



DOM

Gold - Chapter 4 - Topic 2

Selamat datang di **Chapter 4 Topic 2** online course
Full-stack Web dari Binar Academy!





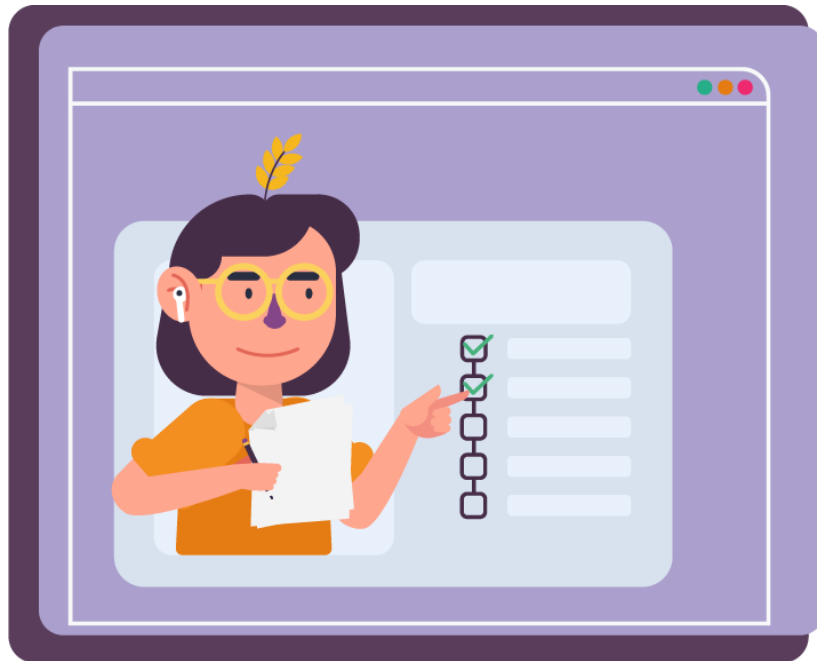
Pada topik kedua ini, kita bakal belajar **konsep DOM dan implementasinya pada pengembangan web**. Cus langsung aja~





Detailnya, kita bakal bahas hal-hal berikut ini:

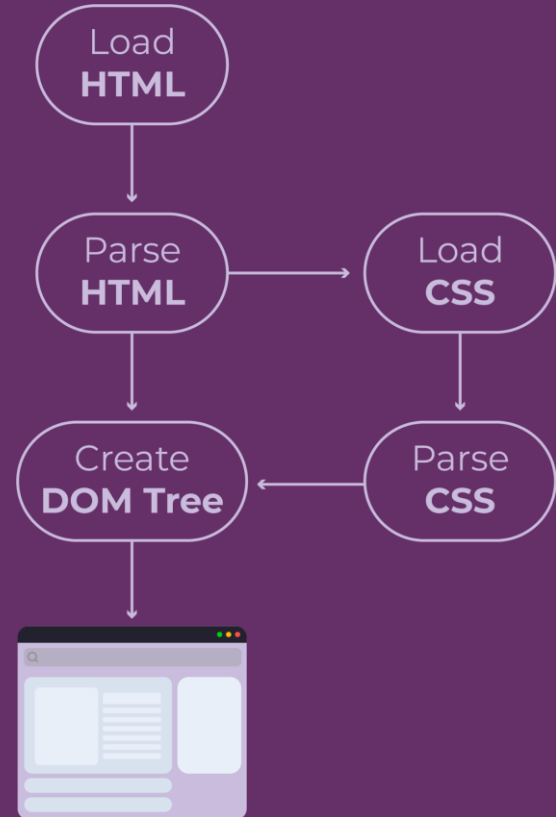
- Konsep DOM
- DOM di JavaScript
- DOM Selector
- DOM Manipulation





Kamu masih ingat kan sama gambar di samping? Kalo lupa, boleh cek lagi materi tentang HTML dan CSS yaa~

Nah, kalo kemarin kita bahas tentang HTML dan CSS dari gambar ini, sekarang kita bakal bahas tentang **DOM (Document Object Modeling)**. Skuyy!





Recap dulu, yuk!

