

# Pemrograman Web Part 1

*Pertemuan 6*

*MK Algoritma Pemrograman II*

**Ika Qutsiati Utami, S.Kom., M.Sc.**

**M. N. Fakhruzzaman, S.Kom., M.Sc.**

Program Studi S1 Teknologi Sains Data  
Fakultas Teknologi Maju dan Multidisiplin  
Universitas Airlangga Indonesia

# Tujuan Pembelajaran

1. Mampu memahami konsep dasar Pemrograman Web
2. Mampu memahami istilah-istilah yang ada dalam pemrograman web
3. Mampu menggunakan text editor
4. Mampu mengimplementasikan Struktur Navigasi

# Pengertian Website

**Website** adalah kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara dan video atau gabungan dari semuanya yang terkoneksi oleh internet, sehingga dapat dilihat oleh siapapun yang terkoneksi dengan jaringan internet.



# Statis vs Dinamis

- **Statis:**

- Halaman tidak/jarang berubah (statis)
- Perubahan suatu halaman dilakukan secara manual dengan mengedit codingan website tersebut
- Minim interaksi dari pengunjung atau admin
- Contoh: *company profile*, *landing page* (dengan catatan hanya menampilkan informasi saja tanpa fitur lain)

- **Dinamis:**

- Sering update (dinamis)
- Ada halaman *backend* untuk melakukan perubahan konten dari website tersebut
- Menyediakan fitur-fitur untuk interaksi dengan pengguna
- Contoh: toko online, web berita, forum, web media sosial, dll.

# Statis vs Dinamis

Website Statis	Website Dinamis
Konten jarang berubah	Konten selalu update
Cukup dibuat menggunakan HTML saja atau ditambah CSS agar lebih bagus	Biasanya dibangun menggunakan bahasa pemrograman tertentu seperti: PHP, Javascript, Ruby, dan lainnya. Jika ingin cepat bisa menggunakan framework (laravel, codeigniter, nodejs) atau CMS ( Wordpress, Magento, dan Pretashop)
User biasanya hanya melihat-lihat saja	User bisa melakukan sesuatu seperti berkomentar, mendaftar, tanya jawab, dan lainnya
Tidak menggunakan database	Menggunakan database
Loading lebih cepat karena halaman biasanya hanya sedikit, tidak ada database, dan fitur-fitur yang memberatkan server.	Loading bisa lebih lambat dari web statis, karena meload banyak halaman dan fitur

# Pemrograman Web

- **Pemrograman web** adalah pembuatan aplikasi/program menggunakan Bahasa pemrograman yang dapat diakses/dijalankan menggunakan web browser.
- **Bahasa:**
  1. **Client side:** HTML, CSS, JavaScript, XML.
  2. **Server side:** PHP, ASP, ASP.Net, Java, Ruby, Perl, Python dll



# Terminologi Penting

1. Internet
2. World Wide Web (WWW)
3. Web Server
4. URL (Universal Resource Locator)
5. HTTP (Hypertext Transfer Protocol)
6. DNS (Domain Name System)
7. IP (Internet Protocol)

