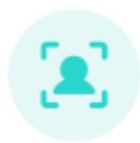


Cascading Style Sheets & JavaScript

Aturan **Instructor/Expert Led Session**



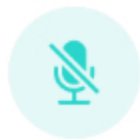
Fokus Penuh



Always-on Camera



Raise hand atau Chat jika ingin bertanya



Mute jika sedang tidak berbicara

Porsi Skor Penilaian (Tech)

Jenis Aktivitas	Porsi Skor
Rata-rata Kuis pada ILT	20%
Aktivitas selama ILT (bertanya, menjawab, membantu diskusi)	10%
Aktivitas Forum Diskusi*	10%
Rata-rata Exam/Submission Project	60%

*Jika ditemukan peserta spamming pada forum diskusi, maka nilai terbaik yang bisa didapat adalah 60 (dari skala 100)

Materi

Cascading Style Sheets (CSS)

Pengenalan CSS

- Bukan bahasa pemrograman, melainkan *declarative language*.
- Melampirkan CSS di dokumen HTML:
 - Inline style: `<div style="background: #303034;">...</div>`
 - Embedded style: `<style>...</style>`
 - External style: `<link href="style.css" rel="stylesheet">`

Anatomi CSS

- Selector: menyeleksi elemen apa saja yang akan menerapkan style tersebut.
- Declaration: kumpulan pasangan property dan value-nya.
- Property: visual karakteristik dari elemen yang akan dimodifikasi.
- Value: nilai dari properti.

CSS ruleset

selector

```
h1 {  
  color: green;  
}
```

declaration block

```
h1 {  
  color: green;  
}
```

declaration

```
h1 {  
  color: green;  
}
```

property value

```
h1 {  
  color: green;  
}
```

CSS inline style

opening tag

```
<h1 style="color: green;">Halo Bandung!</h1>
```

attribute

```
<h1 style="color: green;">Halo Bandung!</h1>
```

declaration

```
<h1 style="color: green;">Halo Bandung!</h1>
```

property value

```
<h1 style="color: green;">Halo Bandung!</h1>
```

CSS Conception

- Inheritance
- Group selector
- Rule order (the cascade)

The Cascade

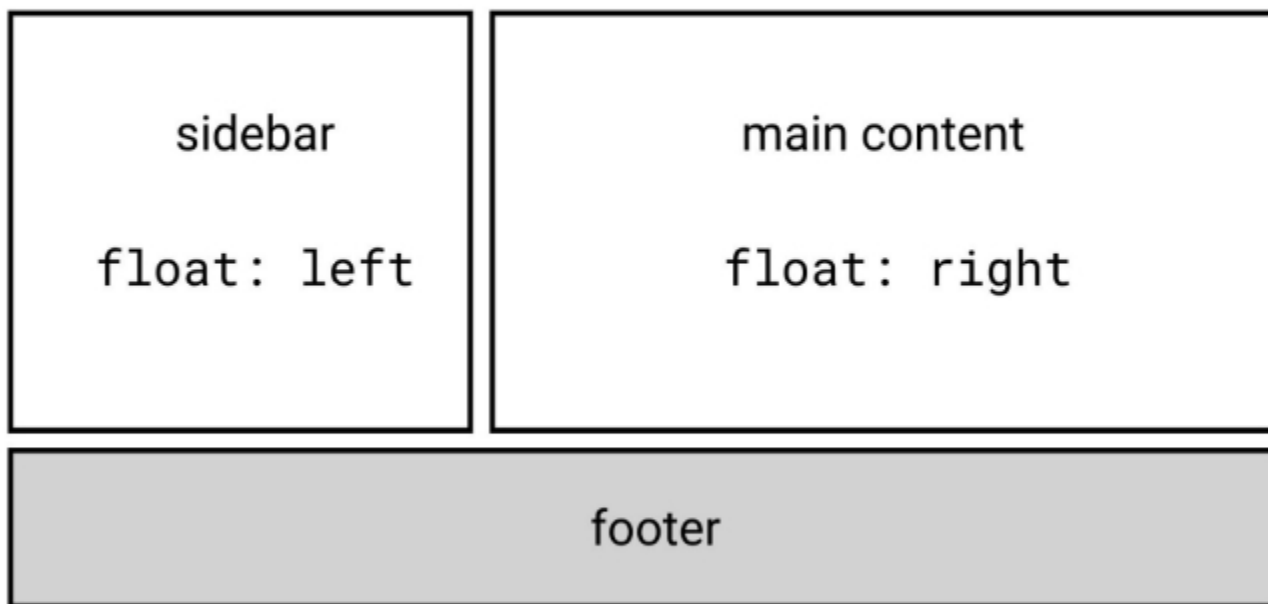
- Algoritma untuk menentukan rules mana yang akan diterapkan pada sebuah elemen.
- Kondisi yang mempengaruhi:
 1. Posisi dan urutan kode CSS
 2. Specificity
 3. Origin
 4. Importance (!important)