Di chapter 1, kita udah belajar gimana HTML dan CSS bekerja untuk menampilkan dokumen pada web. Yaitu dengan mengubah HTML dan CSS menjadi DOM (Document Object Modeling).

DOM memiliki struktur seperti pohon. **Setiap element, attribute, dan potongan teks dalam bahasa markup, menjadi DOM node dalam struktur pohon.** Berbagai macam node akan berkumpul dan saling terhubung menjadi DOM. Beberapa elemen merupakan parent dari node anak (child), dan mungkin juga node anak memiliki saudara kandung (siblings).

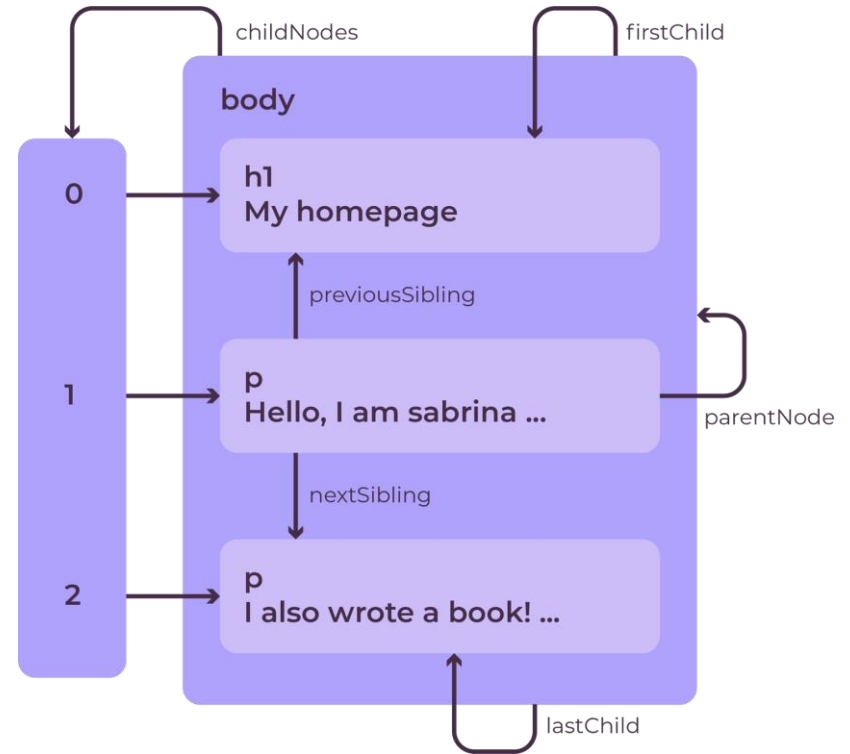




Parent, child, siblings, apa tuh?

Parent adalah induk dari suatu elemen yang membungkus beberapa elemen. Kalo child, elemen utama yang berada satu tingkat di bawah komponen parent. Sementara siblings, elemen turunan yang berasal dari elemen child utama.

Contohnya kayak gambar di samping ya~





Jadi, apa itu DOM?

DOM (Document Object Model) adalah **dokumen (HTML) yang dimodelkan dalam sebuah object yang menyediakan sekumpulan fungsi dan attribute/data yang bisa dimanfaatkan dalam membuat program JavaScript**. Fungsinya itu dikenal dengan sebutan **API (Application Programming Interface)**.

Ohiya, DOM nggak cuma untuk dokumen HTML aja, lho! DOM juga bisa digunakan untuk dokumen XML dan SVG serta bahasa pemrograman lain.

Tapi, berhubung kita lagi belajar JavaScript, jadi penggunaan DOM yang kita bahas kali ini akan berfokus pada JavaScript yaa~





Nah, kalo udah tahu apa itu DOM,
sekarang kita bahas gimana
penggunaan DOM di JavaScript.