# Why Domain Name

## 1. User friendly

- › No need to remember IP address

## 2. Load balancing

- › Same name maps to changing IP address

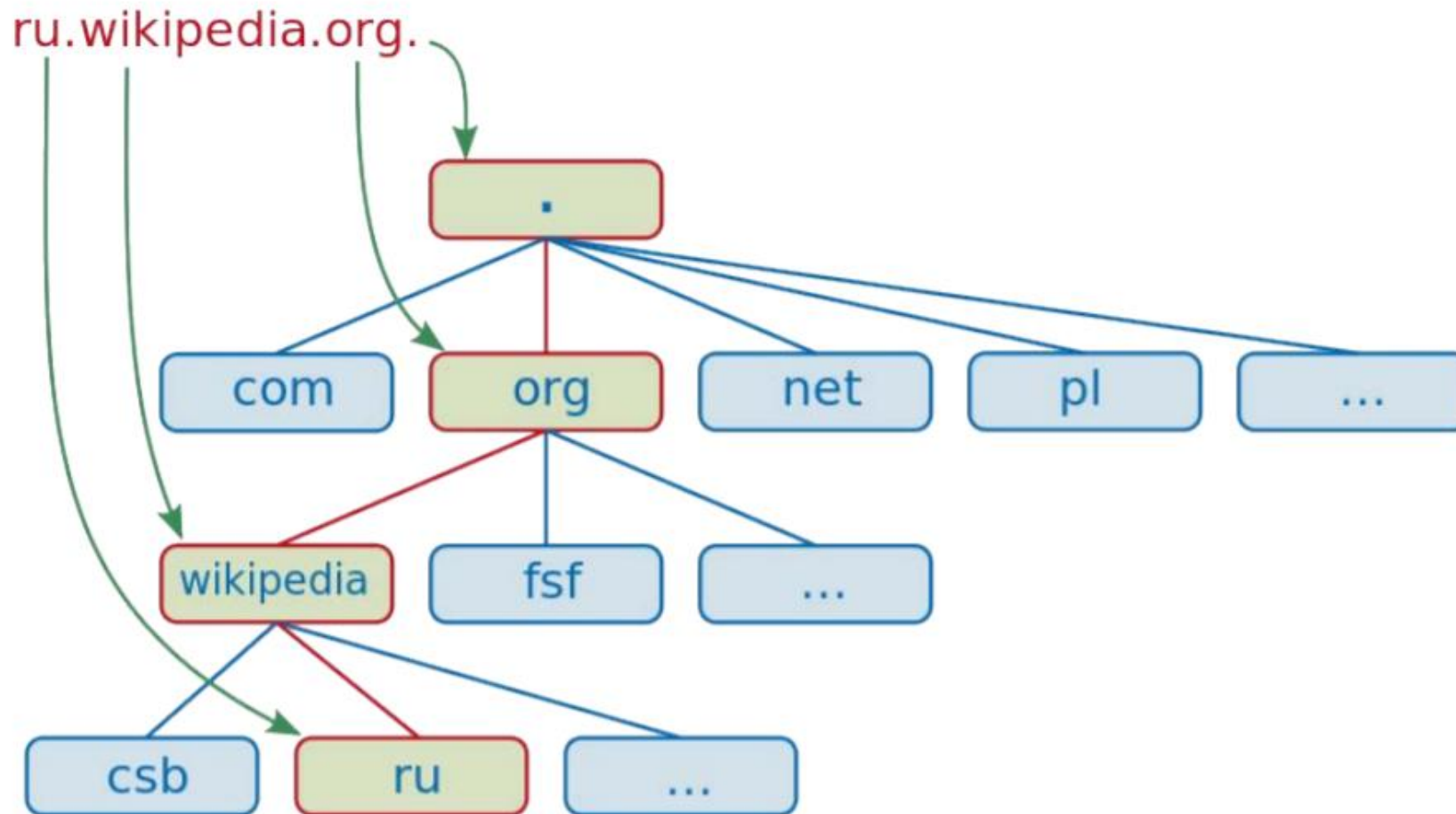
## 3. Decoupling

- › Can move server to different network, ISP, etc

| 194.31.53.144/openlearningftmm/



# Struktur Hirarki



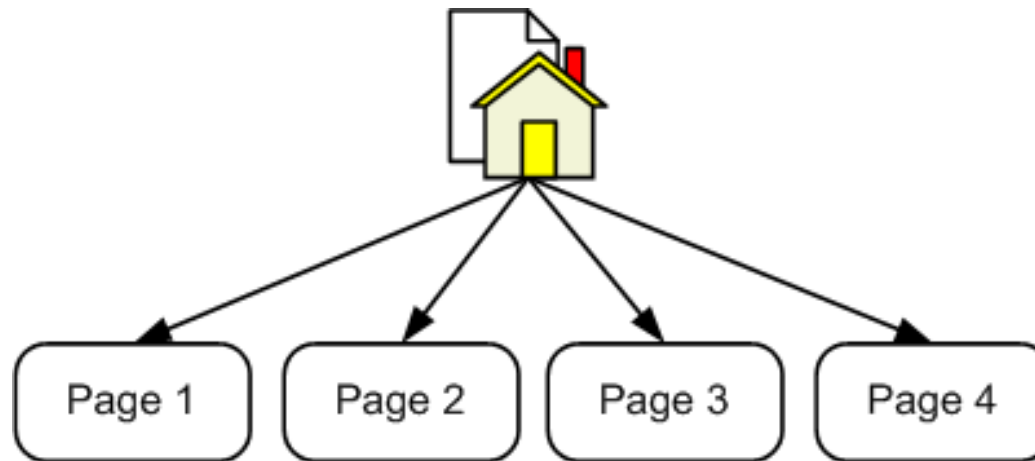
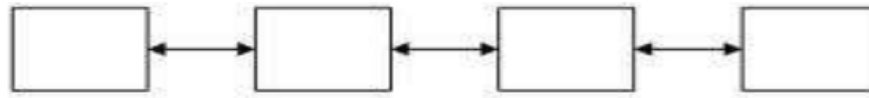
# Struktur Navigasi Website

- **Struktur navigasi:** bagan hirarki dari suatu website yang menggambarkan isi dari setiap halaman dan link.
- **Struktur navigasi:** gambaran dari hubungan atau rantai kerja dari seluruh elemen yang akan digunakan dalam aplikasi.
- **Kriteria pengelompokan struktur navigasi:**
  - Kebutuhan
  - Kemudahan pemakaian
  - Kemudahan pembuatan
  - Interaktif

