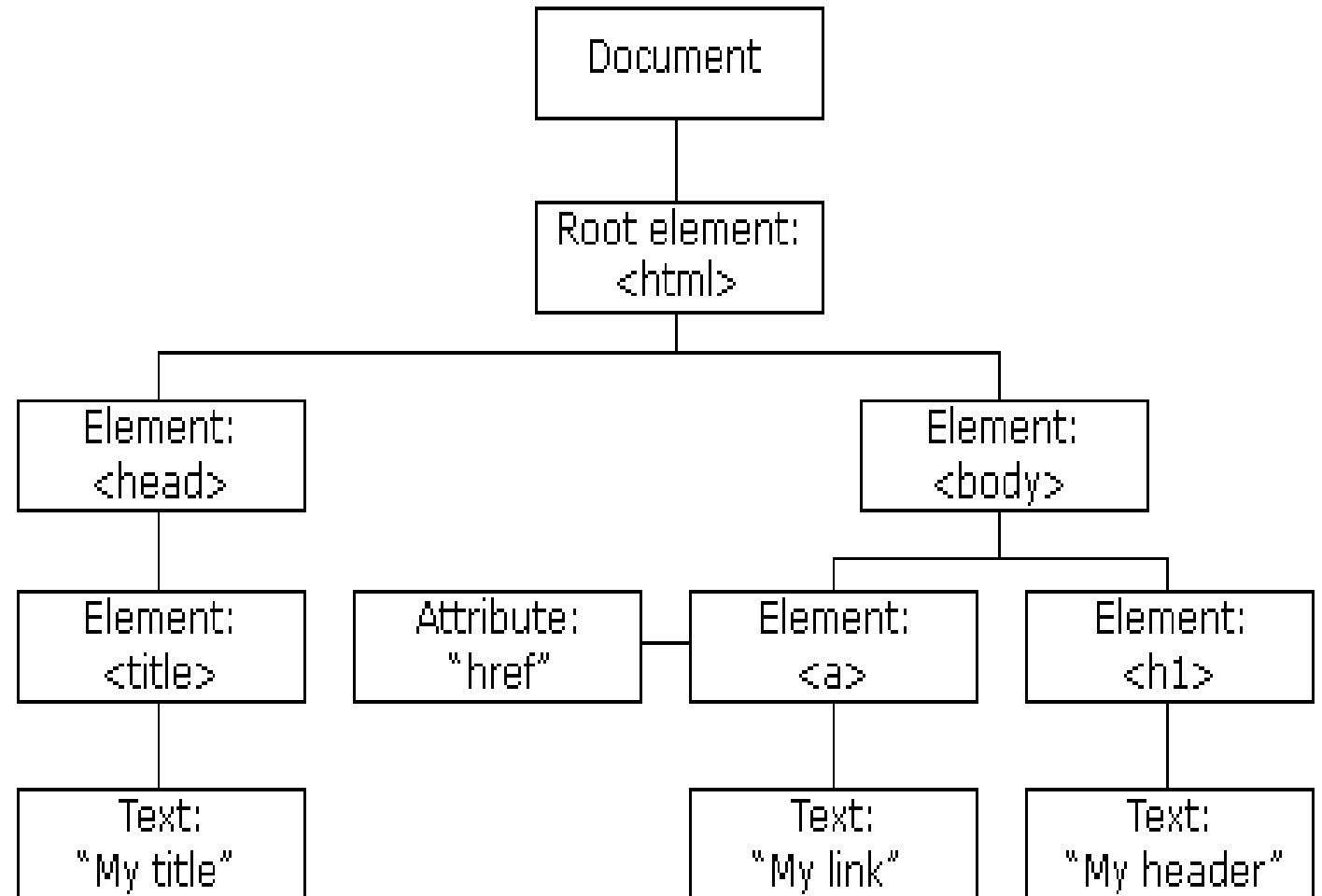


HTML DOM

- HTML DOM (Hypertext Markup Language Document Object Model): sebuah standar untuk mengambil, mengubah, menambah, atau menghapus elemen HTML termasuk attribute/properties, event, dan methodnya.
- Dengan ini kita bisa membuat halaman HTML yang dinamis dengan JavaScript.

HTML DOM (Continue)

- Ketika sebuah halaman HTML dimuat di browser, maka browser akan membuat Document Object Model untuk halaman tersebut.
- Model HTML DOM disusun dalam bentuk struktur tree.



HTML DOM Example

Kasus 1:

- Contoh berikut akan mendemokan penggunaan HTML DOM dengan JavaScript.
- Halaman ini akan menampilkan 3 buah elemen (button, span, a) yang akan memberikan hasil yang berbeda.
- Jika tombol diklik maka akan menampilkan informasi mengenai Bandung.
- Jika elemen span diklik maka akan menampilkan informasi mengenai Jakarta.
- Jika mouse berada di atas elemen a (link) (event mouseover) maka akan menampilkan informasi mengenai Surabaya.

HTML DOM Example (Continue)

Nama file : htmldomjs.html

```
<!DOCTYPE html>
<html>
<head><title>Demo HTML DOM + Javascript</title></head>
<body>
<button onclick="showBandung()">Bandung</button>
<span onclick="showJakarta()" style="border:1px solid black">Jakarta</span>
<a href="#" onmouseover="showSurabaya()">Surabaya</a>
<div id="isi" style="border:1px solid black;font-size:28pt">&nbsp;</div>

function showBandung(){
    var isi=document.getElementById("isi");
    isi.innerHTML="Bandung adalah ibukota Provinsi Jawa Barat<br>";
    isi.style.backgroundColor="blue";
    isi.style.color="white";
}
function showJakarta(){
    var isi=document.getElementById("isi");
    isi.innerHTML="Jakarta adalah ibukota Republik Indonesia<br>";
    isi.style.backgroundColor="red";
    isi.style.color="yellow";
}
function showSurabaya(){
    var isi=document.getElementById("isi");
    isi.innerHTML="Surabaya adalah ibukota Provinsi Jawa Timur<br>";
    isi.style.backgroundColor="green";
    isi.style.color="cyan";
}
</script>
</body>
</html>
```

Container untuk
menampilkan isi

Memanipulasi isi sebuah
elemen

Memanipulasi style/css
sebuah elemen

HTML DOM Example (Continue)

Run : `http://localhost/nama_folder/pertemuan12/htmldomjs.`

html
§ Tampilan awal halaman

Bandung Jakarta Surabaya

§ Setelah mengklik tombol
“Bandung”

Bandung adalah ibukota Provinsi Jawa Barat

§ Setelah memilih span Jakarta

Jakarta adalah ibukota Republik Indonesia

§ Setelah menggerakkan mouse di atas
“Surabaya”

Surabaya adalah ibukota Provinsi Jawa Timur

JSON



JSON

- JSON singkatan dari JavaScript Object Notation (Notasi Objek JavaScript).
- JSON adalah sintak untuk penyimpanan dan pertukaran data.
- JSON adalah teks.
- JavaScript mempunyai function bawaan yang digunakan untuk mengkonversi sebuah string yang ditulis dalam format JSON menjadi objek-objek JavaScript.

JSON: What's the Rules?

- Data disusun berpasangan antara nama data dan isinya (name/value pair). Nama data dan isinya dipisahkan dengan titik dua (:).
- Antar data dipisahkan dengan koma.
- Tanda kurawal ({}) menyatakan objek.
- Tanda kurung siku ([]) menyatakan array.

JSON: Example

Contoh JSON

```
var data={"kelas":"ATOL",  
         "tahun":2017,
```

Array

```
{ "nama": "Aa", "nilai": 90 },  
{ "nama": "Bebe", "nilai": 70 },  
{ "nama": "Cece", "nilai": 85 }  
};
```

Cara mengakses data tersebut

- Jika ingin memanggil property “kelas” gunakan data.kelas atau data["kelas"]
 - Jika ingin memanggil property “ tahun” gunakan data.tahun kelas atau data["tahun"]
 - Jika ingin mengambil objek mahasiswa pertama, gunakan data.mahasiswa[0]
 - wa kedua, gunakan data.mahasiswa[1].nama
- Jika ingin mengambil nama mahasiswa a

JSON: Example (Continue)

Nama File: json.html

```
<!DOCTYPE html>
<html>
<head><title>JSON 1</title></head>
<body>
<script>
function tampilJSON(){
  var data={"kelas":"ATOL",
    "tahun":2017,
    "mahasiswa":[
      {"nama":"Aa","nilai":90},
      {"nama":"Bebe","nilai":70},
      {"nama":"Cece","nilai":85}
    ]
  };
  var isi=document.getElementById("isi"); isi.innerHTML=data.kelas+"<br>"; isi.innerHTML+=data["tahun"]+"<br>";
  isi.innerHTML+="Peserta Kelas : "+data.mahasiswa.length+"<br>"; for(var i=0;i<data.mahasiswa.length;i++){
  isi.innerHTML+=data.mahasiswa[i].nama+" dengan nilai "+
    data.mahasiswa[i].nilai+"<br>";
  }
}
</script>
<body onload="tampilJSON()">
<div id="isi"></div>
</body>
</body>
</html>
```

JSON: Example (Continue)

- Data disusun berpasangan antara nama data dan isinya (name/value pair). Nama data dan isinya dipisahkan dengan titik dua (:).
- Antar data dipisahkan dengan koma.
- Tanda kurawal ({}) menyatakan objek.
- Tanda kurung siku ([]) menyatakan array.