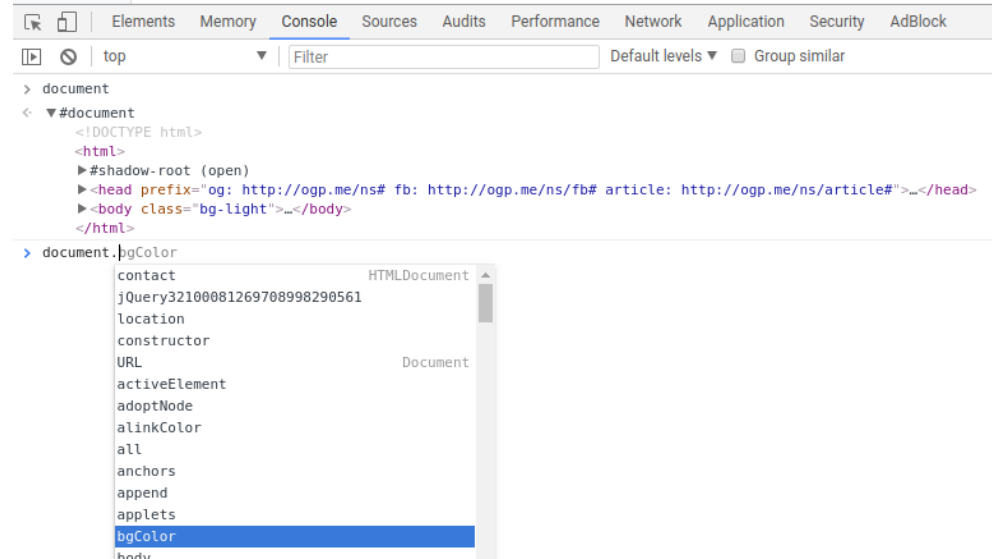




Kayak yang udah kita tahu, DOM adalah sebuah object untuk memodelkan dokumen HTML.

Nah, **pada JavaScript, object DOM itu bernama document.** Object DOM ini berisi segala hal yang kita butuhkan untuk melakukan manipulasi HTML.

Jadi, kalo kamu ketik “document” pada console JavaScript kayak contoh di samping, maka yang akan tampil adalah kode HTML.





Dalam object document, terdapat fungsi-fungsi dan atribut yang bisa kita gunakan untuk memanipulasi dokumen HTML. Contohnya fungsi `document.write()` yang digunakan untuk menulis sesuatu ke dokumen HTML.

Kalo kamu mengetik kode berikut ini di dalam console JavaScript Chrome:

```
document.write("Hello World");  
document.write("<h2>Saya Sedang Belajar Javascript</h2>");
```

Maka hasilnya akan langsung muncul pada dokumen HTML seperti gambar di bawah ini:

Hello World

Saya Sedang Belajar Javascript

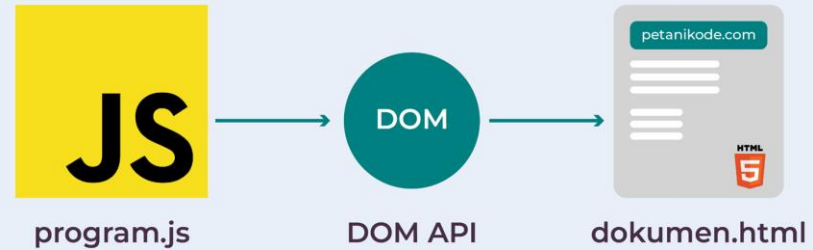


Udah tau kan gimana peran DOM dalam menampilkan dokumen HTML?

Jadi, **kita bisa menggunakan DOM ketika ingin:**

- membuat pengalaman pengguna (user experience) menjadi lebih baik;
- memanipulasi elemen HTML secara real time;
- menghindari refresh halaman setiap kali user ingin melakukan interaksi.

