

MODUL LAB MINGGU 2 - MANIPULASI DOM HTML

1.1. Tujuan Instruksi Khusus

Setelah mengikuti modul ini, mahasiswa diharapkan mampu:

- a. Memahami konsep DOM HTML
- b. Mampu mendapatkan value element DOM HTML
- c. Mampu memanipulasi DOM HTML

1.2. Dasar Teori

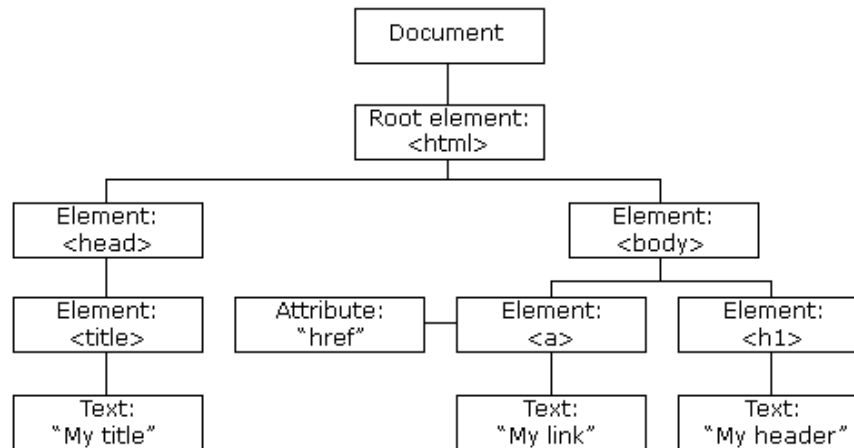
a) Javascript

Javascript adalah bahasa pemrograman yang digunakan untuk membuat sebuah halaman web (*web page*) lebih interaktif. Javascript bersifat client-side programming, artinya javascript dijalankan di sisi client yaitu web browser. Tanpa server, javascript masih tetap bisa dijalankan karena hanya membutuhkan web browser. Javascript dapat membuat *web page* lebih interaktif karena dapat berhubungan atau berinteraksi langsung dengan dokumen HTML. Cara javascript melihat suatu halaman HTML adalah melalui DOM.

b) DOM

DOM adalah singkatan dari *Document Object Model*. DOM adalah sebuah platform dan interface yang memperbolehkan pengaksesan dan perubahan pada konten, struktur, dan style pada sebuah dokumen oleh program dan script. Istilah HTML DOM mengacu kepada dokumen html. Kasusnya disini ialah konten, struktur, dan style pada dokumen html dapat diakses dan dirubah dengan menggunakan sintaks javascript.

Pada model DOM ini, setiap elemen html dipandang sebagai sebuah object. Setiap object bisa terdiri dari object-object lain, sama halnya dengan dokumen html yang terdiri dari elemen root (elemen <html>), elemen root terdiri dari elemen <head> dan elemen <body>, elemen <body> boleh jadi terdiri dari elemen <a>, <h1>, <p>, dst. Elemen-elemen pada dokumen html membentuk sebuah object document yang merupakan object dari dokumen html itu sendiri. Dibawah ini adalah representasi HTML DOM dalam model pohon.



Elemen pada dokumen html memiliki property yaitu nilai pada elemen html yang bisa ditetapkan atau dirubah dan method yakni aksi yang dapat dilakukan pada elemen html. Contoh property yang biasa digunakan ialah innerHTML yang digunakan untuk memperoleh atau merubah konten dari elemen html itu sendiri. Kemudian method, contoh misalnya getElementById("id") milik object document yang digunakan untuk mengakses elemen html dalam dokumen html berdasarkan id. Terakhir adalah event, yaitu sesuatu kejadian yang terjadi pada atau dialami oleh element HTML, misal klik pada suatu button, hover mouse pada suatu link, halaman HTML selesai di load, dll.

Yang bisa dilakukan oleh javascript melalui HTML DOM adalah :

- Merubah setiap elemen HTML pada halaman web.
- Merubah setiap atribut HTML pada halaman web.
- Merubah setiap style CSS pada halaman web.
- Menghilangkan elemen dan atribut HTML yang ada.
- Menambahkan elemen dan atribut HTML baru.
- Bereaksi pada setiap event yang ada pada halaman web.
- Membuat event baru pada halaman web.