# Struktur Navigasi Website

## 4 Struktur dasar:

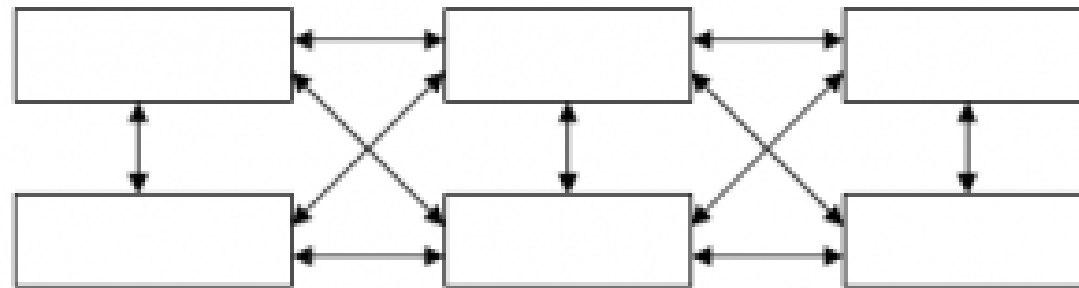
- Linear: berurut sesuai urutan



# Struktur Navigasi Website

## 4 Struktur dasar:

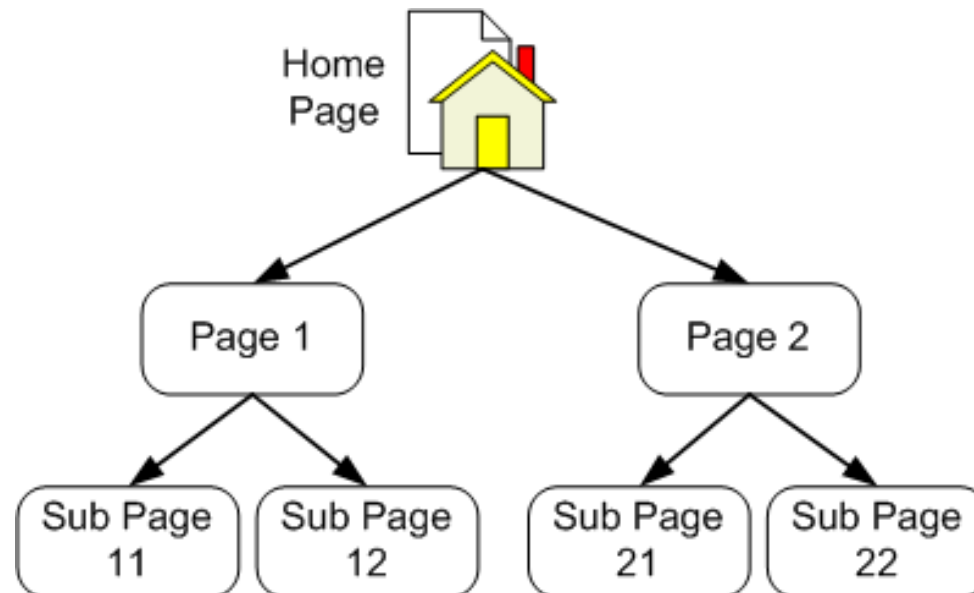
- Non-linear: pengembangan dari struktur navigasi linier dan bisa membuat navigasi bercabang. Pencabangan struktur non linier berbeda dengan struktur hirarki, karena tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada parent atau child.



# Struktur Navigasi Website

## 4 Struktur dasar:

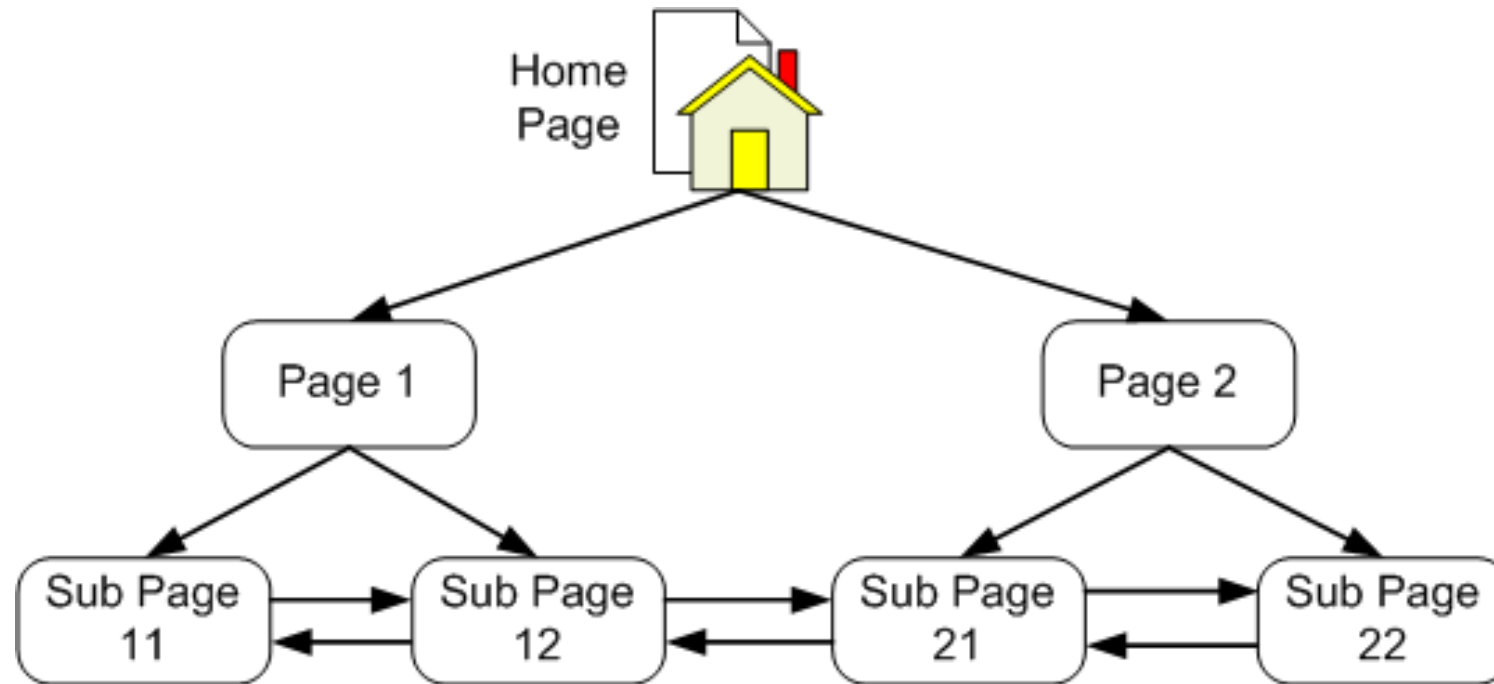
- Hirarki: hubungan antara satu kategori peringkat tinggi dengan subkategori bermakna pengkhususan kategori peringkat tinggi. Pergerakan ke atas hierarki bermakna perluasan kategori tersebut, sementara pergerakan ke bawah hierarki bermakna pengkhususan kategori tersebut



# Struktur Navigasi Website

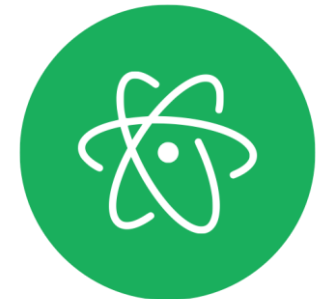
## 4 Struktur dasar:

- Campuran: bebas disesuaikan dengan kebutuhan customer.