DOM: Tolong ganti warna background header web ini menjadi hijau





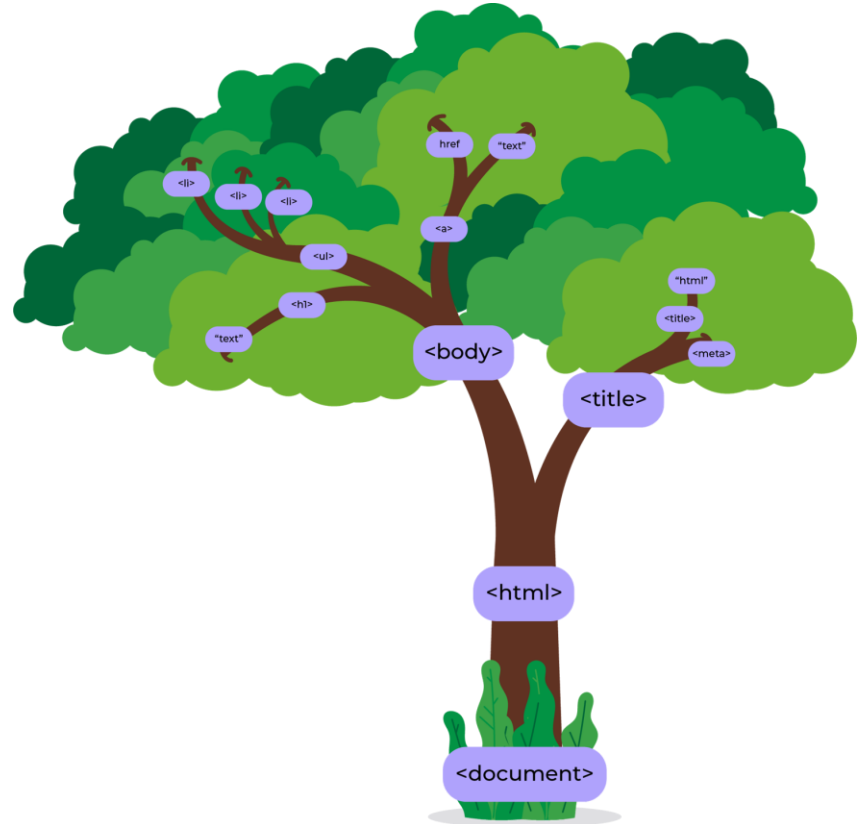
Terus, gimana ya caranya **mengakses element tertentu dengan DOM?**





Gambar di samping adalah contoh struktur DOM. Object document yang ada dalam DOM adalah representasi model dari dokumen HTML. Object ini berisi kumpulan fungsi dan atribut berupa object dari elemen HTML.

Nah, struktur ini akan memudahkan kita dalam menggunakan element tertentu dengan DOM. Dimana kita bisa mengakses element (DOM selector) dan membuat serta menghapus element (DOM manipulation).





Mengakses element (DOM Selector).

Jika kita mengetik kode seperti contoh yang ada pada gambar console di bawah ini, maka hasilnya kita bisa mengakses element tertentu yang kita inginkan (DOM selector). Misalnya `<head>`, `<body>`, dan lain-lain. Bahkan, kita juga bisa mengetahui panjang dari suatu element `<title>`.

```
> document.head
< ▶<head prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns/fb# article: http://ogp.me/ns/article#">...</head>

> document.body
< ▶<body class="bg-light" bgcolor="blue">...</body>

> document.title.length
< 66

>
```



Terus, jika kita ingin mengakses suatu element yang spesifik, terdapat beberapa fungsi yang bisa digunakan:

Kode di console	Fungsi
<code>getElementById()</code>	memilih element berdasarkan atribut id
<code>getElementByName()</code>	memilih element berdasarkan atribut name
<code>getElementsByClassName()</code>	memilih element berdasarkan atribut class
<code>getElementsByTagName()</code>	memilih element berdasarkan nama tag
<code>getElementsByTagNameNS()</code>	memilih element berdasarkan nama tag
<code>querySelector()</code>	memilih element berdasarkan query
<code>querySelectorAll()</code>	memilih element berdasarkan query



Biar kebayang, coba kita praktekin penggunaan fungsi yang tadi ya!

Misalnya, kita punya kode HTML seperti di samping ini~

Dalam kode tersebut, kita coba memilih element `<div>` yang memiliki id bernama "app". Lalu, kita ingin mengaksesnya dengan fungsi `getElementById()` dan menyimpannya ke dalam suatu variabel, yaitu `tutorial`.

Variabel `tutorial` akan menjadi sebuah object DOM dari element yang kita pilih.

Setelah itu, kita bisa melakukan hal apapun yang kita inginkan, seperti mengubah teks dan style CSS.