1.3. Instruksi Modul

1.3.1. Mendapatkan value 1 Element DOM HTML

- 1) Buka file **latihan manipulasi DOM.html**, lalu di barisan sebelum tag penutup `</body>`, tambahkan kode untuk memanggil file javascript. Gunakan sintaks :

```
<script type="text/javascript" src="dom1val.js"></script>
```

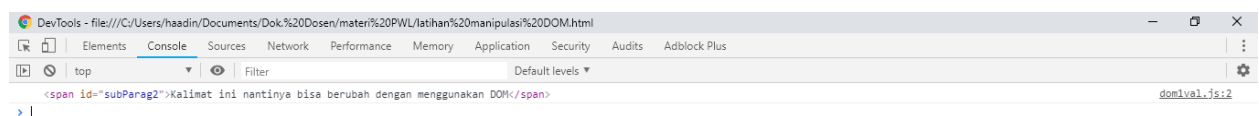
- 2) Buatlah file javascript baru, beri nama **dom1val.js** (sesuai nama file yang dipanggil melalui HTML)
- 3) Untuk mendapatkan value 1 element HTML bisa menggunakan 2 perintah, yaitu **document.getElementById()** dan **document.querySelector()**.
- 4) **document.getElementById()** digunakan untuk memanggil element HTML sesuai atribut ID-nya. ID harus unik dan tidak boleh sama dengan element lain. Misal, kita akan memanggil element HTML yang ber-ID “**subParag2**” maka sintaks yang digunakan adalah:

```
document.getElementById('subParag2');
```

untuk mengetahui value DOM dari element tersebut, maka kita bisa memanfaatkan `console.log()`. Tambahkan perintah berikut ke dalam file **dom1val.js** :

```
var element = document.getElementById('subParag2');  
console.log(element);
```

lalu buka file **latihan manipulasi DOM.html** menggunakan web browser (chrome misalnya). Klik kanan dan masuk ke Inspect (CTRL+SHIFT+I) dan pilih tab console. Jika perintah benar, maka console akan mengeluarkan / menampilkan value DOM dari element yang kita panggil seperti gambar berikut.



- 5) **document.querySelector()** digunakan untuk memanggil element HTML layaknya suatu CSS memanggil element HTML. Sama halnya dengan CSS, terdapat 3 cara untuk memanggil element HTML yaitu dengan menggunakan *class*, *id*, atau *tag* HTML. Contohnya adalah seperti berikut :

- Memanggil element yang memiliki *class* **paragraf** :

```
document.querySelector('.paragraf')
```

- Memanggil element yang ber-ID **subParag2** :

```
document.querySelector('#subParag2');
```

- Memanggil element dengan tag **p** :

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

- 6) Tambahkan perintah berikut ini ke file **dom1val.js** :

```
var element7 = document.querySelector('#subParag2');  
console.log(element7);  
  
var element8 = document.querySelector('.paragraf');  
console.log(element8);  
  
var element9 = document.querySelector('p');  
console.log(element9);
```

- 7) Reload file **html** di browser dan console pun akan terupdate menampilkan data DOM HTML yang sudah dipanggil.

1.3.2. Mendapatkan value lebih dari 1 Element DOM HTML

- 1) Buka file **latihan manipulasi DOM.html**, lalu di barisan sebelum tag penutup `</body>`, tambahkan kode untuk memanggil file javascript. Gunakan sintaks :

```
<script type="text/javascript" src=" dom2val.js"></script>
```

- 2) Buatlah file javascript baru, beri nama **dom2val.js** (sesuai nama file yang dipanggil melalui HTML)
- 3) Untuk mendapatkan value >1 element HTML bisa menggunakan 3 perintah, yaitu ***document.getElementsByClassName()***, ***document.getElementsByTagName()***, dan ***document.querySelectorAll()***.
- 4) ***document.getElementsByClassName()*** digunakan untuk memanggil element HTML sesuai atribut *class*-nya. Misal, kita akan memanggil semua element HTML yang memiliki *class* “**paragraf**” maka sintaks yang digunakan adalah:

```
document.getElementsByClassName('paragraf');
```

untuk mengetahui value DOM dari element tersebut, maka kita bisa memanfaatkan `console.log()`. Tambahkan perintah berikut ke dalam file **dom2val.js** :

```
var element2 = document.getElementsByClassName('paragraf');  
console.log(element2);
```

- 5) ***document.getElementsByTagName()*** digunakan untuk memanggil element HTML sesuai nama tag element-nya. Misal, kita akan memanggil semua element HTML yang memiliki tag “li” maka sintaks yang digunakan adalah:

```
document.getElementsByTagName('li');
```

untuk mengetahui value DOM dari element tersebut, maka kita bisa memanfaatkan `console.log()`. Tambahkan perintah berikut ke dalam file **dom2val.js** :

```
var element3 = document.getElementsByTagName('li');  
console.log(element3);
```