# Text Editor

- Dalam membuat sebuah halaman web dibutuhkan **text editor**.
- **Contoh:**
  - Notepad
  - Notepad++
  - Sublime
  - Atom
  - Netbeans
  - Dll.



# Pengenalan HTML



# Hypertext Markup Language (HTML)

- **HTML adalah bahasa pemrograman yang digunakan untuk menampilkan sebuah halaman website.**
- File HTML dapat diedit oleh editor teks apapun.
- Disimpan dengan extension **.html atau .htm**
- Kode-kode dalam HTML biasanya disebut **TAG**.
- Tag dituliskan diapit oleh tanda lebih kecil ( < ), tanda lebih besar ( > ), dan garis miring ( / ).

# Struktur Dasar HTML

- Default dimulai dengan tag **<HTML>** dan diakhiri dengan **</HTML>**.
- Tag **<HEAD> ... </HEAD>** adalah tag kepala dimana akan terlebih dulu dieksekusi sebelum tag badan.
- Di dalam tag HEAD berisi tag **<META>** dan **<TITLE>**.
- Tag **<META>** merupakan informasi atau header suatu dokumen HTML.
- Tag **<TITLE> ... </TITLE>** adalah tag judul dan muncul pada titlebar browser.
- Tag **<BODY> ... </BODY>** adalah tag berisi content dari suatu halaman web.

```
<html>
<head>
    <title> Judul Web </title>
</head>
<body>

</body>
</html>
```

# HTML Element

- The HTML element is everything from the start tag to the end tag:

`<tagname>Content goes here...</tagname>`

- Contoh:

`<h1>My First Heading</h1>`

`<p>My first paragraph.</p>`

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
 	<i>none</i>	<i>none</i>

# Scripting

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

```
</html>
```

1. Buka text editor
2. Tuliskan kode disamping
3. Simpan file dengan ekstensi .htm atau .html
4. Jalankan file dengan double click
5. File akan tampil di web browser