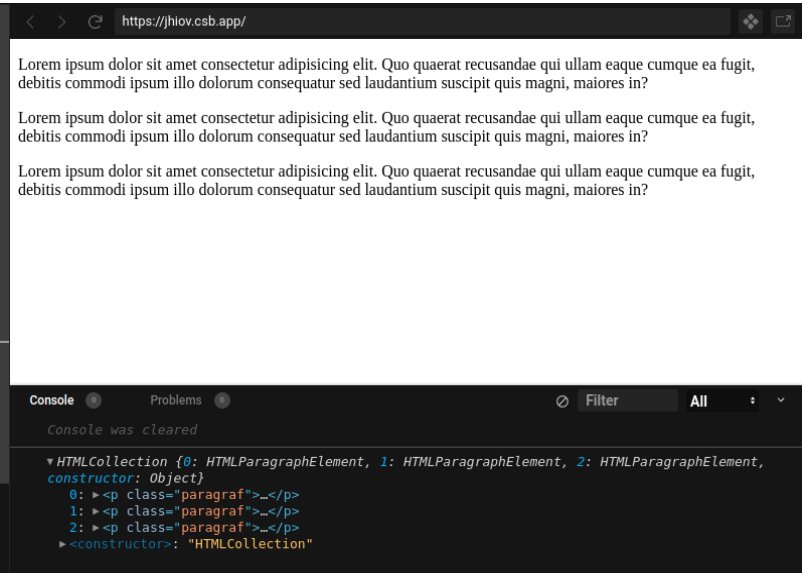
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Memilih Elemen Berdasarkan ID</title>
5     <meta charset="UTF-8" />
6   </head>
7   <body>
8     <!-- Elemen div yang akan kita pilih dari JS -->
9     <div id="app"></div>
10    <script type="text/javascript">
11      // mengakses elemen tutorial
12      var tutorial = document.getElementById("app");
13
14      // mengisi teks ke dalam elemen
15      tutorial.innerText = "Tutorial Javascript";
16
17      // memberikan CSS ke elemen
18      tutorial.style.backgroundColor = "gold";
19      tutorial.style.padding = "10px";
20    </script>
21  </body>
22 </html>
```



Tapi, gimana ya kalo ada lebih dari satu element yang dipilih?

Kalo kita mau pilih element berdasarkan nama tag atau attribute class, biasanya akan ada lebih dari satu element yang dipilih. Nah, element yang akan terpilih itu bakalan jadi sebuah array, karena kita memilih sekumpulan element. Array itu akan berisi objek DOM dari element-element yang terpilih. Biar lebih jelas, bisa diliat contoh kode di bawah ini yaa!

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Mengakses lebih dari satu elemen di DOM</title>
5     <meta charset="UTF-8" />
6   </head>
7   <body>
8     <p class="paragraf">
9       Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo
10      dolorum consequatur sed laudantium suscipit quis magni, maiores in?
11    </p>
12    <p class="paragraf">
13      Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat
14      recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo
15      dolorum consequatur sed laudantium suscipit quis magni, maiores in?
16    </p>
17    <p class="paragraf">
18      Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat
19      recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo
20      dolorum consequatur sed laudantium suscipit quis magni, maiores in?
21    </p>
22  </body>
23  <script>
24    var paragraf = document.getElementsByClassName("paragraf");
25    console.log(paragraf);
26  </script>
27 </html>
```





Pada contoh sebelumnya, kita memiliki tiga buah paragraf dengan nama class “paragraf”. Lalu, kita coba memilih ketiga paragraf tersebut melalui JavaScript dengan metode atau fungsi `getElementsByClassName()`. Hasilnya, variabel paragraf akan berisi sebuah array yang di dalamnya terdapat tiga buah objek DOM dari paragraf.



Oke, sekarang kita coba bereksperimen dengan mengubah warna teks dari paragraf pertama. Paragraf pertama akan berada pada posisi indeks ke-0 di dalam array. Jika kita mengetik perintah berikut ini di dalam console JavaScript, maka hasilnya paragraf pertama akan berwarna merah.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

```
Console Problems Filter All
Console was cleared

▼ HTMLCollection {0: HTMLParagraphElement, 1: HTMLParagraphElement, 2: HTMLParagraphElement, constructor: Object}
  0: ▶ <p class="paragraf">...</p>
  1: ▶ <p class="paragraf">...</p>
  2: ▶ <p class="paragraf">...</p>
  ▶ <constructor>: "HTMLCollection"

> paragraf[0].style.color = "red"

« red
```



Mari kita buat sebuah animasi warna dengan mengubah kode JavaScript pada halaman sebelumnya.

