

## Materi Pemrograman Web Materi

### Materi DOM / AJAX

Pengertian DOM :

**DOM (Dokumen Object Model)** antarmuka pemrograman (programming interface) yang digunakan untuk merepresentasikan dan berinteraksi dengan dokumen berbasis HTML atau XML. DOM memungkinkan web developer untuk berinteraksi dengan konten dan struktur halaman web sehingga aplikasi web yang dibuat dapat menjadi dinamis dan interaktif.

Konsep dari DOM :

**DOM (Document Object Model)** adalah konsep dalam pemrograman web yang merepresentasikan dokumen HTML atau XML sebagai struktur pohon (tree structure) di mana setiap bagian dari dokumen, seperti elemen, atribut, dan teks, dianggap sebagai objek atau node. DOM memungkinkan bahasa pemrograman seperti JavaScript untuk mengakses, membaca, dan memanipulasi isi, struktur, dan gaya dari halaman web secara dinamis. Melalui DOM, developer dapat menambahkan elemen baru, mengubah teks, merespons interaksi pengguna seperti klik atau input, serta membuat halaman web menjadi lebih interaktif tanpa perlu memuat ulang halaman. DOM adalah penghubung antara kode program dan tampilan halaman web yang ditampilkan di browser.

Fungsi dari DOM :

- DOM membantu developer untuk memperbarui konten dan struktur halaman web sebagai respons terhadap interaksi pengguna atau event lainnya. Misalnya, JavaScript dapat digunakan untuk menambah, menghapus, atau memodifikasi elemen HTML pada sebuah halaman.
- DOM bermanfaat dalam pembuatan aplikasi web interaktif yang dapat merespons input pengguna dan memberikan feedback secara real time. Sebagai contoh, seorang developer dapat menggunakan JavaScript untuk membuat animasi, formulir interaktif, atau game yang diperbarui secara real time saat pengguna berinteraksi dengan halamannya.
- DOM merupakan cara standar JavaScript untuk berinteraksi dengan dokumen HTML dan XML dan memastikan kompatibilitas lintas browser. Hal ini berarti bahwa kode yang ditulis untuk satu browser secara umum akan berfungsi di browser lain juga.
- DOM sangat penting dalam meningkatkan kecepatan loading website. Dengan mengizinkan penambahan atau modifikasi elemen menggunakan JavaScript, tugas tertentu dapat dilakukan lebih cepat, dan waktu muat halaman dapat dikurangi. Misalnya, menambahkan iframe secara dinamis dapat meningkatkan kecepatan loading halaman.

Properti/Komponen DOM :

- Document Object  
Mewakili dokumen HTML yang dimuat dalam window; memiliki banyak properti untuk mengakses elemen.
- Window Object  
Objek tertinggi di browser; dibuat otomatis dan digunakan untuk mengakses semua elemen global.
- Anchor Object  
Mewakili elemen `<a href>` dalam dokumen HTML.
- Link Object  
Mewakili elemen `<link>` dalam dokumen HTML, seperti untuk CSS.
- Form Object  
Mewakili elemen `<form>` dalam HTML, bisa diakses dan dimanipulasi lewat DOM
- Form Control Elements  
Mewakili elemen-elemen dalam form seperti input, button, checkbox, dll.

Contoh dari penerapan DOM :

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

  <h1 id="judul">Halo Dunia!</h1>
```

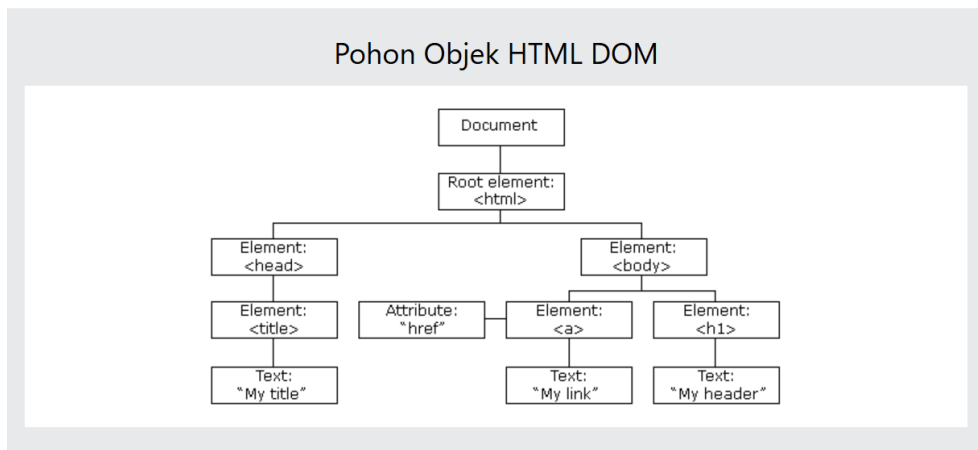
```