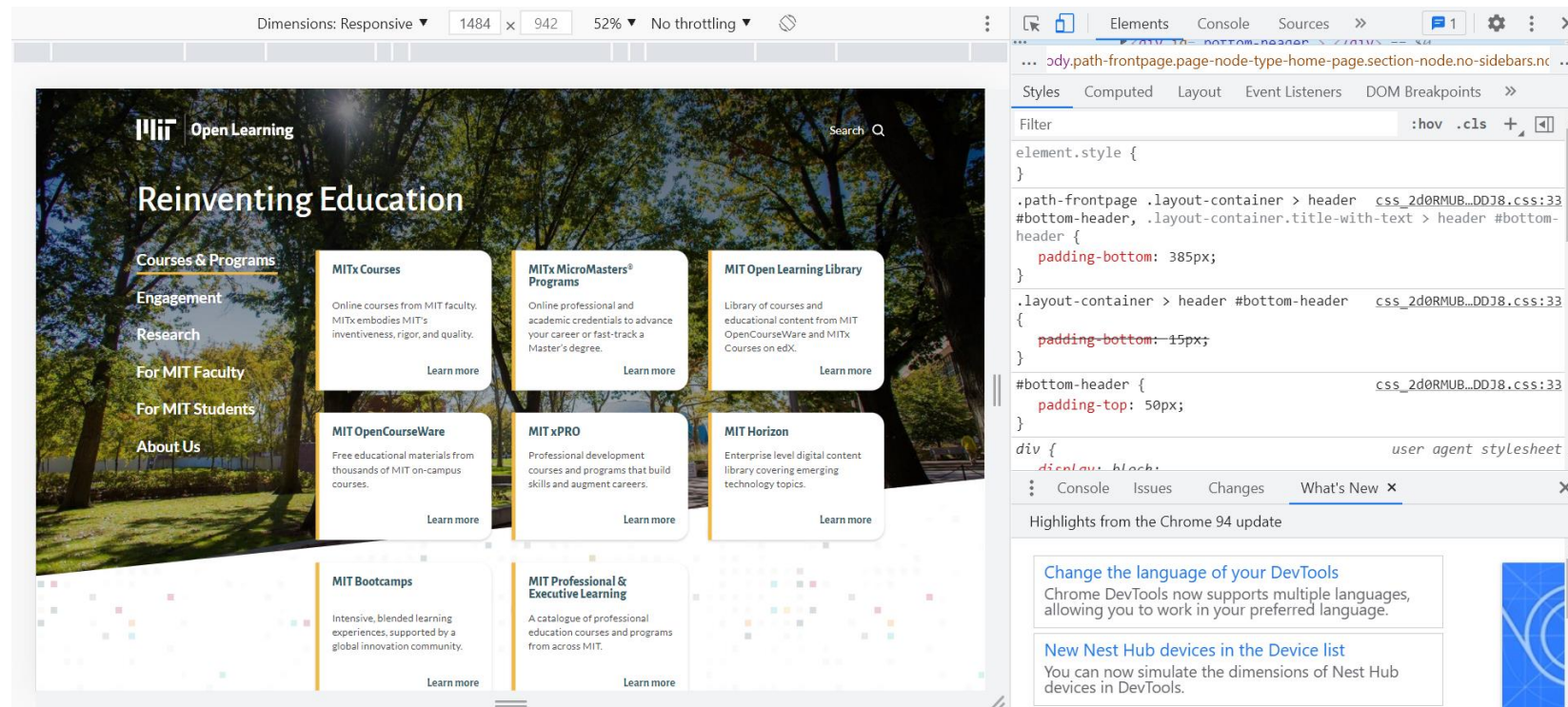


## My First Heading

My first paragraph.

# View HTML Source

- Right-click in an HTML page and select "**View Page Source**" atau
- Right-click on an element (or a blank area) and choose "**Inspect**" or "**Inspect Element**" to see HTML elements (and CSS).



# Tag Dasar

No	Nama Tag	Fungsi
1	<! DOCTYPE html>	Deklarasi untuk mendefinisikan dokumen menjadi HTML
2	<html>	Tag pembuka untuk membuat dokumen HTML
3	<head>	Informasi meta tentang dokumen
4	<title>	Membuat judul halaman yang nantinya akan ditampilkan di browser
5	<body>	Tempat dibuatnya semua konten website menggunakan HTML

# Tag Dasar

Nama Tag	Fungsi
<h1> to <h6>	Tag untuk membuat heading
<p>	Tag untuk membuat paragraf
 	Memasukan satu baris putus
<hr>	Tag untuk membuat perubahan dasar kata didalam isi
<!--...-->	Tag untuk membuat komentar
<a>	Tag untuk membuat hyperlink
<link>	Tag untuk membuat hubungan antara dokumen dan sumber daya eksternal

# Tag Dasar: Formatting

No	Nama Tag	Fungsi
1	<small>	Tag untuk membuat teks kecil
2	<strike>	Tag untuk membuat teks yang di coret tengah (tidak disupport lagi di HTML5)
3	<strong>	Tag untuk membuat teks penting
4	<b>	Tag untuk membuat huruf bercetak tebal
5	<abbr>	Tag untuk membuat sebuah singkatan



# Tag Dasar: Form

Nama Tag	Fungsi
<form>	Tag untuk membuat sebuah form HTML untuk input pengguna
<input>	Tag untuk membuat sebuah kontrol input
<textarea>	Tag untuk membuat sebuah kontrol input multibaris (text area)
<button>	Tag untuk membuat sebuah tombol yang dapat diklik
<select>	Tag untuk membuat sebuah daftar drop-down
<option>	Tag untuk membuat pilihan dalam daftar drop-down
<label>	Tag untuk membuat sebuah label untuk sebuah elemen <input>

# Tag Dasar: Images

Nama Tag	Fungsi
<code>&lt;img&gt;</code>	Tag untuk membuat gambar
<code>&lt;map&gt;</code>	Tag untuk membuat gambar-peta
<code>&lt;area&gt;</code>	Tag untuk membuat area dalam gambar-peta
<code>&lt;canvas&gt;</code>	Digunakan untuk menggambar grafik, melalui scripting (JavaScript ) (tag baru HTML5)
<code>&lt;figcaption&gt;</code>	Tag untuk membuat sebuah caption untuk elemen <code>&lt;figure&gt;</code> (tag baru HTML5)
<code>&lt;figure&gt;</code>	Menentukan konten mandiri (tag baru HTML5)

# Tag Dasar: List

Nama Tag	Fungsi
<ul>	Tag untuk membuat daftar dengan selain nomor
<ol>	Tag untuk membuat daftar dengan nomor
<li>	Tag untuk membuat sebuah item daftar
<dl>	Tag untuk membuat sebuah daftar definisi
<dt>	Tag untuk membuat istilah (item) dalam daftar definisi
<menu>	Tag untuk membuat deskripsi dari item dalam daftar definisi

# Tag Dasar: Tables

Nama Tag	Fungsi
<code>&lt;table&gt;</code>	Tag untuk membuat tabel
<code>&lt;caption&gt;</code>	Tag untuk membuat sebuah caption tabel
<code>&lt;th&gt;</code>	Tag untuk membuat sebuah sel header tabel
<code>&lt;tr&gt;</code>	Tag untuk membuat baris dalam sebuah tabel
<code>&lt;td&gt;</code>	Tag untuk membuat sel dalam sebuah tabel
<code>&lt;thead&gt;</code>	Mengelompokkan isi header dalam sebuah tabel
<code>&lt;tbody&gt;</code>	Mengelompokkan isi tubuh dalam sebuah tabel
<code>&lt;col&gt;</code>	Menentukan properti kolom untuk setiap kolom dalam elemen <code>&lt;colgroup&gt;</code>
<code>&lt;colgroup&gt;</code>	Menentukan kelompok dari satu atau lebih kolom dalam sebuah tabel untuk diformat

# Pengenalan CSS

# Cascading Style Sheets (CSS)

- **CSS:** bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup/markup language.
- Dalam konteks web, CSS adalah bahasa yang digunakan untuk mengatur **tampilan/desain sebuah halaman HTML.**