Kita akan memanfaatkan fungsi `setInterval()` dan fungsi `setTimeout()` untuk menentukan waktu animasinya. Pada kode ini, rentang waktu (interval) yang kita berikan adalah 1.000 milidetik, yang berarti 1 detik. Sedangkan untuk mengubah warnanya menjadi hitam, kita berikan waktu 500 milidetik atau 0.5 detik. Maka, hasilnya akan terjadi animasi warna.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Mengakses lebih dari satu elemen di DOM</title>
5 <meta charset="UTF-8" />
6 </head>
7 <body>
8 <p class="paragraf">
9   Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaeat
10   recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo
11   dolorum consequatur sed laudantium suscipit quis magni, maiores in?
12 </p>
13 <p class="paragraf">
14   Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaeat
15   recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo
16   dolorum consequatur sed laudantium suscipit quis magni, maiores in?
17 </p>
18 <p class="paragraf">
19   Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaeat
20   recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo
21   dolorum consequatur sed laudantium suscipit quis magni, maiores in?
22 </p>
23 <script>
24   var paragraf = document.getElementsByClassName("paragraf");
25   setInterval(function() {
26     paragraf[0].style.color = "red";
27     paragraf[1].style.color = "green";
28     paragraf[2].style.color = "blue";
29
30     setTimeout(function() {
31       paragraf[0].style.color = "black";
32       paragraf[1].style.color = "black";
33       paragraf[2].style.color = "black";
34     }, 500);
35   }, 1000);
36 </script>
37 </body>
38 </html>
```

Browser Tests

https://blor.csb.app/

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaeat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaeat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaeat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?



Nggak cuma untuk mengakses element aja, lho! DOM juga punya fungsi untuk **membuat element HTML (DOM Manipulation)**.





Membuat element (DOM Manipulation).

Salah satu fungsi untuk membuat element HTML adalah fungsi createElement().

```
document.createElement('p');
```

Dengan fungsi ini, maka akan tercipta element baru, yaitu paragraf. Tapi, hasilnya nggak akan ditampilkan ke dalam halaman web. Karena, kita belum menambahnya ke dalam body document.

Nah, untuk menambahnya ke body document, kita bisa gunakan fungsi append().



Nah, setelah kita menggunakan fungsi `append()`, baru deh hasilnya akan muncul di web browser.

```
Index.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Membuat element di DOM</title>
5     <meta charset="UTF-8" />
6   </head>
7   <body>
8     <script>
9       // membuat element h1
10      var judul = document.createElement("h1");
11
12      // mengisi content element
13      judul.textContent = "Belajar JavaScript";
14
15      // menambahkan element ke dalam tag body
16      document.body.append(judul);
17    </script>
18  </body>
19 </html>
```

Browser Tests
https://jhiov.csb.app/
Belajar JavaScript



Kalo tadi kita bisa membuat element, sekarang kita bisa menghapus element.

Sebelumnya, kita udah memakai fungsi `append()` untuk menambahkan element. **Untuk menghapusnya, kita bisa menggunakan fungsi `remove()`.**

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Menghapus element di DOM</title>
5     <meta charset="UTF-8" />
6   </head>
7   <body>
8     <h2 id="judul2">Delete Saya!</h2>
9   <script>
10    // memilih etement berdasarkan ID
11    var h2 = document.getElementById("judul2");
12
13    // menghapus etement yang sudah dipilih
14    h2.remove();
15
16    console.log("Element sudah dihapus");
17    console.log(h2);
18  </script>
19 </body>
20 </html>
```

