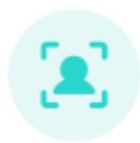


Membangun Aplikasi Frontend

Aturan **Instructor/Expert Led Session**



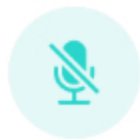
Fokus Penuh



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)

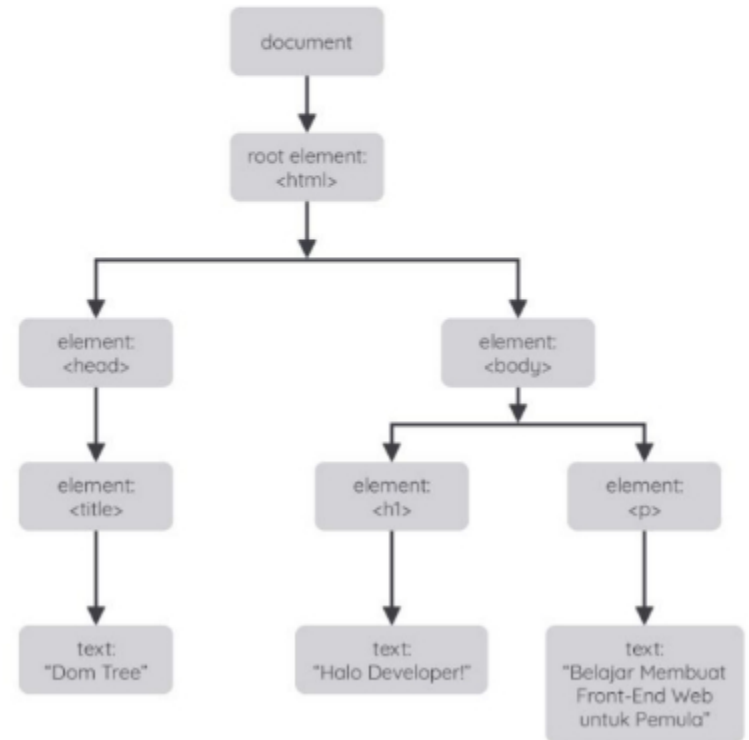
Materi

DOM Manipulation & Event

Document Object Model (DOM)

```
<html>
  <head>
    <title>Dom Tree</title>
  </head>

  <body>
    <h1>Halo Developer!</h1>
    <p>Belajar Membuat Front-End Web untuk Pemula</p>
  </body>
</html>
```



Bekerja dengan Element

- Mengakses element di sekitar
- Manipulasi element
- Bekerja dengan atribut
- Manipulasi styles
- Bekerja dengan event
- etc.

```
const element = document.querySelector('h1')
```

```
element.parentNode  
element.firstChild  
element.lastChild  
element.nextSibling  
...
```

```
element.prepend(otherElement),  
element.remove(),  
element.appendChild(otherElement),  
element.cloneNode(),  
...
```

```
element.getAttribute('id'),  
element.hasAttribute('href'),  
element.setAttribute('id', 'item-1')  
...
```

```
element.classList.add('text-green'),  
element.className = 'blog-item'  
element.style.display = 'block'  
...
```

Mencari Element

<code>.getElementById('book-1')</code>	Mendapatkan element pertama dengan id="book-1".
<code>.getElementsByName('book')</code>	Mendapatkan semua element yang memiliki name="book".
<code>.getElementsByTagName('p')</code>	Mendapatkan semua element dengan tag <p>.
<code>.getElementsByClassName('my-class')</code>	Mendapatkan semua element yang memiliki atribut class="my-class".
<code>.querySelector('input[checked]')</code>	Mendapatkan element <input> pertama yang memiliki atribut checked.
<code>.querySelectorAll('.book-level')</code>	Mendapatkan semua element yang memiliki class="book-level".

Event Listener

JavaScript + No-Library (pure JS) ▼

```
1 let myButton = document.querySelector('button.btn-info');
2
3 myButton.addEventListener('click', (e) => {
4   if ( e.target.innerText !== 'Clicked!' )
5   {
6     e.target.innerText = 'Clicked!'
7   }else{
8     e.target.innerText = 'Click Me!'
9   }
10 });
```

Click Me!

`addEventListener` : method untuk mendaftarkan listener pada sebuah target element

`'click'` : jenis event yang akan didaftarkan

`(e) => ...` : event handler, apa yang akan dilakukan ketika event di atas terjadi

Bekerja dengan Event Listener

- Menentukan target element
- Membuat fungsi event handler
- Mendaftarkan event handler untuk sebuah event pada target element

```
const el = document.querySelector('button')

function showAlert() {
  alert('Halooo 🖐️')
}

/* on[event] properties */
el.onclick = showAlert

/* method addEventListener */
el.addEventListener('dblclick', showAlert)

el.removeEventListener('dblclick', showAlert)
```

Web Component

Apa itu **Web Components**?