# Cara Penggunaan CSS

- **External Style Sheet:** CSS disimpan pada suatu file sehingga terpisah dari dokumen HTML. Cara menggunakan tinggal melakukan pemanggilan file CSS dalam dokumen HTML.

```
<html>
<head>
  <title>CSS Secara Internal</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

- **Internal Style Sheet:** CSS ditulis pada bagian HEAD dokumen HTML menggunakan tag <style>

```
<html>
<head>
  <title>CSS Secara Internal</title>
  <style type="text/css">P {text-align:justify;}
</style>
</head>
```

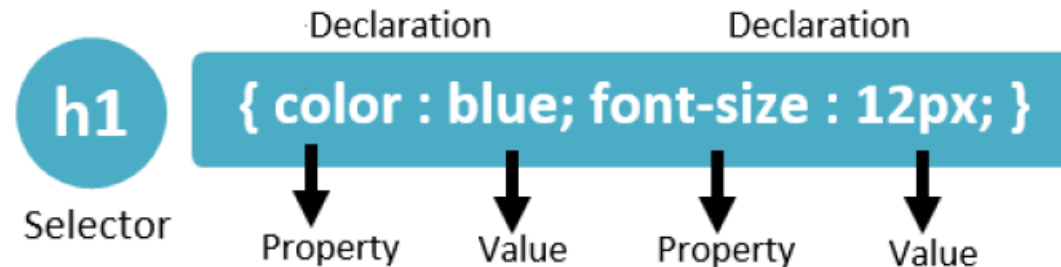
# Cara Penggunaan CSS

- **Inline Style Sheet:** CSS ditulis langsung pada tag HTML yang akan diatur tampilannya menggunakan atribut style

```
<html>
<head>
  <title>CSS Secara Internal</title>
  <style type="text/css">P{text-align:justify;}
</style>
</head>
<body>
<p style ="text-align:justify;"> Paragraph yang diatur
CSS Secara Internal</p>
</body>
</html>
```



# Penulisan CSS



## Struktur CSS:

1. **Selector:** biasanya berupa tag HTML, id, class
2. **Declaration:** berisi aturan-aturan css yang terdiri dari properti dan nilainya yang dipisahkan oleh tanda titik dua. Setiap aturan css harus diakhiri dengan tanda titik koma.

# Penulisan CSS: Selector

- **Biasanya berupa tag HTML, id, class**
  - id menggunakan tanda # didepan nama selector
  - class menggunakan tanda titik didepan nama selector
- **Contoh:**

`h1 { color : blue ; }` → tag html h1

`#teks { color :green; }` → id

`.warna { color : red; }` → class

# Penulisan CSS: Declaration

- **Selector ID pada CSS**

```
<body>  
  <p id="teks">TEST  
</p>  
</body>
```



```
#teks  
{  
  Color : blue;  
  Font-family: Calibri;  
}
```

- **Aturan ID:**

- Sebuah elemen HTML hanya boleh memiliki 1 id
- Setiap halaman hanya boleh memiliki 1 elemen dengan id tersebut
- Sebaiknya tidak digunakan untuk css ( lebih baik gunakan class)

# Penulisan CSS: Declaration

- **Selector Class pada CSS**