Browser: Tests
https://jhiov.csb.app/

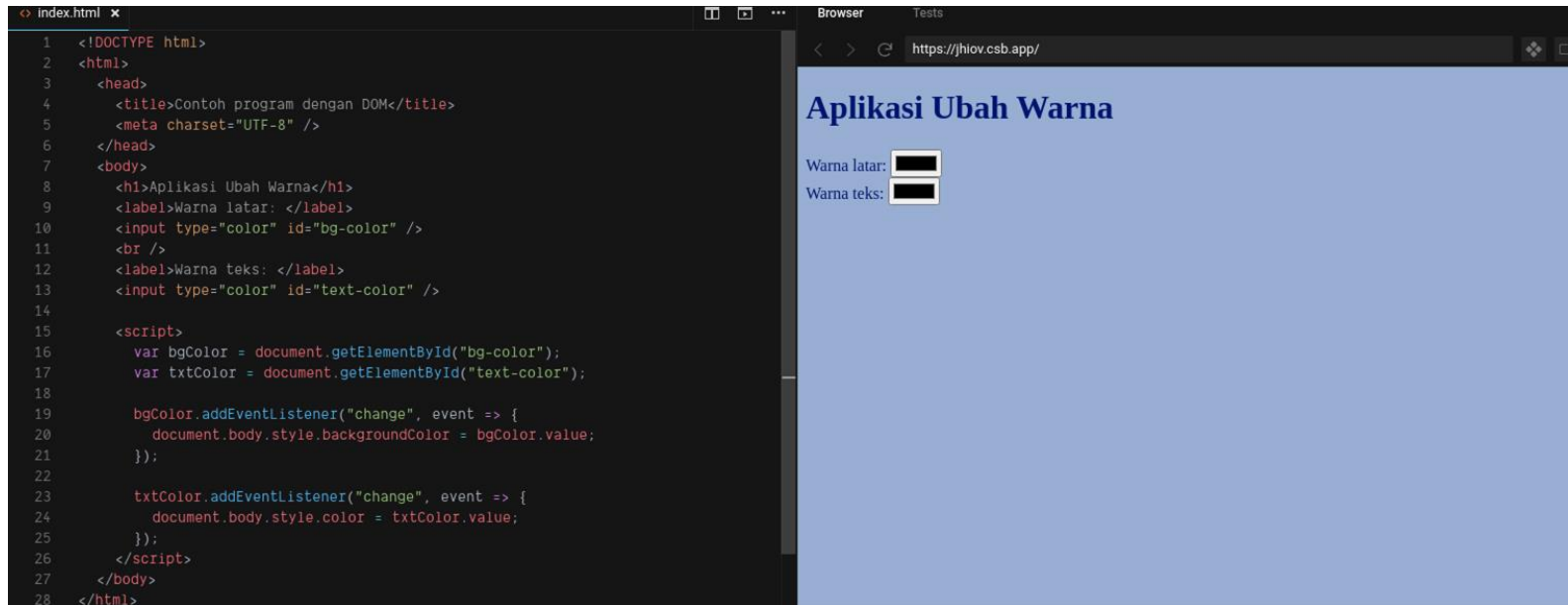
Console Problems Filter All
Console was cleared
Elemen sudah dihapus
<h2 id="judul2">Delete Saya!</h2>

Walaupun element berhasil dihapus dari halaman web, namun, element masih tersimpan dalam memori.

Biar makin mantap, Sabrina bakalan kasih **contoh pembuatan program yang memanfaatkan DOM di JavaScript**. Silahkan coba untuk diketik di laptop kamu yaa!



Program ini berfungsi untuk mengubah warna latar belakang dan warna teks dari suatu element. Untuk itu, kita akan menggunakan event "change" pada element bgColor dan txtColor. Maksudnya, ketika setiap nilai dari element ini berubah, maka kode di dalamnya akan dieksekusi.