- Sekumpulan alat yang memungkinkan developer untuk membuat custom element
- Custom element: element-element yang bukan merupakan bagian dari spesifikasi HTML, misal:
 - `<blog-item>`
 - `<card>`

The Benefits

- Cleaner code
- Reusable
- Easy to maintain
- Custom element
- Encapsulate component view (with Shadow DOM)



Penerapan HTML Template & Slot

- Membuat html template dengan slot sebagai placeholder konten dalam element.
- Mendaftarkan custom element menggunakan template.

```
--html--
<template id="my-salam">
  <span>
    <slot name="name">NAME</slot>
  </span>
  <slot name="greeting-word"><p>GREETING</p></slot>
  <br />
</template>

<my-salam>
  <span slot="name">Adi</span>
  <span slot="greeting-word">Selamat Malam!</span>
</my-salam>

--js--
customElements.define('my-salam',
  class extends HTMLElement {
    constructor() {
      super();
      const template =
document.getElementById('my-salam').content;
      const shadowRoot = this.attachShadow({mode: 'open'})
      .appendChild(template.cloneNode(true));
    }
  }
);
```

Life Cycle

<code>connectedCallback()</code>	Akan dijalankan ketika element berhasil ditambahkan ke DOM.
<code>disconnectedCallback()</code>	Akan dijalankan ketika element dihapus dari DOM.
<code>attributeChangedCallback(name, oldValue, newValue)</code>	Akan dijalankan ketika ada perubahan nilai pada atribut, juga dapat mengembalikan nilai atribut sebelum dan sesudah perubahan.

Penyimpanan Data di Sisi Klien (Browser)

Opsi yang Tersedia

localStorage	<ul style="list-style-type: none">• Format key-value• Hanya mengenali tipe data string• Tidak memiliki waktu kedaluwarsa• Synchronous
sessionStorage	<ul style="list-style-type: none">• Memiliki sifat dan API yang sama dengan localStorage• Memiliki waktu kedaluwarsa, ketika halaman ditutup
indexedDB	<ul style="list-style-type: none">• Untuk data dengan struktur yang kompleks• Asynchronous

Security Concern

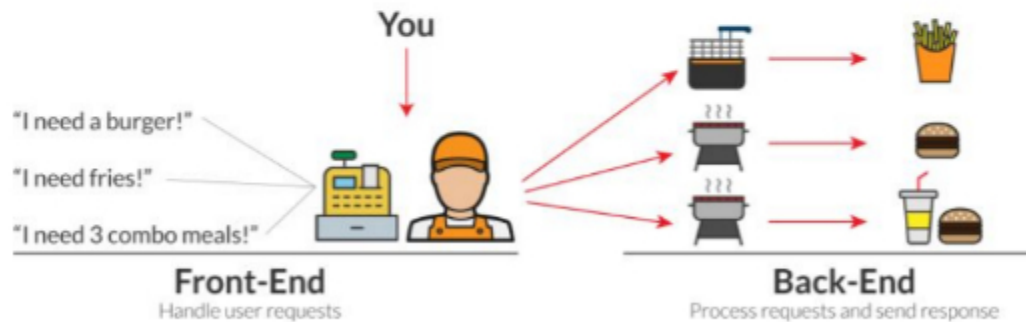
“Jangan menyimpan informasi sensitif di browser storage.”

- Data pribadi atau *Personally Identifiable Information* (PII)
- Credentials sistem yang seharusnya tidak dapat diakses oleh publik
- Access token

Asynchronous JavaScript And XML (AJAX)

Asynchronous Process

Proses asynchronous bisa dianalogikan ketika kita sedang asik bermain game dan merasa lapar dan ingin makan pizza, kita akan memesan pizza. Kita dapat terus bermain game sambil menunggu pizza datang.



Mengapa menggunakan **AJAX**?

- Aplikasi lebih interaktif tanpa perlu memuat ulang halaman
- UX seperti aplikasi native
- PWA ready



Menerapkan **AJAX**

- XMLHttpRequest
 - Metode lama
 - Pendekatan callback
- Fetch API
 - Standar baru
 - Menggunakan Promise

```
// XHR
const request = new XMLHttpRequest()

request.onload = function() {
  console.log(this.responseText)
}

request.onerror = function() {
  alert('Gagal :(')
}

request.open('GET', 'https://some-api.com/json')
request.send()

// fetch API
fetch('https://some-api.com/json')
  .then(response => {
    console.log(response)
  })
  .catch(() => {
    alert('Gagal :(')
  })
})
```

Web APIs

Browser API yang Menarik untuk Diulik

- Bluetooth API
- Geolocation API
- Canvas API
- Image Capture API
- WebRTC
- Background Sync
- Push notification
- Web assembly

Ujian

Sesi Sharing

Sesi Diskusi

dicoding

Kampus
Merdeka
INDONESIA JAYA