Run File :

http://localhost/nama_folder/pertemuan12/json.html

ATOL

2017

Peserta Kelas : 3

Aa dengan nilai 90

Bebe dengan nilai 70

Cece dengan nilai 85

XMLHttpRequest



XMLHttpRequest

- XMLHttpRequest sebuah class yang digunakan untuk melakukan request ke server. Untuk melakukan request ke server, buatlah object dari class tersebut.
- Walau pun namanya XMLHttpRequest, object ini bisa mereturnkan/menghasilkan response dengan banyak tipe, tidak hanya xml.
- Seperti objek-objek yang lain, objek XMLHttpRequest juga memiliki property dan method.

XMLHttpRequest: Property

Properti yang penting dari object XMLHttpRequest adalah:

- onreadystatechange : Diisi dengan sebuah function yang akan dipanggil (callback) ketika property readyState berubah.
- readyState : Menyimpan status dari XMLHttpRequest. Properti ini mempunyai nilai 0 s.d 4 yang menyatakan urutan request ke server, dimana :
 - § 0 : Request belum diinisialisasi
 - § 1 : Koneksi ke server berhasil
 - § 2 : Request telah diterima
 - § 3 : Request sedang diproses
 - § 4 : Request selesai dan response/balasan telah siap dipakai
- status : angka yang menyatakan status request. Seperti 200, 403, 404, dan lain-lain
- statusText : string yang menjelaskan property status seperti "OK" (status=200), "Forbidden" (status=403), "Page Not Found" (status=404) dan lain-lain.

XMLHttpRequest: Method

Method yang penting dari object XMLHttpRequest adalah :

- § `open(method, url, async)` : Digunakan untuk menentukan jenis request yang akan dilakukan. Parameternya :
 - § Method: tipe request, misalny : GET atau POST
 - § url: lokasi file yang direquest
 - § async: isi true kalau ingin melakukan asynchronous request.
- § `send()` : Kirim request (untuk method GET)
- § `send(string)` : Kirim request (untuk method POST). String berisi data yang akan dikirim.
- § `setRequestHeader(nama_header, isi)` : Digunakan jika ingin mengirimkan informasi HTTP header.

XMLHttpRequest: Response Server

- Ketika readyState request bernilai 4 dan status request bernilai 200 maka browser akan menerima response dari server.
- Properti dari response adalah
 - responseText berisi data response dalam bentuk string
 - responseXML berisi data response dalam bentuk XML

XMLHttpRequest: Example

Contoh Kasus :

- § Buatlah sebuah halaman yang menampilkan informasi mengenai 3 buah kota di Indonesia (Bandung, Jakarta dan Surabaya).
- § Ketiga kota tersebut memiliki informasi yang tersimpan pada file yang terpisah, misalnya kota Bandung informasinya tersimpan di server dengan nama bandung.txt. Begitu juga dengan yang lainnya.
- § Jika tombol kota diklik, maka javascript akan merequest data ke server sesuai nama kota yang
 - sedang aktif dan akan menampilkan response di container “isi”.

XMLHttpRequest: Example (Continue)

Isi file bandung.txt

Kota Bandung merupakan kota metropolitan terbesar di Provinsi Jawa Barat, sekaligus menjadi ibu kota provinsi tersebut. Kota ini terletak 140 km sebelah tenggara Jakarta, dan merupakan kota terbesar di wilayah Pulau Jawa bagian selatan. Sedangkan wilayah Bandung Raya (Wilayah Metropolitan Bandung) merupakan metropolitan terbesar ketiga di Indonesia setelah Jabodetabek dan Gerbangkertosusila.

Isi file jakarta.txt

Daerah Khusus Ibukota Jakarta (DKI Jakarta) adalah ibu kota negara dan kota terbesar di Indonesia. Jakarta merupakan satu-satunya kota di Indonesia yang memiliki status setingkat provinsi. Jakarta terletak di pesisir bagian barat laut Pulau Jawa. Dahulu pernah dikenal dengan beberapa nama di antaranya Sunda Kelapa, Jayakarta, dan Batavia. Di dunia internasional Jakarta juga mempunyai julukan J-Town, atau lebih populer lagi The Big Durian karena dianggap kota yang sebanding New York City (Big Apple) di Indonesia.

Isi file Surabaya.txt

Kota Surabaya adalah ibu kota Provinsi Jawa Timur, Indonesia, sekaligus kota metropolitan terbesar di provinsi tersebut. Surabaya merupakan kota terbesar kedua di Indonesia setelah Jakarta. Kota ini terletak 796 km sebelah timur Jakarta, atau 415 km sebelah barat laut Denpasar, Bali. Surabaya terletak di pantai utara Pulau Jawa bagian timur dan berhadapan dengan Selat Madura serta Laut Jawa.

AJAX & PHP



AJAX & PHP

- AJAX + PHP : Menggabungkan keuntungan dari kedua jenis pemrograman.
- AJAX (baca informasi mengenai Javascript) sulit untuk terkoneksi ke data yang dari database, oleh karena itu AJAX akan melakukan request ke file PHP, dimana file PHP akan melakukan koneksi ke database (semua operasi “SELECT”, “INSERT”, “UPDATE”, “DELETE” dll). Bukan hanya melakukan request ke database. Semua yang bisa dilakukan bisa direquest dari AJAX.

AJAX & PHP Example: Method Load()

Metode load() adalah metode yang tersedia di dalam library jQuery yang digunakan untuk memuat data dari server yang kemudian dapat ditempatkan pada elemen tertentu di halaman website.

Bentuk umum:

```
$(selector).load(URL,data,callback);
```

AJAX & PHP Example: Method Load() (Continue)

Langkah 1: Buat file **demo.php**.

File demo.php berfungsi sebagai file yang akan di muat dengan AJAX menggunakan metode load().

```
<?php
    echo "Info ini dimuat dari file demo.php";
?>
```


Method Load

Langkah 2: Buat file **index.php**.

- File **index.php** berfungsi sebagai file utama yang akan memuat informasi yang di request menggunakan metode **load()** pada file **demo.php**.
- Atribut ID ditandai tanda # dan juga . (titik) sebagai selector.
- Elemen yang digunakan yaitu **<div>**

```
<!DOCTYPE
html>

<html>

  <head>

    <script type="text/javascript" src="jquery-
3.5.1.min.js"></script>

  </head>

  <body>

    <div id='tampil'>Informasi akan di.tampil.kan
disini.</div>

    <button id="btn_tampil">Tampilkan</button>

    <script>
      $(document).ready(function(){
        $('#btn_tampil').click(function(){
          $('#tampil').load('demo.php');
        })
      });
    </script>

  </body>

</html>
```

AJAX & PHP Example: Method Get, Post, & Ajax

- Metode GET(), POST(), dan AJAX() merupakan metode untuk meminta data dari sumber lain.
- Metode GET() digunakan untuk mendapatkan data dari server dan mengembalikan data yang di cache.
- Metode POST() digunakan untuk mendapatkan data dari beberapa server, namun tidak menyimpan data.
- Metode AJAX() digunakan lebih kompleks.

AJAX & PHP Example: Method Get

- Metode `get()` akan meminta data dari server dengan jenis HTTP GET dengan bentuk umum sebagai berikut:

```
$.get (URL, callback);
```

- Terdapat dua parameter yaitu URL dan callback.
- Parameter URL untuk menentukan URL yang diminta.
- Parameter callback untuk mengeksekusi permintaan.