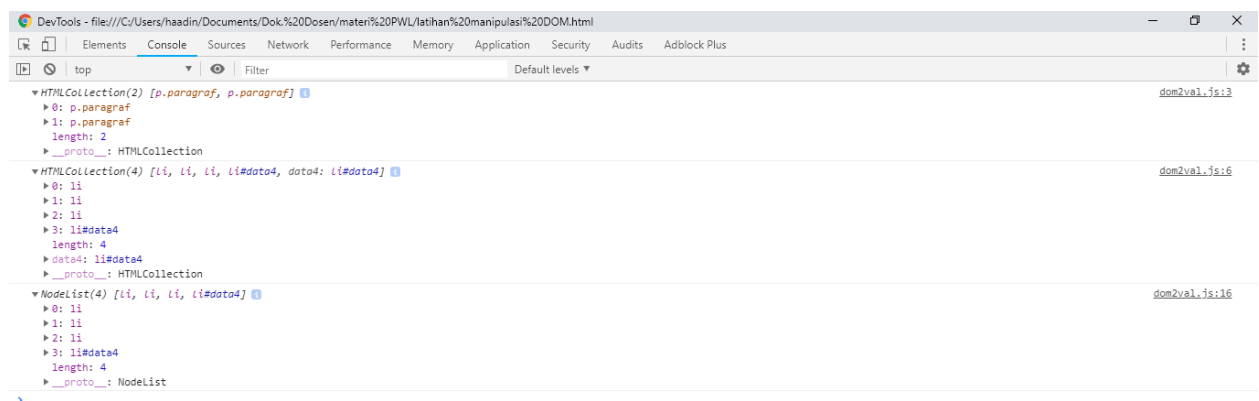
- 6) ***document.querySelectorAll()*** hampir sama dengan ***document.querySelector()***. Bedanya adalah ***document.querySelectorAll()*** memiliki output element lebih dari 1. Contohnya memanggil element dengan tag **li** :

```
document.querySelectorAll("li");
```

Tambahkan perintah berikut ini ke file **dom2val.js** :

```
var element10 = document.querySelectorAll("li");  
console.log(element10);
```

- 7) Reload file **html** di browser dan console pun akan terupdate menampilkan data DOM HTML yang sudah dipanggil seperti gambar berikut.



1.3.3. Mendapatkan value pada sub Element DOM HTML

- 1) Ada kalanya terdapat 2 atau lebih element HTML yang memiliki *class* atau *tag* yang sama, tapi kita hanya ingin memilih elemen HTML tertentu saja, tidak ingin mendapatkan semua element tersebut. Menggunakan javascript, kita dapat melakukannya.

- 2) Buka file **latihan manipulasi DOM.html**, lalu di barisan sebelum tag penutup `</body>`, tambahkan kode untuk memanggil file javascript. Gunakan sintaks :

```
<script type="text/javascript" src="domSubEl.js"></script>
```

- 3) Buatlah file javascript baru, beri nama **domSubEl.js** (sesuai nama file yang dipanggil melalui HTML)
- 4) Untuk mendapatkan sub element HTML tertentu bisa menggunakan perintah yang sama seperti perintah-perintah sebelumnya, bedanya adalah jika sebelumnya menggunakan obyek **“document”** maka sekarang untuk mendapatkan sub element harus menggunakan nama variabel/obyek DOM *parent*.
- 5) Misal, kita akan memanggil element HTML yang memiliki *class* **“paragraf”** tapi hanya yang berada didalam *id* **paragraf1** maka sintaks yang digunakan adalah :

```
var element5 = document.getElementById('paragraf1');  
var element6 = element5.getElementsByClassName('paragraf');  
console.log(element6);
```

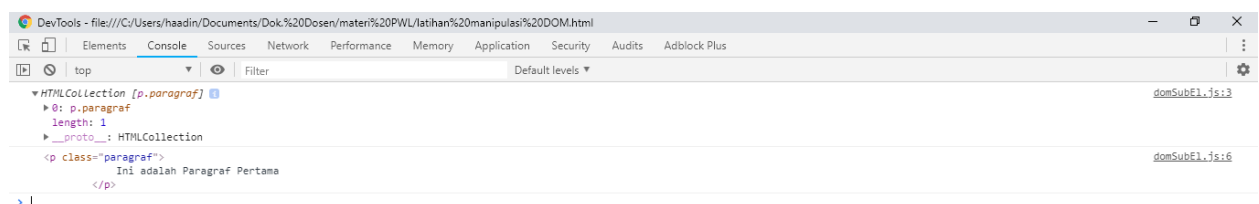
Tambahkan perintah tersebut ke dalam file **domSubEl.js**.