Specificity

Selector	Contoh	Nilai
Universal selector	*	0 poin
Element Pseudo-element	h1 p::before	1 poin
Class Pseudo-class	.title :hover [name="phone"]	10 poin
ID	#title	100 poin
Inline style	<p style="">	1000 poin
!important rule	font-size: 1rem;	10000 poin

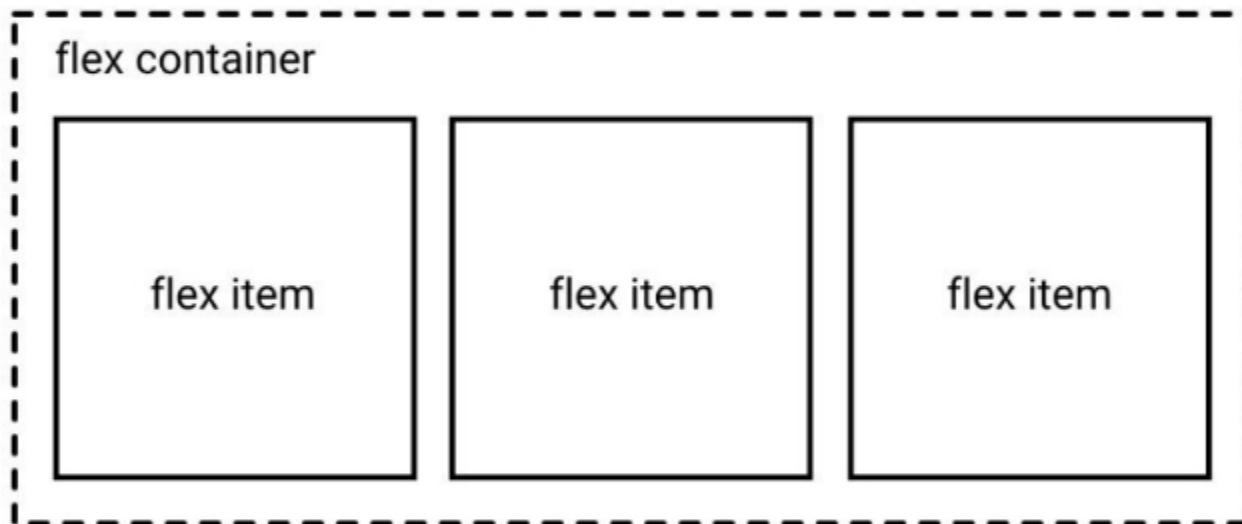
Layouting Menggunakan Float

```
#sidebar {  
  float: left;  
}  
  
#main-content {  
  float: right;  
}  
  
#footer {  
  clear: both;  
}
```



Layouting Menggunakan Flexbox

```
.flex-container {  
  display: flex;  
}
```



Membangun Website Responsive

- Menambahkan viewport meta tag.

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

- Menggunakan satuan relative.

```
.articles {  
  width: 100%;  
  max-width: 1000px;  
}  
  
.article {  
  width: 25%;  
}
```



Membangun Website Responsive

- Menggunakan media query.
- Menggunakan pendekatan mobile-first.


```
/* layout for small devices */  
  
@media screen and (min-width: 480px) {  
  /* layout for medium devices */  
}  
  
@media screen and (min-width: 720px) {  
  /* layout for large devices */  
}
```

JavaScript (JS)