AJAX & PHP Example: Method Get (Continue)

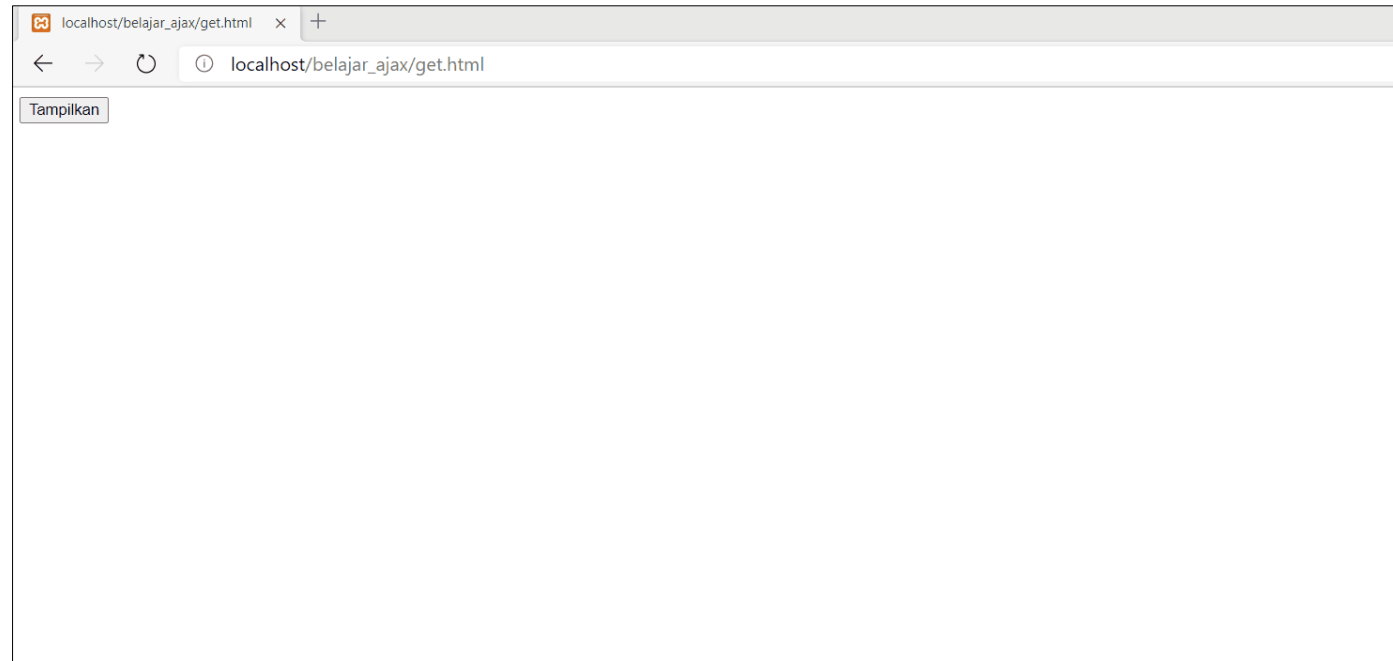
Langkh 1: buat file **get.html**

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery-3.5.1.min.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <button id="btn_tampil">Tampilkan</button>
    <script>
      $(document).ready(function(){
        $('#btn_tampil').click(function(){
          $.get('info.php',function(data, status){
            alert('Data : '+data+'\nStatus: '+status);
          });
        });
      });
    </script>
  </body>
</html>
```

AJAX & PHP Example: Method Get (Continue)

Langkh 2: buat file baru dengan nama **info.php**

```
<?php  
    echo "Info ini dimuat dari file info.php";  
?>
```



AJAX & PHP Example: Method Post

- Bentuk umum penggunaan metode post() adalah sebagai berikut:

```
$.post(URL,data,callback);
```

- Pada metode post sedikit ada tambahan pada parameter opsional dimana kita dapat menentukan beberapa data yang akan dikirim bersamaan saat permintaan (request) dilakukan.

AJAX & PHP Example: Method Post (Continue)

Langkah 1: buatlah file **post.html**

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="jquery-3.5.1.min.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <button id="btn_tampil">Tampilkan</button>
    <script>
      $(document).ready(function(){
        $('#btn_tampil').click(function(){
          $.post('info.php',{nama : 'Setiawan Dimas', alamat : 'Jl
Panjaitan no 88'},
          function(data, status){
            alert('Data : '+data+'\nStatus: '+status);
          });
        });
      });
    </script>
  </body>
</html>
```


Method P

Langkah 2: buatlah file info2.php

```
<?php
```

```
$nama=$_POST['nama']
```

```
$alamat=$_POST['alamat'];
```

```
echo "Nama : " . $nama;
```

```
echo "Alamat : " . $alamat;
```

[S] localhost/belajar_ajax/post.html X +

0 0 localhost/belajar_ajax/post.html

Tampilkan

AJAX & PHP Example: Use of AJAX with Form

Langkah 1: buatlah database dengan nama **belajar_ajax** dan nama table **mahasiswa**;

```
create database belajar_ajax;  
  
create table mahasiswa (  
    nik char(10) NOT NULL PRIMARY KEY,  
    nama VARCHAR(50) NOT NULL,  
    jurusan char(2) NOT NULL  
);
```

AJAX & PHP Example: Use of AJAX with Form

Langkah 2: membuat file **koneksi.php**

```
<?php
$host="localhost";
$user="root";
$password="";
$db="belajar_ajax";

$kon = mysqli_connect($host,$user,$password,$db);
if (!$kon){
    die("Koneksi gagal:".mysqli_connect_error());
}
?>
```

AJAX & PHP

Example: Use of AJAX with Form

Langkah 3: membuat file
index2.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <script type="text/javascript" src="jquery-3.5.1.min.js"></script>
5  </head>
6  <body>
7      <form method="post" id="form_mahasiswa">
8          <label>NIK </label>
9          <input type="number" name="nik"><br>
10         <label>Nama </label>
11         <input type="text" name="nama"><br>
12         <label>Jurusan</label>
13         <select name="jurusan">
14             <option value="TI">Teknik Informatika</option>
15             <option value="TK">Teknik Komputer</option>
16             <option value="SI">Sistem Informasi</option>
17             <option value="KA">Komputerisasi Akuntansi</option>
18             <option value="MI">Manajemen Informatika</option>
19         </select>
20     </form>
21     <button id="btn_tampil">Tampilkan</button>
22     <hr>
23     <div id="tampil_data"></div>
24     <script>
25         $(document).ready(function(){
26             $('#tampil_data').load("tampil.php");
27             $('#btn_tampil').click(function(){
28                 var data = $('#form_mahasiswa').serialize();
29                 $.ajax({
30                     type : 'POST',
31                     url : "tambah.php",
32                     data: data,
33                     cache : false,
```

AJAX & PHP

Example: Use of AJAX with Form (Continue)

File index2.html berfungsi sebagai halaman utama untuk menampilkan form.

```
30         type : 'POST',
31         url : "tambah.php",
32         data: data,
33         cache : false,
34         success : function(data){
35             $('#tampil_data').load("tampil.php");
36         }
37     });
38 });
39 });
40 </script>
41 </body>
42 </html>
```

AJAX & PHP

Example: Use of AJAX with Form (Continue)

- Langkah 4: membuat file **tambah.php**
- Fungsi file tambah.php adalah menerima data yang dikirim dari form ajax yang selanjutnya kita insert kedalam tabel mahasiswa.

```
1 <?php
2 //Include file koneksi ke database
3 include "koneksi.php";
4 //menerima nilai dari kiriman form
5 $nik=$_POST["nik"];
6 $nama=$_POST["nama"];
7 $jurusan=$_POST["jurusan"];
8
9 //Query input menginput data kedalam tabel mahasiswa
10 $sql="insert into mahasiswa (nik,nama,jurusan) values('$nik','$nama','$jurusan')";
11
12 //Mengeksekusi/menjalankan query diatas
13 $hasil=mysqli_query($kon,$sql);
14
15 ?>
```


AJAX & PHP

Example: Use of AJAX with Form (Continue)

Fungsi file tambah.php adalah menerima data yang dikirim dari form ajax yang selanjutnya kita insert kedalam tabel mahasiswa.