The screenshot shows a code editor on the left and a web browser on the right. The code editor displays the HTML and JavaScript code for the application. The web browser shows the rendered application with a blue background and the title 'Aplikasi Ubah Warna'.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Contoh program dengan DOM</title>
5     <meta charset="UTF-8" />
6   </head>
7   <body>
8     <h1>Aplikasi Ubah Warna</h1>
9     <label>Warna latar: </label>
10    <input type="color" id="bg-color" />
11    <br />
12    <label>Warna teks: </label>
13    <input type="color" id="text-color" />
14
15    <script>
16      var bgColor = document.getElementById("bg-color");
17      var txtColor = document.getElementById("text-color");
18
19      bgColor.addEventListener("change", event => {
20        document.body.style.backgroundColor = bgColor.value;
21      });
22
23      txtColor.addEventListener("change", event => {
24        document.body.style.color = txtColor.value;
25      });
26    </script>
27  </body>
28 </html>
```

The browser window shows the URL <https://jhiov.csb.app/>. The page title is 'Aplikasi Ubah Warna'. The page content includes two labels: 'Warna latar:' and 'Warna teks:'. Each label is followed by a color selection input field. The background of the page is currently blue, and the text is dark blue.

Saatnya kita
Quiz!





1. DOM menyediakan sekumpulan fungsi dan attribute/data yang bisa dimanfaatkan dalam membuat program JavaScript yang dikenal dengan istilah...

- A. Document Object Model
- B. Application Programming Interface
- C. Node



1. DOM menyediakan sekumpulan fungsi dan attribute/data yang bisa dimanfaatkan dalam membuat program JavaScript yang dikenal dengan istilah...

A. Document Object Model

B. Application Programming Interface

C. Node

Sekumpulan fungsi dan attribute/data yang disediakan oleh DOM untuk membuat program JavaScript, dikenal dengan istilah **API (Application Programming Interface)**.



2. DOM API `getElementsByClassName()` digunakan untuk...

- A. Memilih element berdasarkan atribut id
- B. Memilih element berdasarkan atribut name
- C. Memilih element berdasarkan atribut class



2. DOM API `getElementByClassName()` digunakan untuk...

- A. Memilih element berdasarkan atribut id
- B. Memilih element berdasarkan atribut name
- C. Memilih element berdasarkan atribut class**

DOM API `getElementByClassName()` digunakan untuk memilih element berdasarkan atribut class.



3. Kode di samping digunakan untuk...

- A. Menampilkan element yang sudah dibuat ke body document.
- B. Memilih element berdasarkan query.
- C. Membuat element HTML baru.



```
document.body.append(judul);
```



3. Kode di samping digunakan untuk...

- A. Menampilkan element yang sudah dibuat ke body document.
- B. Memilih element berdasarkan query.
- C. Membuat element HTML baru.



```
document.body.append(judul);
```

DOM API `append()` digunakan untuk menampilkan element yang sudah dibuat ke dalam body document.



4. Sabrina ingin memilih elemen berdasarkan atribut name, sehingga dia menuliskan kode `getElementByTagName()` di console. Yang dilakukan Sabrina adalah

- A. Benar
- B. Salah



4. Sabrina ingin memilih elemen berdasarkan atribut name, sehingga dia menuliskan kode `getElementByTagName()` di console. Yang dilakukan Sabrina adalah

A. Benar

B. Salah

`getElementByTagName()` memiliki fungsi untuk memilih elemen berdasarkan nama tag



5. Dibawah ini yang merupakan kegunaan dari DOM adalah..

- A. memanipulasi database di dalam server yang nantinya akan ditampilkan di browser
- B. memanipulasi elemen HTML secara real time
- C. menghindari bugs pada halaman website yang dikembangkan



5. Dibawah ini yang merupakan kegunaan dari DOM adalah..

A. memanipulasi database di dalam server yang nantinya akan ditampilkan di browser

B. memanipulasi elemen HTML secara real time

C. menghindari bugs pada halaman website yang dikembangkan

Beberapa fungsi DOM, diantaranya membuat pengalaman pengguna (user experience) menjadi lebih baik; memanipulasi elemen HTML secara real time; menghindari refresh halaman setiap kali user ingin melakukan interaksi.



Referensi dan bacaan lebih lanjut~

1. https://www.w3schools.com/js/js_htmlDOM.asp
2. https://developer.mozilla.org/en-US/docs/Web/API/HTML_DOM_API
3. <https://www.petanikode.com/javascript-dom/>
4. https://eloquentjavascript.net/14_dom.html
5. <https://blog.10pines.com/2018/08/27/reactjs-virtual-dom/>





Nah, selesai sudah pembahasan kita di Chapter 4 Topic 2 ini.

Setelah ini, **kita akan lanjut ke Chapter 4 Topic 3** untuk membahas **Node.js** !

Sayonara~



Terima Kasih!



Next Topic

loading...