```
<body class="warna">  
</body>
```



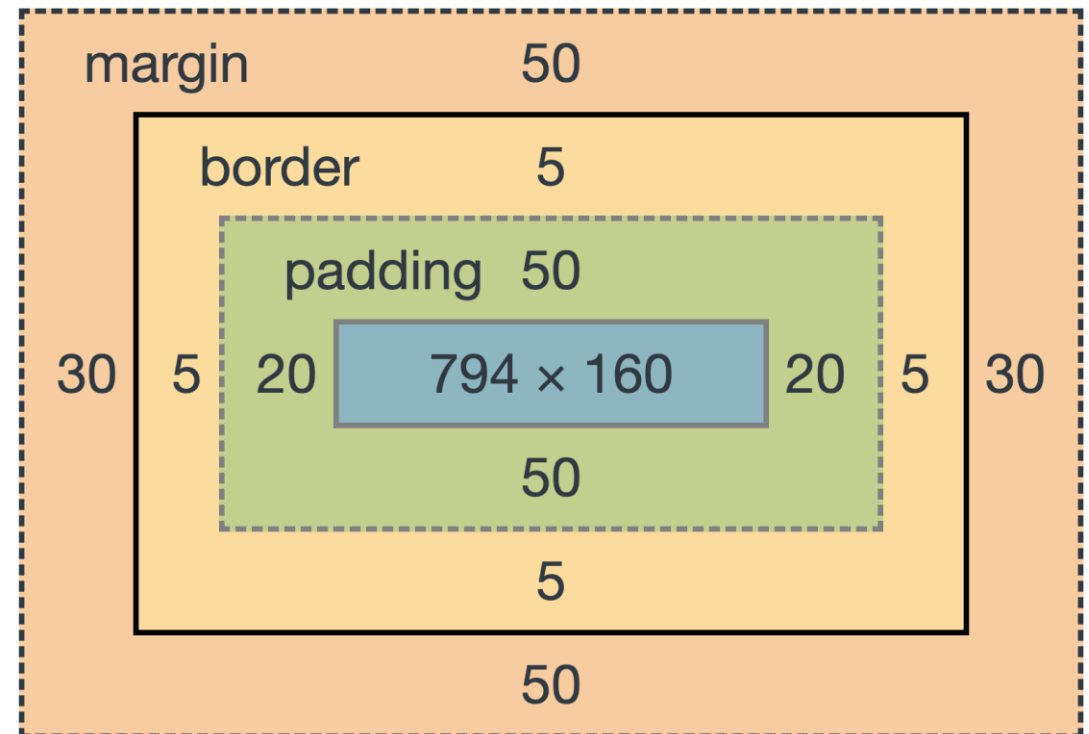
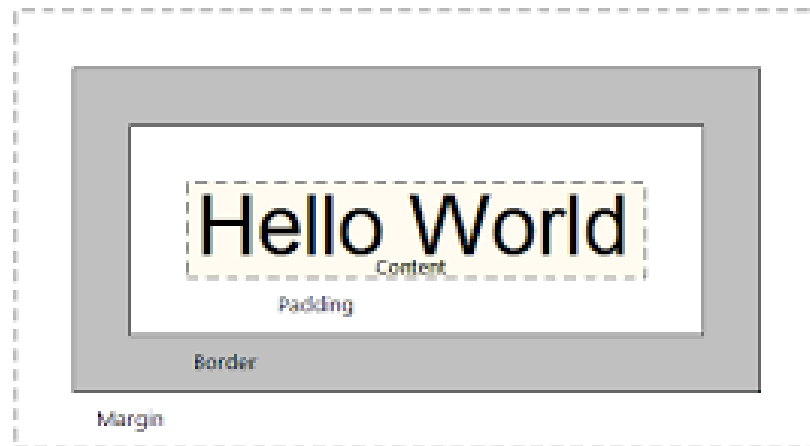
```
.warna  
{ background-color: lightgreen; }
```

# Properti CSS

Properti	Fungsi	Nilai	Contoh penulisan :
Color	Mengatur warna pada Teks	Nama warna, kode hexa warna	Color : blue; Color:#ffffff;
Background-color	Mengatur warna latar	Nama warna, kode hexa warna,rgb	Background-Color:rgb(200,0,55);
Background-image	Mengatur gambar latar	Nama file gambar	Background-image:url(header.jpeg);
Text-align	Mengatur perataan teks	Left,Center, right, justify	Text-align:left;
Font-family	mengatur jenis font	Nama-nama jenis huruf, co: arial, times new roman, georgia	Font-family:calibri;
Dll.			

# Padding, Margin, dan Border

- Dalam CSS dikenal istilah '**Box Model**'.
- Berguna untuk **tata letak elemen**



# Responsive Web

- Teknik atau metode bagi web designer untuk membuat suatu layout website yang dapat menyesuaikan diri sesuai dengan ukuran layar pengguna.
- Baik dari ***ukuran huruf, user interface, gambar*** dan ***tata letak*** akan menyesuaikan dengan lebar layar dan resolusi device yang digunakan.
- **Popular CSS framework:** Bootstrap, Materialize CSS, Foundation, Bulma, Tailwind CSS, UIKit, Milligram, Pure.CSS, etc.
- Keuntungan:
  - Kenyamanan
  - Dapat diakses oleh berbagai device
  - Hemat biaya
  - Dll.

# Contoh Responsive Web

- Media queries
- Fluid grids
- Flexible visuals
- Dribbble
- GitHub
- Klientboost
- Magic Leap
- DII.

**Trend website design 2021:** <https://99designs.com/blog/trends/web-design-trends/>

**Tips Interface Design bagus:** <https://www.webstyleguide.com/7-interface-design.html>



# Pengenalan JavaScript

# Programming the browser

- **Java**

- › 1990: project at Sun to replace C++
- › 1994: “Oak”retargeted to the web for “applets”
- › Java takes off, first safe language in widespread use

- **Javascript**

- › 1995: “Mocha” project at Netscape
- › Javascript takes off, included in Microsoft’s IE
- › 1996: submitted to Ecma as standard

- **today**

- › Java alive and well server-side
- › but JS dominates client-side
- › making inroads server-side too (eg, node.js)

# Syntax

- **statements like Java**
  - while, for, if, switch, try/catch, return, break, throw
- **comments**
  - use `//`, avoid `/**/`
- **semicolons**
  - inserted if omitted
- **declarations**
  - function scoping with `var`

```
var MAX = 10;
var line = function (i, x) {
    var l = i + " times " + x
        + " is " + (i * x);
    return l;
}
var table = function (x) {
    for (var i = 1; i <= MAX; i += 1) {
        console.log(line(i, x));
    }
}
// display times table for 3
table(3);
```

1 times 3 is 3
2 times 3 is 6
3 times 3 is 9
4 times 3 is 12
5 times 3 is 15
6 times 3 is 18
7 times 3 is 21
8 times 3 is 24
9 times 3 is 27
10 times 3 is 30
< undefined
>

# Basic Types

- **primitive types**
  - › strings, numbers, booleans
  - › operators auto convert
- **arrays**
  - › can grow, and have holes
- **funny values**
  - › undefined: lookup non-existent thing
  - › null: special return value
- **equality**
  - › use ===, !==