- 6) Reload file **html** di browser dan console pun akan terupdate menampilkan data DOM HTML yang sudah dipanggil.
- 7) Selain cara diatas, kita juga bisa menggunakan perintah **document.querySelectorAll()** atau **document.querySelector()**. Untuk memanggil element HTML yang memiliki *class* **“paragraf”** tapi hanya yang berada didalam *id* **paragraf1** maka sintaks yang digunakan adalah :

```
var element11 = document.querySelector('#paragraf1 .paragraf');  
console.log(element11);
```

Tambahkan perintah tersebut ke dalam file **domSubEl.js**.

- 8) Reload file **html** di browser dan console pun akan terupdate menampilkan data DOM HTML yang sudah dipanggil.



1.3.4. Manipulasi Konten HTML / Text Element DOM HTML

- 1) Buka file **latihan manipulasi DOM.html**, lalu di barisan sebelum tag penutup `</body>`, tambahkan kode untuk memanggil file javascript. Gunakan sintaks :

```
<script type="text/javascript" src="domKonten.js"></script>
```

- 2) Buatlah file javascript baru, beri nama **domKonten.js** (sesuai nama file yang dipanggil melalui HTML)
- 3) Misal, kita akan mengubah konten element HTML yang memiliki *id* "**subParag2**". Pertama-tama kita harus menampung obyek HTML kedalam suatu variabel, lalu kemudian menset ulang properti konten obyek tersebut. Perintahnya adalah sebagai berikut :

```
var element = document.getElementById('subParag2');  
element.innerHTML = 'Henshin ... Berubah';
```

Tambahkan perintah tersebut ke dalam file **domKonten.js**.

- 4) Reload file **html** di browser dan perhatikan pada paragraf kedua, maka konten HTML nya akan berubah sesuai yang kita set pada javascript.
- 5) Selain merubah sepenuhnya, kita juga bisa menambahkan konten baru selain konten lama yang ada di HTML. Sama seperti cara sebelumnya, kita harus menampung obyek HTML terlebih dahulu ke suatu obyek, lalu kita menampung nilai/value konten HTML tersebut ke dalam variabel lain. Selanjutnya gabungkan variabel konten HTML dengan konten baru. Perintahnya sebagai berikut :

```
var element = document.getElementById('subParag2');  
var konten = element.innerHTML;  
element.innerHTML = konten + ' => Henshin ... Berubah';
```

- 6) Reload file **html** di browser dan perhatikan pada paragraf kedua, maka konten HTML nya akan berubah sesuai yang kita set pada javascript.
- 7) Untuk merubah konten lebih dari 1 element, bisa menggunakan perulangan. Contoh perintahnya :

```
var element8 = document.getElementsByClassName('paragraf');  
  
for (var i = 0; i < element8.length; i++) {  
    element8[i].innerHTML = 'Semua Class Paragraf Jadi Seragam';  
}
```

- 8) Reload file **html** di browser dan perhatikan pada semua paragraf (yang memiliki *class* **paragraf**), konten HTML nya akan berubah sesuai yang kita set pada javascript.

1.3.5. Manipulasi Style CSS Element DOM HTML

- 1) Buka file **latihan manipulasi DOM.html**, lalu di barisan sebelum tag penutup `</body>`, tambahkan kode untuk memanggil file javascript. Gunakan sintaks :

```
<script type="text/javascript" src="domStyle.js"></script>
```

- 2) Buatlah file javascript baru, beri nama **domStyle.js** (sesuai nama file yang dipanggil melalui HTML)
- 3) Misal, kita akan mengubah style element HTML yang memiliki *id* "**subParag2**". Pertama-tama kita harus menampung obyek HTML kedalam suatu variabel, lalu kemudia menset ulang properti style obyek tersebut. Perintahnya adalah sebagai berikut :

```
var element = document.getElementById('subParag2');  
element.style = 'background-color:pink; font-size:20px';
```

Tambahkan perintah tersebut ke dalam file **domStyle.js**.

- 4) Reload file **html** di browser dan perhatikan pada paragraf kedua, maka style CSS HTML nya akan berubah sesuai yang kita set pada javascript.
- 5) Untuk merubah style CSS lebih dari 1 element, bisa menggunakan perulangan. Contoh perintahnya :

```
var element8 = document.getElementsByClassName('paragraf');  
  
for (var i = 0; i < element8.length; i++) {  
    element8[i].style = 'background-color:pink; font-size:20px';  
}
```

- 6) Reload file **html** di browser dan perhatikan pada semua paragraf (yang memiliki *class* **paragraf**), konten HTML nya akan berubah sesuai yang kita set pada javascript.