<button onclick="ubahJudul()">Klik untuk Ubah Judul</button>

<script>
  function ubahJudul() {
    document.getElementById("judul").textContent = "Judul Telah Diubah!";
  }
</script>

</body>
</html>

```



Element Dokumen DOM :

## Menemukan Elemen HTML

Method	Description
<code>document.getElementById(<i>id</i>)</code>	Find an element by element id
<code>document.getElementsByTagName(<i>name</i>)</code>	Find elements by tag name
<code>document.getElementsByClassName(<i>name</i>)</code>	Find elements by class name

## Mengubah Elemen HTML

Property	Description
<code>element.innerHTML = new html content</code>	Change the inner HTML of an element
<code>element.attribute = new value</code>	Change the attribute value of an HTML element
<code>element.style.property = new style</code>	Change the style of an HTML element
Method	Description
<code>element.setAttribute(<i>attribute</i>, <i>value</i>)</code>	Change the attribute value of an HTML element

## Menambahkan dan Menghapus Elemen

Method	Description
<code>document.createElement(<i>element</i>)</code>	Create an HTML element
<code>document.removeChild(<i>element</i>)</code>	Remove an HTML element
<code>document.appendChild(<i>element</i>)</code>	Add an HTML element
<code>document.replaceChild(<i>new</i>, <i>old</i>)</code>	Replace an HTML element
<code>document.write(<i>text</i>)</code>	Write into the HTML output stream

#### Pengertian AJAX :

AJAX (Asynchronous JavaScript and XML) adalah teknik dalam pengembangan web yang memungkinkan halaman web mengirim dan menerima data dari server secara asinkron, tanpa harus memuat ulang (reload) halaman. Dengan AJAX, web menjadi lebih cepat, interaktif, dan responsif, karena hanya sebagian data saja yang diperbarui.

#### Konsep dari AJAX :

Konsep AJAX adalah teknik dalam pengembangan web yang memungkinkan halaman web berkomunikasi dengan server untuk mengambil atau mengirim data secara asinkron, tanpa harus me-refresh (reload) seluruh halaman. Dengan AJAX, hanya bagian tertentu dari halaman yang diperbarui, sehingga pengalaman pengguna menjadi lebih cepat dan interaktif. Meskipun namanya mengandung “XML”, saat ini AJAX lebih sering digunakan dengan data format modern seperti JSON.

#### Fungsi AJAX :

1. Mengambil data dari server tanpa reload halaman  
AJAX memungkinkan pengguna melihat atau memuat data baru tanpa harus menyegarkan halaman.
2. Mengirim data ke server secara asinkron  
Data dari form atau input pengguna dapat dikirim ke server tanpa menghentikan aktivitas pengguna di halaman.
3. Memperbarui sebagian halaman secara real-time  
AJAX bisa mengubah isi bagian tertentu (seperti div atau tabel) secara langsung sesuai data terbaru.
4. Meningkatkan kecepatan dan interaktivitas web  
Karena tidak perlu reload halaman penuh, proses jadi lebih cepat dan terasa mulus bagi pengguna.
5. Membuat aplikasi web lebih user-friendly dan responsif  
Dengan tampilan yang terus diperbarui otomatis, pengguna merasa lebih nyaman dan tidak bosan menunggu.

#### Komponen AJAX :

- a. JavaScript – Mengatur proses pengiriman dan penerimaan data.
- b. XMLHttpRequest atau Fetch API – Objek yang digunakan untuk membuat permintaan HTTP ke server.
- c. Server – Tempat data diambil atau dikirim.
- d. Respons – Data dari server yang digunakan untuk memperbarui tampilan halaman.

#### Cara Kerja AJAX :

- a. AJAX Javascript akan dipanggil oleh browser guna mengaktifkan XMLHttpRequest dan mengirimkan HTTP Request ke server.
- b. Kemudian, XMLHttpRequest dibuat sebagai proses pertukaran data secara asinkron di server.
- c. Server akan menerima, lalu memprosesnya, dan mengirimkan data tersebut kembali ke browser.
- d. Browser kemudian menerima data tersebut dan langsung akan ditampilkan di halaman website tanpa harus melakukan reload.

#### Contoh dari AJAX :