```
1 <?php
2 //Include file koneksi ke database
3 include "koneksi.php";
4 //menerima nilai dari kiriman form
5 $nik=$_POST["nik"];
6 $nama=$_POST["nama"];
7 $jurusan=$_POST["jurusan"];
8
9 //Query input menginput data kedalam tabel mahasiswa
10 $sql="insert into mahasiswa (nik,nama,jurusan) values('$nik','$nama','$jurusan')";
11
12 //Mengeksekusi/menjalankan query diatas
13 $hasil=mysqli_query($kon,$sql);
14
15 ?>
```


AJAX & PHP

Example: Use of AJAX with Form (Continue)

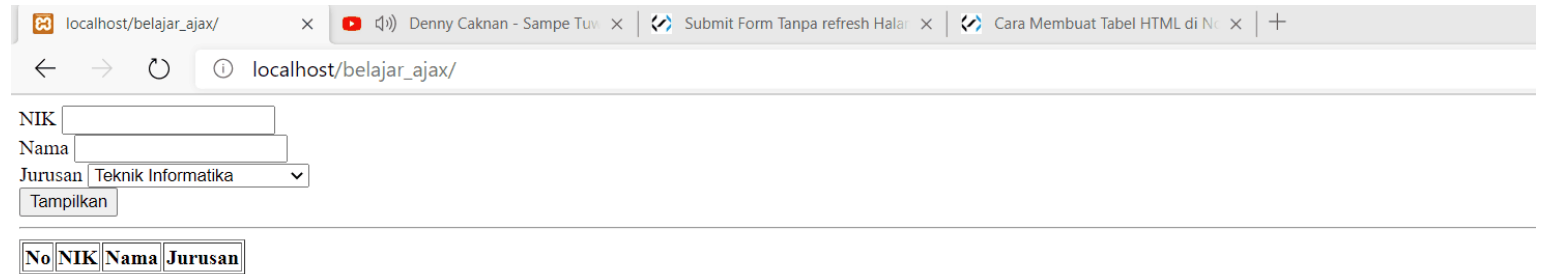
- Langkah 6: membuat file **tampil.php**
- File ini kita tampilkan menggunakan metode **load()**.