Variabel JavaScript


- Var: rawan menimbulkan bug
- Let: pengganti var
- Const: nilai akan selalu tetap

var

var apple = 




a thing in a box
named "apple"

apple = 



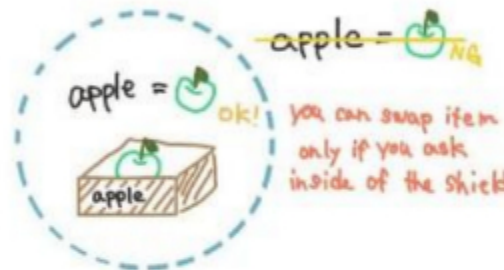
you can swap
item later

let

let apple = 




a thing in a box
named "apple" w/
protection shield



you can swap item
only if you ask
inside of the shield

const

const apple = 



a thing in
LOCKED cage
named "apple"



you can't
swap item
later.



apple.multiply(3)
... but you can ask
the item to change itself
(if the item has method
to do that)

Sumber: @mpjme (<https://twitter.com/mpjme>)

Tipe Data

- Undefined
- Numbers
- BigInt
- Strings
- Boolean
- Null

```
// Undefined
```

```
const x;
```

```
// Number
```

```
const length = 16;
```

```
// BigInt
```

```
const length =
```

```
1234567890123456789012345678907890n;
```

```
// String
```

```
const firstName = "Joe";
```

```
const lastName = 'Johnson';
```

```
// Boolean
```

```
const x = true;
```

```
const y = false;
```

```
// Null
```

```
let people = null;
```

Object

- Mengelola data berstruktur.
- Di dalamnya mengandung properti dengan format key-value.
- Untuk mengakses properti, gunakan tanda titik.

```
const user = {  
  firstName: 'Luke',  
  lastName: 'Skywalker',  
  age: 19,  
  isJedi: true,  
};
```

```
console.log(user.firstName); // Luke  
console.log(user.lastName); // Skywalker  
console.log(user.age); // 19  
console.log(user.isJedi) // true
```

Array

- Berisi sekumpulan data.
- Bersifat iterable.
- Untuk mengakses item, gunakan teknik indexing.

```
const favoriteFood = ['Soup', 'Pizza',  
  'Potato', 'Spinach'];
```

```
console.log(favoriteFood[0]); // Soup  
console.log(favoriteFood[1]); // Pizza  
console.log(favoriteFood[2]); // Potato  
// etc ..
```

Operator

- Assignment
- Arithmetic
- Comparison
- Logical

```
// Assignment  
const message = 'Hello World';
```

```
// Arithmetic  
4 + 6; // 10  
"John" + "Doe"; // JohnDoe  
6 - 4; // 2  
6 * 2; 36  
8 / 4; // 2  
14 % 6; // 2
```

```
// Comparison  
7 > 6;  
7 >= 7;  
4 < 8;  
4 <= 4;  
4 == '4';  
4 === 4;
```

```
// Logical  
6 < 7 && 9 < 7; // false  
6 < 7 || 9 < 7; // true
```

Percabangan

- If - Else
- Switch - Case

```
// if - else
const x = 50;
if (x > 70) {
  console.log(x);
} else {
  console.log("Nilai kurang dari 70");
}

// switch - case
let language = "English";

switch (language) {
  case "English":
    console.log("Good Morning!");
    break;
  case "Japanese":
    console.log("Ohayou!");
  default:
    console.log("Selamat Pagi!");
}
```

Perulangan

- For Loop
 - Regular For
 - For Of
 - For In
- Do While

```
// for loop
for (let i = 0; i < 5; i++) {
  console.log('Hello World'); // Hello World 5x
}

// for of
const foods = ['Salad', 'Steak', 'Pizza'];
for (const food of foods) {
  console.log(food); // Salad, Steak, Pizza
}

// for in
for (const foodIndex in foods) {
  console.log(foodIndex); // 1, 2, 3
}

// do while
let counter = 0;
do {
  console.log('Hello World'); // Hello World
  5x;
  counter++;
} while (counter < 5);
```

Function

- Kumpulan statement yang menjalankan suatu proses berdasarkan input yang diberikan.
- Bersifat Reusable.
- Dapat menghasilkan atau mengembalikan nilai.

```
function greeting(language) {  
  if(language === 'English') {  
    console.log('Good morning!');  
  } else if (language === 'Indonesia') {  
    console.log('Selamat pagi!');  
  } else {  
    console.log('Hello!');  
  }  
}
```

```
greeting('English'); // Good morning!  
greeting('Indonesia'); // Selamat pagi!
```

```
function adder(a, b) {  
  return a + b;  
}
```

```
const result = adder(4, 6);  
console.log(result); // 10
```