```
> 1 + 2
3
> 1 + '2'
"12"
> 1 * 2
2
> 1 * '2'
2
```

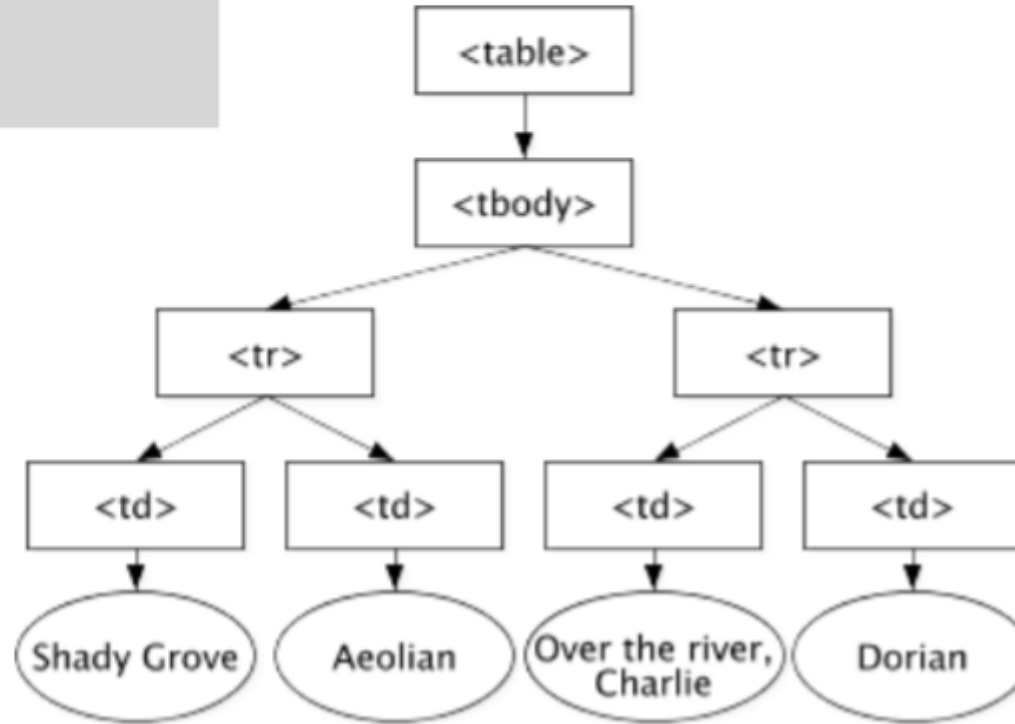
```
> a = []
[]
> a[2] = 'hello'
"hello"
> a.length
3
> a[1]
undefined
> a[2]
"hello"
> a[3]
undefined
```

```
> 1 === 1
true
> 1.0 === 1
true
> 'hello' === 'hello'
true
> [] === []
false
```

# Document Object Model (DOM)

```
<table>
  <tbody>
    <tr>
      <td>Shady Grove</td>
      <td>Aeolian</td>
    </tr>
    <tr>
      <td>Over the River, Charlie</td>
      <td>Dorian</td>
    </tr>
  </tbody>
</table>
```

**DOM:** *interface* yang memungkinkan *developer* untuk memanipulasi konten, struktur, dan *style* situs web.



# Client Side Programming

- **where's the code?**
  - › JavaScript
- **manipulating the DOM**
  - › traverse the elements
  - › add and delete elements
  - › modify properties (eg position)
  - › modify styles (eg color)
- **also**
  - › registering events (against timers, buttons)
  - › making calls to web servers

# Modifying Styles

```
<!DOCTYPE html>
<head>
  <style>
    .plain {color: black}
    .highlight {color: red}
  </style>
</head>
<body>
  <h1 id="header" class="">
    My new webpage
  </h1>
  <div id="content" class="plain">Welcome!
  </div>
</body>
</html>
```

page

**My new webpage**

Welcome!

before

**My new webpage**

Welcome!

after

```
function highlight_content() {
  var e = document.getElementById('content');
  e.className = 'highlight';
}
highlight_content();
```

code executed in console

# Animating Element

```
<!DOCTYPE html>
<head>
  <style>
    #object {
      position: absolute;
      background: yellow;
      left: 0px;
    }
  </style>
</head>

<body>
  <h1 id="header" class="">
    My new webpage
  </h1>
  <div id="object">Watch me!
  </div>
</body>
</html>
```

page

```
var obj = document.getElementById('object');
var i = 0;
function animate() {
  obj.style.left = i + 'px';
  i = i + 10;
  setTimeout(animate, 20);
}
animate();
```

code executed in console

My new webpage

Watch me!

effect



# Inserting Element

```
<!DOCTYPE html>
<head>
  <style>
    #content {
      font-size: 20px;
    }
    .popup {
      position: relative;
      left: 10px;
      width: 100px;
      top: -40px;
      opacity: 0.5;
      background: yellow;
    }
  </style>
</head>
<body>
  <h1 id="header" class="">
    My new webpage
  </h1>
  <div id="content">Content
</div>
</body>
</html>
```

page

```
function popup(elt) {
  var box = document.createElement('div');
  box.innerHTML = "comment";
  box.className = 'popup';
  elt.appendChild(box);
}
var elt = document.getElementById('content');
popup(elt);
```

code executed in console

**My new webpage**

Content

before

**My new webpage**

comment  
Content

after

# Latihan

- Explore lagi penggunaan HTML, CSS, dan Javascript.
- Cobalah membuat halaman website statis sederhana menggunakan HTML, CSS, dan Javascript.

# Terima Kasih

## Referensi:

[\*https://www.w3schools.com/html/\*](https://www.w3schools.com/html/)

[\*https://www.w3schools.com/css/\*](https://www.w3schools.com/css/)

[\*https://www.webstyleguide.com/7-interface-design.html\*](https://www.webstyleguide.com/7-interface-design.html)

*Web programming karya Ani Oktarini Sari et.al.*

[\*https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-170-software-studio-spring-2013/lecture-notes/\*](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-170-software-studio-spring-2013/lecture-notes/)