```
1 <table border="1">
2   <tr>
3     <th>No</th>
4     <th>NIK</th>
5     <th>Nama</th>
6     <th>Jurusan</th>
7   </tr>
8   <?php
9     include "koneksi.php";
10    $hasil=mysqli_query($kon, query: "select * from mahasiswa order by nik desc");
11    $no=0;
12    while ($data = mysqli_fetch_array($hasil)):
13      $no++;
14      ?>
15      <tr>
16        <td><?php echo $no; ?></td>
17        <td><?php echo $data['nik']; ?></td>
18        <td><?php echo $data['nama']; ?></td>
19        <td><?php echo $data['jurusan']; ?></td>
20      </tr>
21    <?php endwhile;?>
22  </table>
```

AJAX & PHP

Example: Use of AJAX with Form (Continue)

- Langkah 7: uji coba



The screenshot shows a web browser window with the address bar displaying 'localhost/belajar_ajax/'. The page contains a form with the following fields:

- NIK:
- Nama:
- Jurusan:
- Tampilkan:

Below the form, there is a table with the following structure:

No	NIK	Nama	Jurusan

References

1. [Nixon, R. \(2005\). Learning PHP, MySQL & JavaScript. In Software World \(Vol. 36, Issue 6\).](#)
2. <https://www.php.net/docs.php>
3. <https://www.petanikode.com/tutorial-php-mysql/>
4. Desrizal. (2017). PHP Ajax Javascript jQuery Tutorial Indonesia. 1–73. <http://blog.codingwear.com>
5. https://www.w3schools.com/js/js_htmlDOM.asp
6. <https://www.w3schools.com/jsref/default.asp>



THANK YOU



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

**FAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA**

Pemrograman Web

Pertemuan 13

PHP Framework MVC

Zuhri Halim, S.Kom., M.Kom.

Outline

1. PHP Framework
2. MVC Concept
3. CodeIgniter



Introduction: Learning Outcomes

Mahasiswa mampu menjelaskan framework MVC berbasis PHP

Mahasiswa mampu menganalisis script MVC berbasis PHP untuk proses manipulasi data

PHP Framework



PHP Framework

- Frameworks adalah kumpulan dari file-file pustaka (libraries/program) atau class-class yang mendukung dalam pengembangan aplikasi secara terstruktur dan independent terhadap aplikasi.
- Software Framework adalah sebuah desain yang bisa digunakan berulang-ulang (re-usable design) untuk sebuah sistem atau sub sistem piranti lunak

PHP Framework: The Rules

Frameworks (kerangka kerja) digunakan untuk membangun aplikasi, dimana aturan-aturan dalam proses pengembangan aplikasi mengikuti ketentuan yang ada pada framework.

Aturan-aturan dalam Framework meliputi :

- Kode Standard
- Konfigurasi program
- Alur kerja

PHP Framework

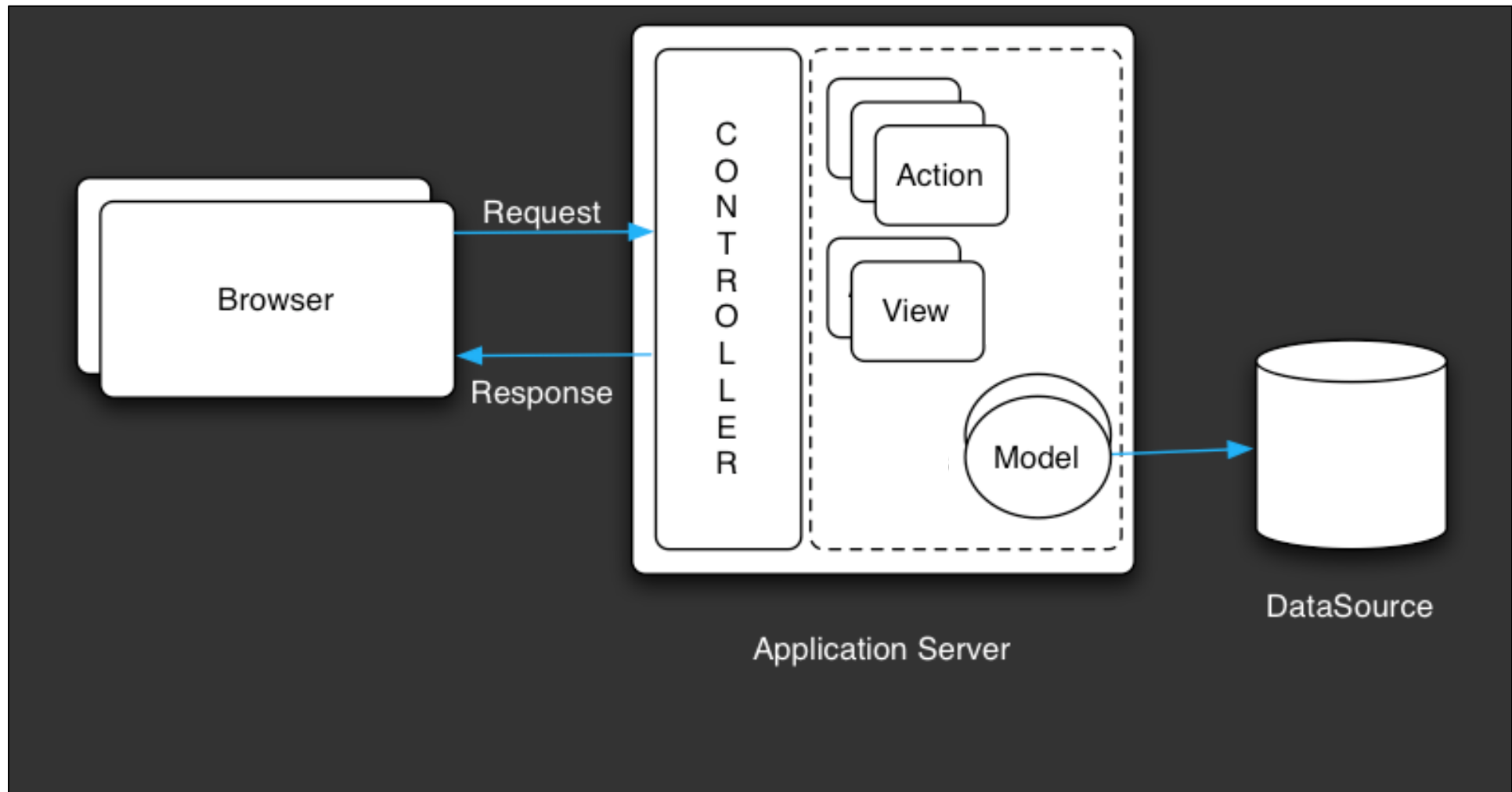
- CakePHP
- CodeIgniter
- SeaGull Framework
- FuseBox
- ZendFramework