JavaScript pada Browser

Browser Global Object (*window*)

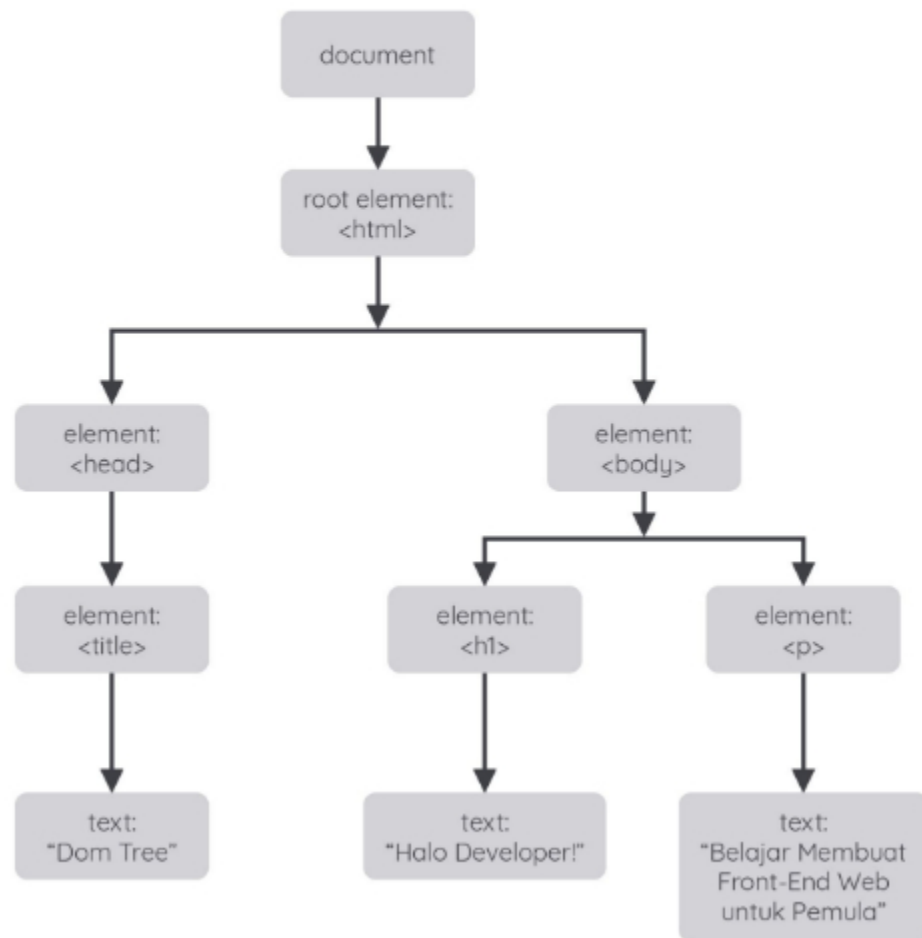
Properti	Deskripsi
document	Mendapatkan struktur halaman website dalam bentuk DOM.
console	Mencetak nilai pada konsol.
alert	Menampilkan pesan pop-up pada browser.
localStorage	Menyimpan data pada storage lokal browser.

Lihat seluruh properti window pada dokumentasi:

<https://developer.mozilla.org/en-US/docs/Web/API/Window>

Document Object Model (DOM)

```
<!DOCTYPE html>
<html>
  <head>
    <title>DOM Tree</title>
  </head>
  <body>
    <h1>Hello Developer Front-End Web! </h1>
    <p>Belajar Membuat Front-End Web untuk Pemula </p>
  </body>
</html>
```