```
<button onclick="ambilData()">Tampilkan Data</button>
<div id="hasil"></div>
```

```
<script>
  function ambilData() {
    fetch('https://jsonplaceholder.typicode.com/posts/1') // minta data ke server
      .then(response => response.json()) // ubah respons ke JSON
      .then(data => {
        document.getElementById('hasil').innerHTML =
`<h3>${data.title}</h3><p>${data.body}</p>`;
      });
  }
</script>
```

## DOM :

Keunggulan	Kekurangan
<b>Interaktivitas Tinggi:</b> Membuat halaman lebih interaktif tanpa me-reload halaman.	<b>Kompleksitas Pengelolaan:</b> Pengelolaan banyak elemen bisa rumit, terutama pada aplikasi besar.
<b>Manipulasi Dinamis:</b> Memungkinkan perubahan elemen secara dinamis menggunakan JavaScript.	<b>Kinerja Terpengaruh pada Halaman Besar:</b> Manipulasi DOM pada halaman besar dapat menurunkan kinerja.
<b>Kontrol Penuh terhadap Elemen:</b> Memberikan kontrol penuh untuk mengubah elemen, atribut, dan teks di halaman.	<b>Masalah Kompatibilitas Browser:</b> Perbedaan implementasi di browser dapat menyebabkan masalah kompatibilitas.
<b>Kinerja Lebih Cepat:</b> Memperbarui bagian halaman secara selektif lebih cepat daripada reload halaman penuh.	<b>Memerlukan Penguasaan JavaScript:</b> Membutuhkan pengetahuan JavaScript untuk memaksimalkan penggunaan DOM.
<b>Integrasi dengan JavaScript:</b> Memungkinkan interaksi dengan server menggunakan AJAX dan kontrol elemen lebih baik.	<b>Potensi Penggunaan Berlebihan:</b> Manipulasi DOM yang berlebihan bisa menyebabkan halaman lebih berat dan mengurangi kinerja.

## AJAX :

Keunggulan	Kekurangan
<b>Tidak Perlu Reload Halaman:</b> Memungkinkan pembaruan data tanpa muat ulang.	<b>Tidak SEO Friendly:</b> Konten yang dimuat lewat AJAX sering tidak terbaca oleh mesin pencari.
<b>Lebih Cepat dan Efisien:</b> Hanya bagian tertentu halaman yang dimuat ulang.	<b>Bergantung pada JavaScript:</b> Jika JavaScript dinonaktifkan di browser, AJAX tidak bisa digunakan.
<b>Responsif dan User-Friendly:</b> Memberi pengalaman pengguna yang lebih baik dan lebih interaktif.	<b>Kompleks untuk Debugging:</b> Kesalahan AJAX bisa lebih sulit dilacak dibanding request biasa.
<b>Mengurangi Beban Server:</b> Karena hanya mengambil data yang dibutuhkan, beban server lebih ringan.	<b>Masalah Keamanan:</b> Jika tidak hati-hati, bisa rentan terhadap serangan seperti XSS atau CSRF.
<b>Cocok untuk Aplikasi Real-Time:</b> Seperti chat, live search, atau notifikasi.	<b>Kompatibilitas Browser:</b> Beberapa fitur AJAX mungkin tidak berfungsi di browser lama.