- Akelos
- Symfony
- WACT
- Prado
- ZooP Framework
- Laravel
- Yii Framework

Arsitektur Desain Aplikasi : MVC Framework

Model-View-Controller (MVC) Concept



MVC Concept: The Architecture



MVC Concept: Model

- Model independent terhadap view dan controller.
- Membungkus data dan fungsionalitas domain logic dari core aplikasi.
- Model secara pasif mensuplai data ke aplikasi dan mewakili logika bisnis proses aplikasi.
- Domain Logic: mewakili perhitungan dan penyimpanan data (table pada database) dari form aplikasi, yang diorganisasikan oleh domain model dan skrip program.
- Domain Model: mengidentifikasi dan menjelaskan alur informasi, struktur data dan constraint di dalamnya.

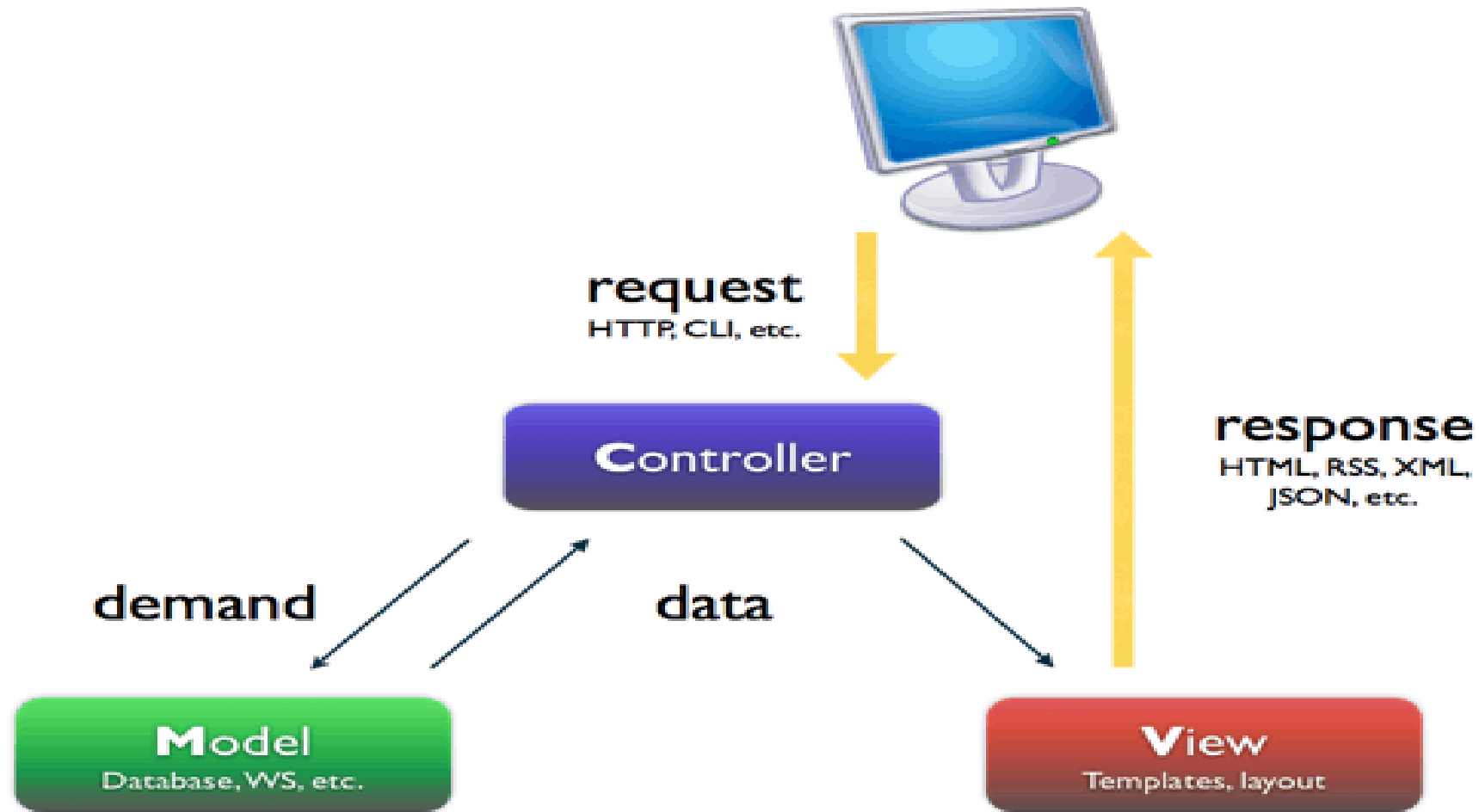
MVC Concept: View

- Menampilkan data dari yang diperoleh oleh controller dari Model ke hadapan user.
- Menampilkan output dari aplikasi.
- Menampilkan data bersifat read only dari model dengan menggunakan fungsi query yang disediakan oleh Model.
- Menampilkan data berupa format HTML atau XML ke user.

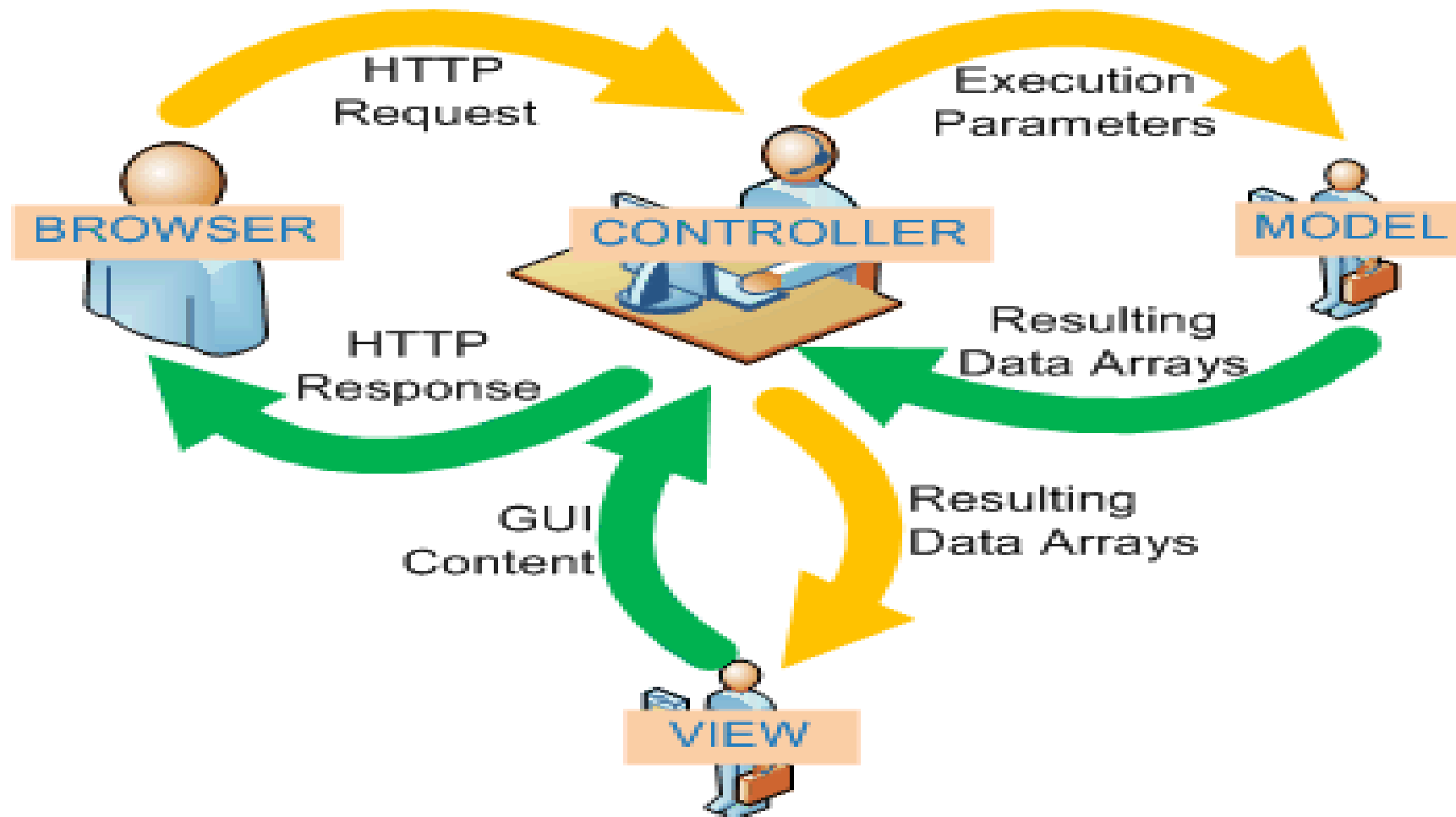
MVC Concept: Controller

- Sebagai entry point semua request user atau inputan ke aplikasi.
- Controller menerima inputan dari user, memproses dan menindak lanjuti terhubung ke Model atau View.
- Controller bertanggung jawab memanggil method-method pada Model yang bisa merubah nilai dari object (model).

MVC Concept (Continue)



MVC Concept (Continue)



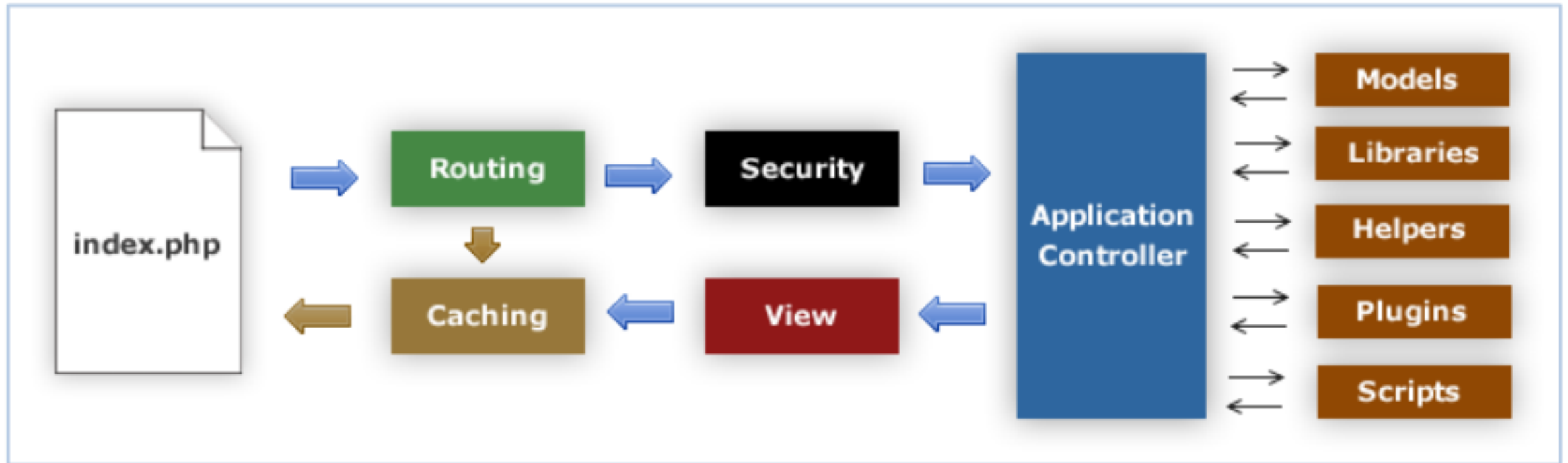
CodeIgniter



CodeIgniter

- Sebuah open source web app framework, yang digunakan untuk membangun sebuah web dinamis dengan bahasa PHP.
- Dibuat oleh Rick Ellis (CEO of EllisLab, Inc.).
- Release 28 Feb 2006 oleh Ellislab, Inc.
- Dapat digunakan pada semua operating system.

CodeIgniter: The Routes

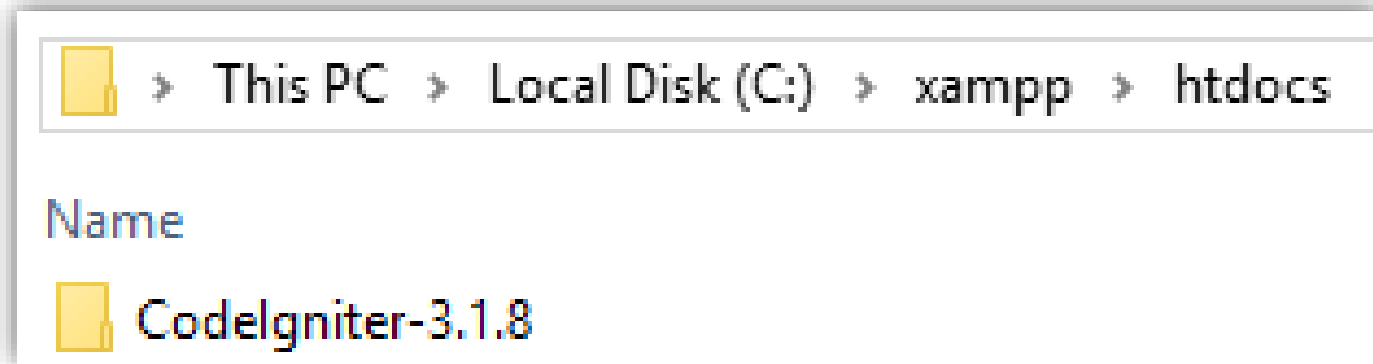


CodeIgniter: The Preparation

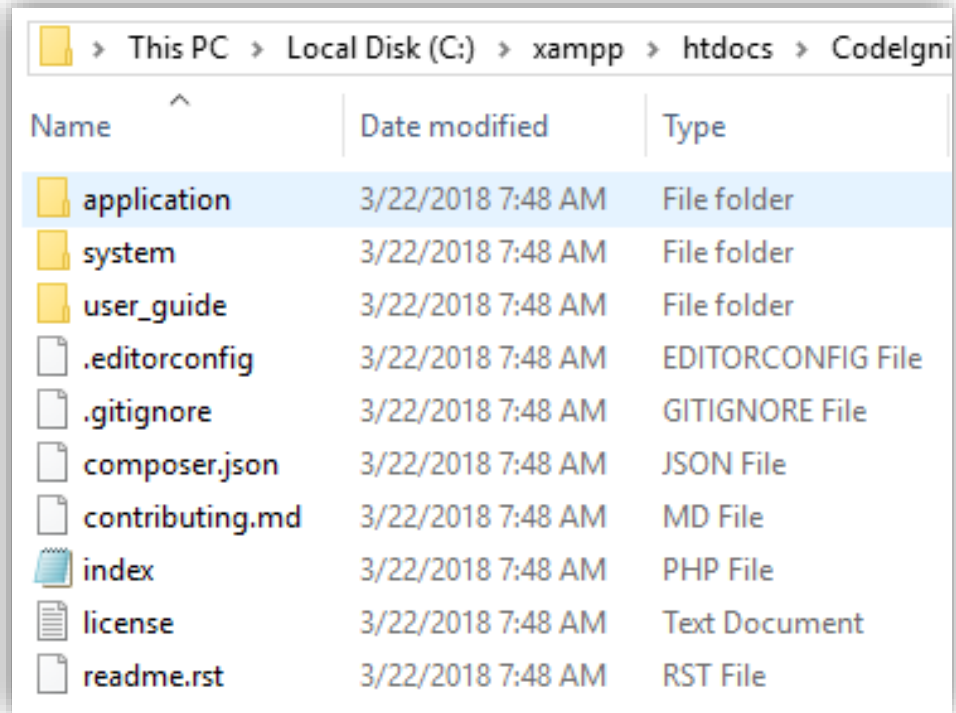
- Software Editor
- Web Browser
- Web Server: PHP, MySQL, Phpmyadmin (XAMPP, dll)
- Composer
- File Project CodeIgniter versi 3.x.x

CodeIgniter: The Structure

- Download Codeigniter di <http://codeigniter.com/>
- Extract di c:/xampp/htdocs



CodeIgniter: The Structure (Continue)



- **Folder application :**

Untuk membangun aplikasi kita secara dinamis

- **Folder system :**

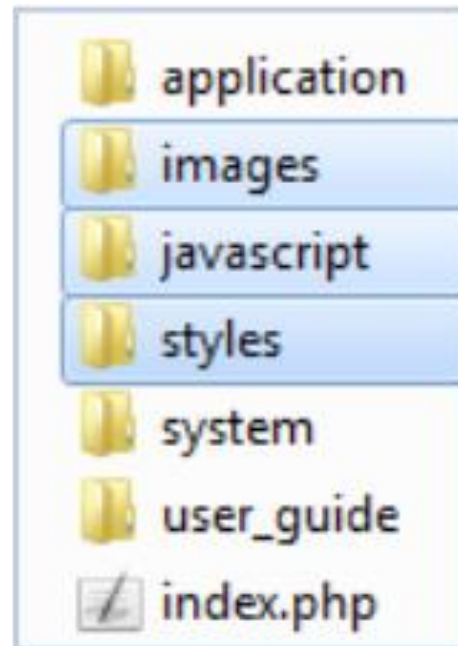
Berisi libraries framework yang dibutuhkan untuk membangun app.

- **Folder user_guide :**

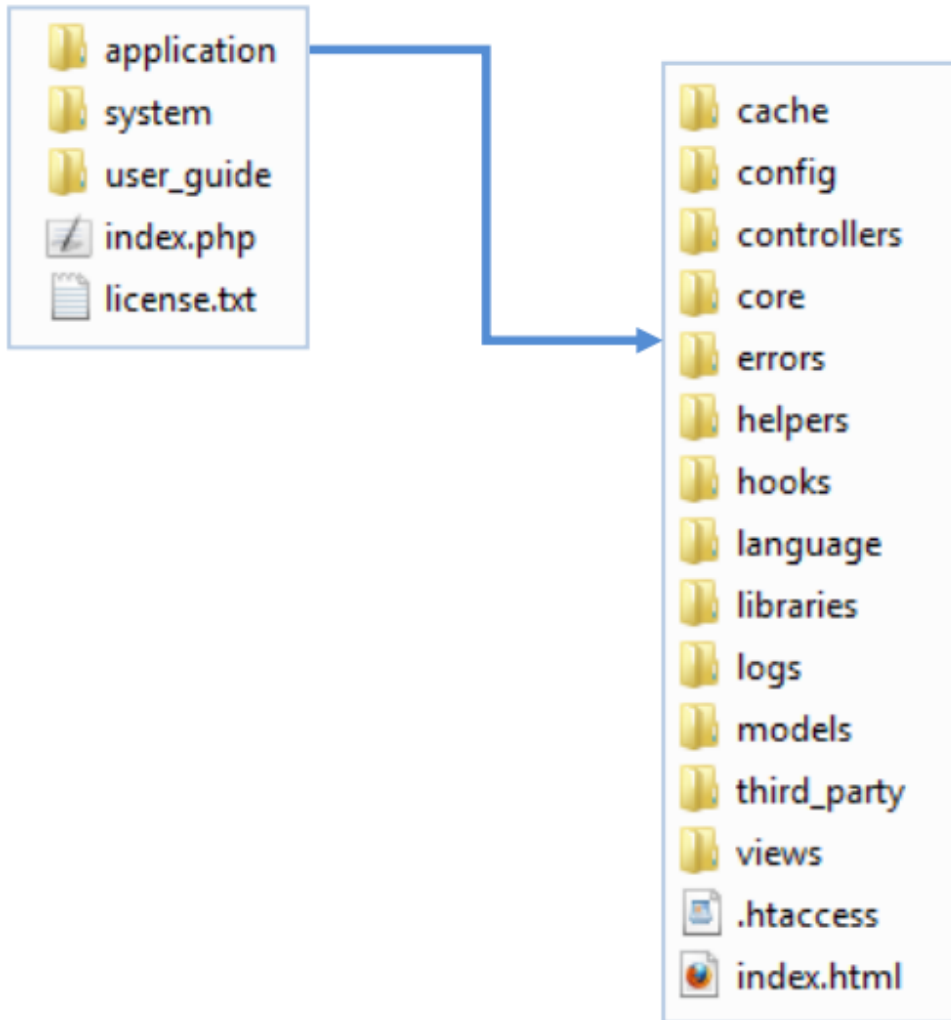
Berisi dokumentasi framework, untuk mempermudah pembelajaran kita.

CodeIgniter: The Structure (Continue)