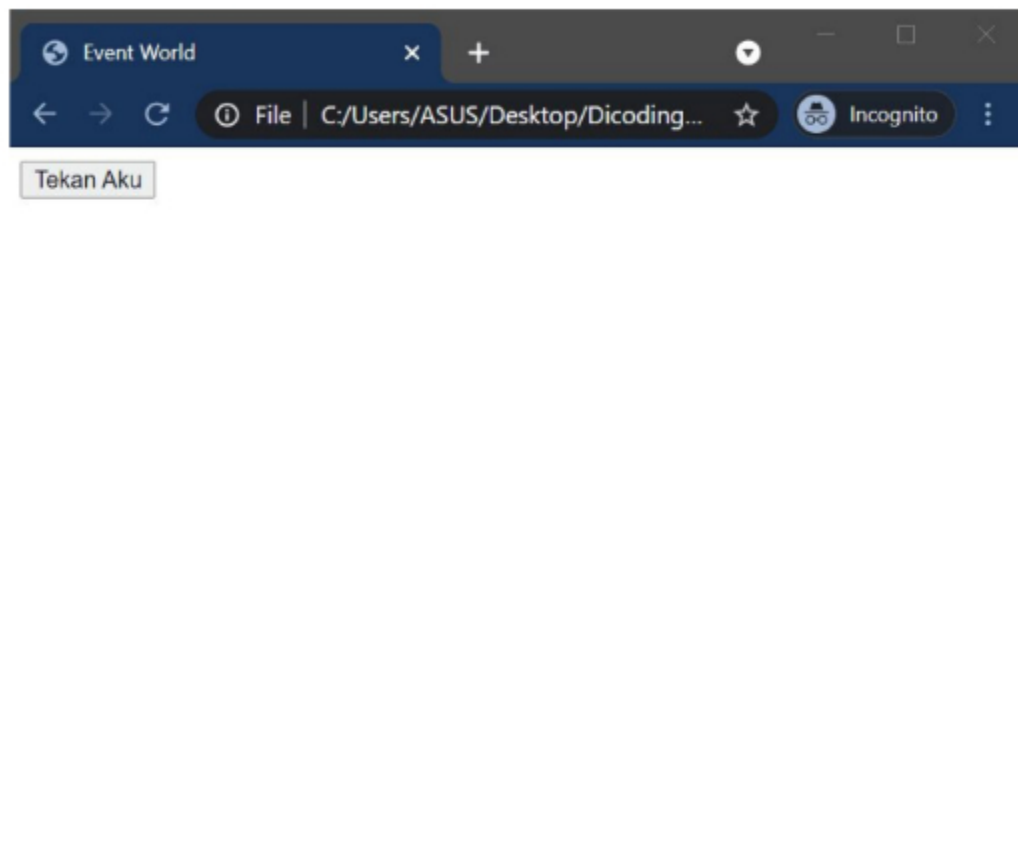
Properti fungsi DOM

Properti fungsi	Deskripsi
querySelector	Mendapatkan satu elemen dalam bentuk DOM berdasarkan CSS selector.
querySelectorAll	Mendapatkan seluruh elemen dalam bentuk DOM (NodeList) berdasarkan CSS selector.
addEventListener	Menambahkan event handler pada elemen HTML.
createElement	Membuat elemen HTML.
appendChild	Menambahkan elemen HTML sebagai child pada elemen HTML lain.

Lihat seluruh properti document (DOM) pada dokumentasi:
<https://developer.mozilla.org/en-US/docs/Web/API/Document>

Event pada elemen HTML

- Kondisi yang terjadi pada elemen HTML.
- Memberikan *handler function* dengan melalui:
 - `addEventListener`
 - event atribut.



Event pada elemen HTML

Tipe	Nama	Atribut	Deskripsi
Window	load	onload	Event ini akan terjadi setelah halaman web selesai diunduh dan ditampilkan.
Form	submit	onsubmit	Event ini akan terjadi ketika user menekan tombol "submit" pada form.
Keyboard	keydown	onkeydown	Event ini akan terjadi ketika sebuah tombol pada keyboard ditekan.
Clipboard	copy	oncopy	Event ini akan terjadi ketika kita melakukan proses copy pada konten elemen.
Mouse	click	onclick	Event ini akan terjadi ketika click/touch elemen dengan cursor/touchscreen.

Lihat seluruh referensi event pada dokumentasi:

<https://developer.mozilla.org/en-US/docs/Web/Events>

Ujian

Sesi Diskusi

dicoding

Kampus
Merdeka
INDONESIA JAYA