- Aplikasi yang akan Anda bangun tentunya menggunakan JavaScript, gambar dan CSS.
- Anda dapat buat direktori untuk masing-masing sumberdaya tersebut pada root aplikasi (setingkat dengan file index.php), sehingga seperti contoh berikut:



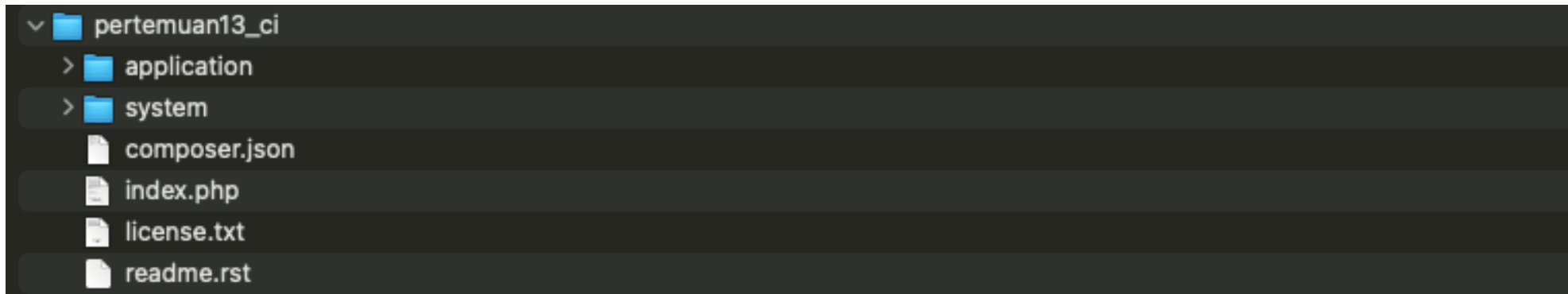
CodeIgniter: The Structure (Continue)



- Config: terkait konfigurasi dasar aplikasi.
- Contrrolers: berfisi file yang terkit dengan logic.
- Core: berisi dari class inti CodeIgniter.
- Helpers: berisi file helper yang dibuat sendiri.
- Libraries: berisi file library/function yang dibuat sendiri.
- Logs: berisi file error log dari aplikasi.
- Models: berisi file yang terkait database.
- View: berrisi file yang terkait interface.

CodeIgniter: The Installations

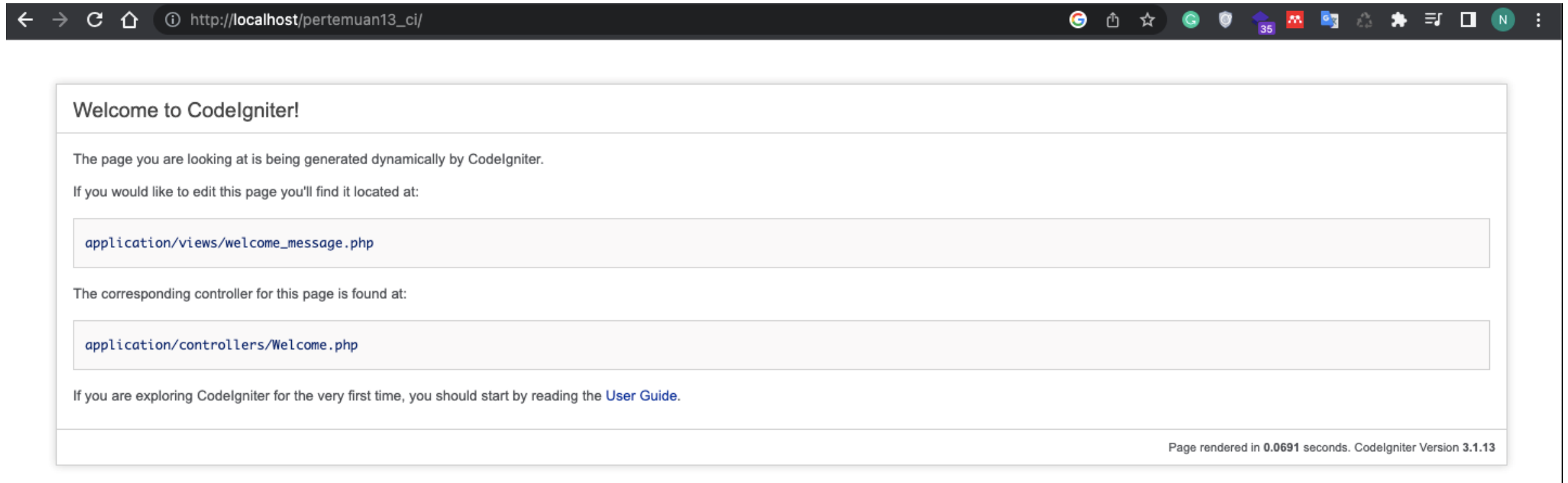
Extract file CodeIgniter yang sudah di-download sebelumnya ke direktori htdocs, kemudian ganti nama folder **CodeIgniter_3.x.x** tersebut dengan **pertemuan13_ci** Anda yang selanjutnya akan menjadi direktori kerja Anda, sehingga seperti ini:



Untuk menguji keberhasilan instalasi, Anda dapat mengetik URL di browser seperti ini **http://localhost/pertemuan13_ci**

CodeIgniter: The Installations (Continue)

Jika berhasil, maka akan muncul gambar seperti ini.



CodeIgniter: The Configurations

Terdapat beberapa konfigurasi awal untuk membangun web dengan CodeIgniter, antara lain sebagai berikut:

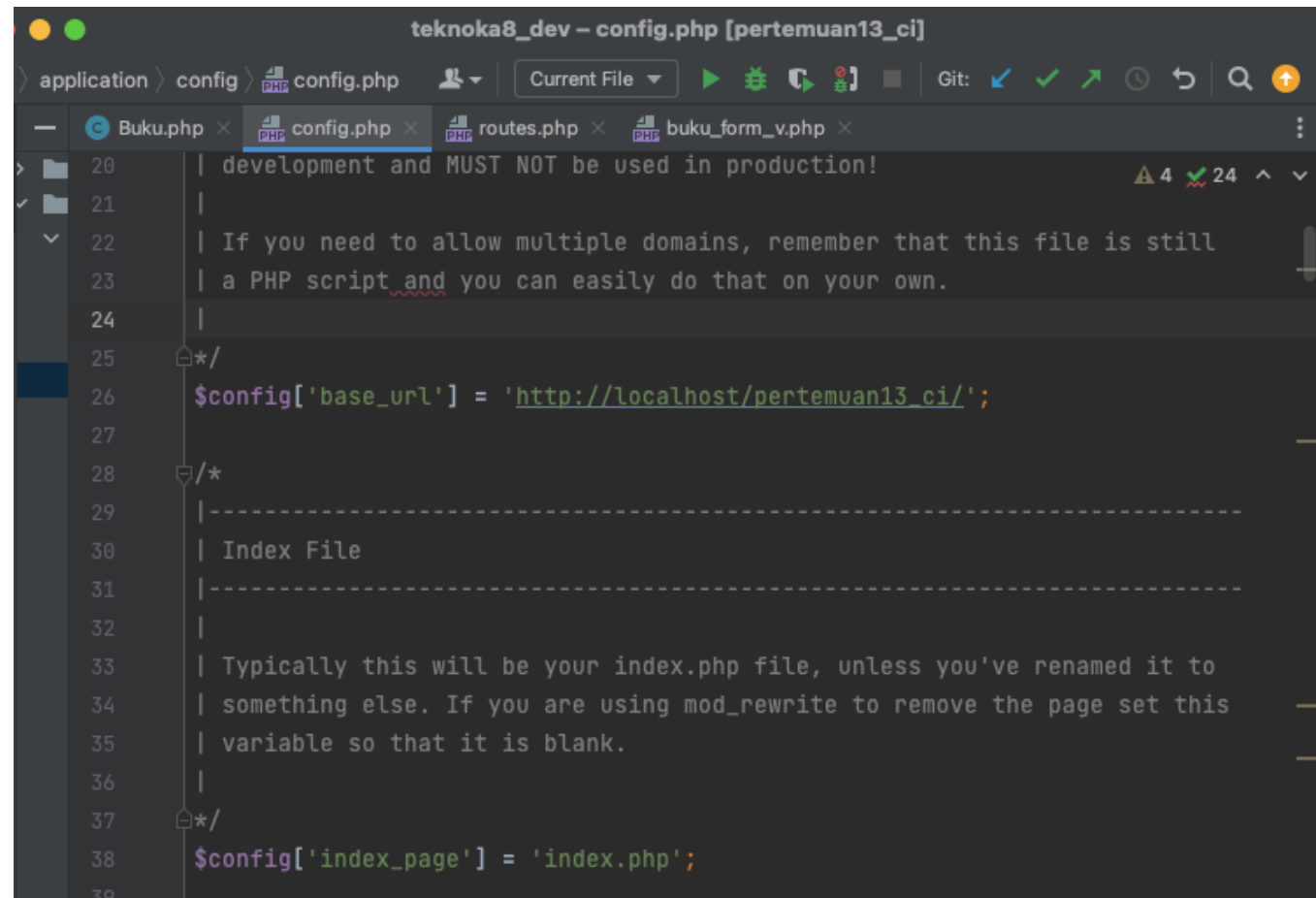
- `index.php`: tidak ada konfigurasi yang diubah pada file ini. Namun, jika ingin masuk ke production/deploy, maka harus dimatikan.
- `autoload.php`: lokasi file ada di `application/config/`. Di CI dapat memanggil fungsi tanpa harus membuat deklarasi, yang dimana dapat otomatis dipanggil adalah:
 - Packages
 - Libraries
 - Helper
 - Custom config, jika Anda membuat file konfigurasi diluar file default CodeIgniter.
 - Language
 - Model.

CodeIgniter: The Configurations (Continue)

- config.php: berfungsi untuk mengkonfigurasi sesuatu yang dibutuhkan.
- XSS filtering: mendefinisikan data yang dikirim oleh browser/pengguna (POST, GET, COOKIE).
- CSRF Protection: data yang dikirim dipastikan masuk ke SESSION atau tidak.
- database.php: file yang berfungsi untuk mengkoneksikan aplikasi ke database.
- router.php: berfungsi untuk mengatur URL yang dapat diatur secara spesifik.

CodeIgniter: The Practice 1

- Rename folder CodeIgniter_3.1.8 dengan nama yg lebih mudah. Misal, nama **NIM** Anda atau yang lainnya: 13121121.
- Masuk ke folder application -> config, ada 2 buah file yg harus di setting terlebih dahulu : config.php & routes.php



```
teknoka8_dev – config.php [pertemuan13_ci]
> application > config > config.php
Buku.php x config.php x routes.php x buku_form_v.php x
20 | development and MUST NOT be used in production!
21 |
22 | If you need to allow multiple domains, remember that this file is still
23 | a PHP script and you can easily do that on your own.
24 |
25 | */
26 | $config['base_url'] = 'http://localhost/pertemuan13_ci/';
27 |
28 | /*
29 | -----
30 | Index File
31 | -----
32 |
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 | */
38 | $config['index_page'] = 'index.php';
39 |
```

CodeIgniter: The Practice 1 (Continue)

- Setting controller pertama yang akan dipanggil oleh aplikasi (load default), di routes.php. Nama controller anda bebas. Misal : halo

```
40
41 $route['default_controller'] = "halo";
42 $route['404_override'] = '';
43
44
45 /* End of file routes.php */
46 /* Location: ./application/config/routes.
```

CodeIgniter: The Practice 1 (Continue)

- Masuk ke folder application -> controller
- Buat sebuah file php bernama **Halo.php**
- Edit isinya sebagai berikut:

```
1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3  class Halo extends CI_Controller {
4
5      public function index()
6      {
7          $this->load->view('hello_world');
8      }
9  }
10
```

CodeIgniter: The Practice 1 (Continue)

- Pada controller Welcome.php kita memiliki view (tampilan) dengan perintah :
`$this->load->view('hello_world');`
- Maka kita harus membuat sebuah file bernama **hello_world.php** dalam folder application -> view.

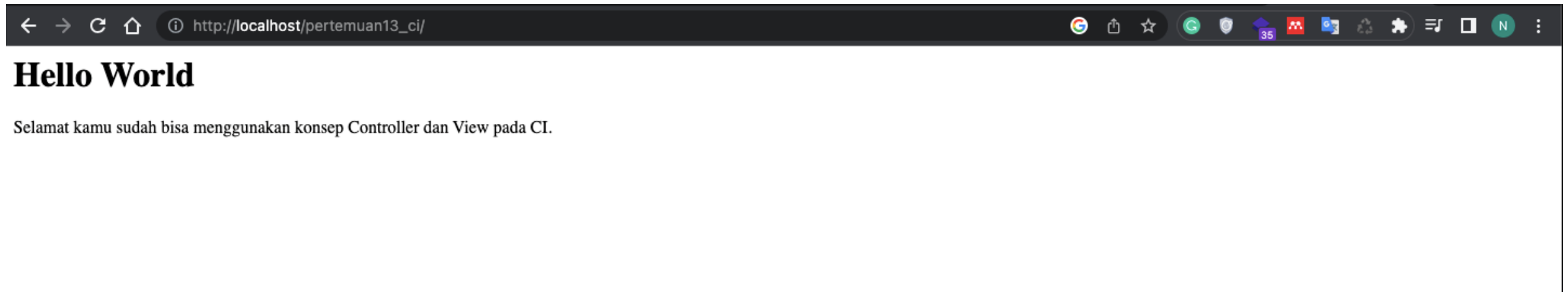
CodeIgniter: The Practice 1 (Continue)

- Setelah file view **hello_world.php** dibuat maka isi sebagai berikut :

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>My First CI App</title>
6  </head>
7  <body>
8
9      <div id="container">
10         <h1>Hello World</h1>
11
12         <div id="body">
13             <p>Selamat kamu sudah bisa menggunakan konsep Controller dan View pada CI</p>
14         </div>
15     </div>
16
17 </body>
18 </html>
```


CodeIgniter: The Practice 1 (Continue)

Run app Anda di browser.



References

1. [Nixon, R. \(2005\). Learning PHP, MySQL & JavaScript. In Software World \(Vol. 36, Issue 6\).](#)
2. <https://www.php.net/docs.php>
3. <https://www.codeigniter.com/>

THANK YOU