

Modul Praktikum 7

PEMROGRAMAN WEB

HTML DOM (Document Object Model)

A. Tujuan

1. Dapat mengganti konten-konten dari elemen-elemen HTML
2. Dapat merubah atau mengganti Style (CSS) dari elemen HTML
3. Dapat memahami Event-event dari HTML DOM
4. Data menambahkan dan menghapus elemen-elemen HTML

B. Alat dan Bahan

Alat dan bahan yang digunakan dalam praktikum adalah :

1. Satu unit PC (Personal Computer)
2. Sistem Operasi
3. Notepad (terintegrasi dengan Windows)
4. Web browser Mozilla Firefox/Chrome/Safari

C. Landasan Teori

1. Pengantar HTML DOM

HTML DOM merupakan standar W3C dalam pengelolaan dokumen web berbasis HTML. Di dalam HTML, DOM mendefinisikan sebuah pengaturan standar HTML sebagai objek sehingga kita di beri keluasaan untuk mengakses dan memanipulasi dokumen HTML. Semua elemen HTML, sepanjang memiliki atribut dan isi, dapat diakses dengan DOM. Isi dapat diubah, dihapus, atau dibuat baru.

HTML DOM merupakan suatu platform yang independent, yang dapat digabungkan dengan bahasa pemrograman lain seperti *java*, *JavaScript*, dan *VBScript*. DOM dapat digunakan dengan *JavaScript* untuk membaca dan mengubah dokumen *HTML*, *XHTML*, dan *XML*. Dasar script yang harus dikuasai adalah *HTML/XHTML* dan *JavaScript*.

DOM dibagi menjadi beberapa bagian yaitu:

1. **Core DOM**: untuk mendefinisikan sebuah standar obyek pada struktur dokumen.
2. **XML DOM**: untuk mendefinisikan sebuah standar obyek pada dokumen XML.
3. **HTML DOM**: untuk mendefinisikan sebuah standar obyek pada dokumen XML.

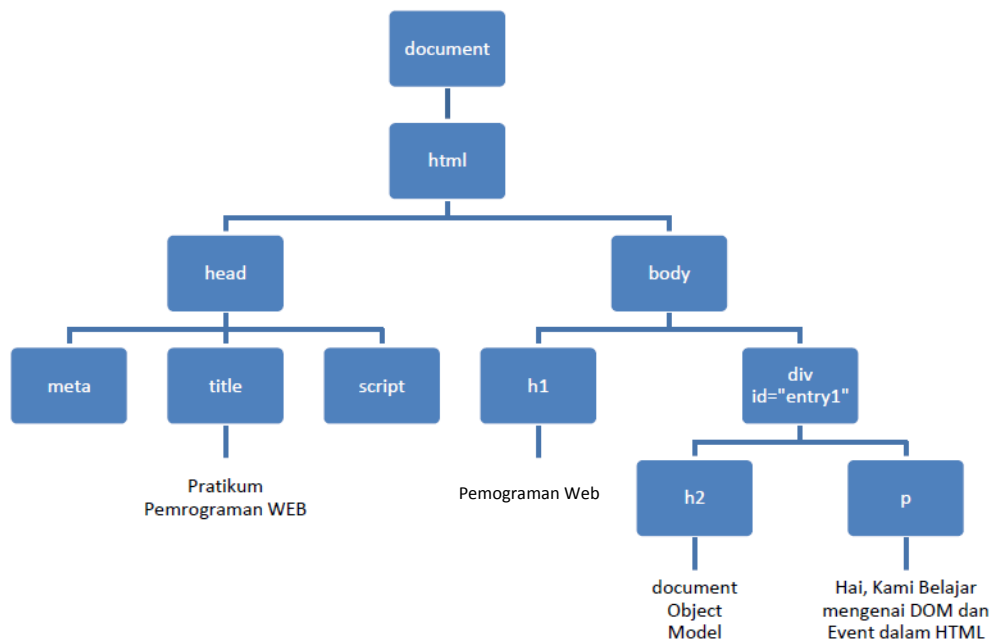
DOM membuat sebuah dokumen HTML sebagai **tree-structure** dengan elemen, atribut dan text, sehingga dengan DOM dapat dikatakan:

1. Semua yang ada di dalam dokumen HTML adalah sebuah node.
2. Setiap tag HTML adalah sebuah elemen node.
3. Teks dalam HTML adalah teks node.
4. Setiap atribut HTML adalah atribut node.
5. Comment adalah comment node.

2. Hirarki HTML DOM

Sebuah dokumen HTML direpresentasikan oleh DOM dalam bentuk struktur hirarki pohon. Dengan kata lain, cara DOM melihat dokumen HTML seperti struktur hirarki pohon. Berikut contoh struktur DOM dari sebuah dokumen HTML dibawah ini.

```
<!DOCTYPE html>
<html lang="eng">
<head>
<title>Praktikum Pemograman Web</title>
<meta charset="utf-8">
<script src="script.js"></script>
</head>
<body>
<h1>Pemograman Web</h1>
<div id="entry1">
  <h2>Pemograman Web</h2>
  <p>
    Hai,
    Kami belajar mengenai DOM dan EVENTS
    dalam HTML
  </p>
</div>
</body>
</html>
```



Gambar 1. HTML direpresentasikan dengan Struktur Tree

Perhatikan , terdapat dua jenis elemen pada pohon : *node* (simpul) ditampilkan dalam bentuk kotak biru, serta teks ditampilkan dalam bentuk tulisan. Setiap elemen HTML berupa node, yang dapat membungkus elemen HTML lainnya / membungkus node lainnya. Sedangkan elemen teks tidak dapat memiliki elemen lainnya.

3. Mengakses Elemen DOM

Elemen **document** selalu berada diatas, seperti akar dari sebuah pohon. Elemen **document** dapat digunakan dalam *Javascript* untuk dapat mengakses semua elemen DOM. Konsep DOM dalam *Javascript* adalah memandang elemen dokumen sebagai suatu objek. Seperti halnya konsep OOP, objek ini bisa memuat method dan properties. Selanjutnya untuk pemrosesan objek-objek dalam dokumen HTML menggunakan method dan properties tersebut.

- Method merupakan aksi yang dapat dilakukan oleh objek, seperti menambah dan menghapus elemen HTML,
- Properties merupakan nilai dari elemen HTML , seperti mengubah konten(nilai) dari sebuah elemen HTML

Contoh :

```
<!DOCTYPE html>
<html>
<head>
<title>Contoh Akses HTML DOM</title>
</head>
<body>
<h2>Pemograman Web HTML DOM</h2>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

Hasilnya yang akan ditampilkan adalah sebagai berikut:

Pemograman Web HTML DOM

Hello World!

Gambar 2. Contoh Hasil akses elemen HTML DOM

Pada contoh diatas **getElementById** adalah *method* , sedangkan **innerHTML** adalah property. **getElementById** digunakan untuk mengakses elemen DOM berdasarkan nilai properti id pada elemen. berikut ini adalah beberapa method yang digunakan untuk mengakses elemen DOM :

- `document.getElementById`, digunakan untuk mengakses elemen DOM berdasarkan nilai **properti id** pada elemen. Hanya akan mengembalikan satu elemen DOM karena idealnya nilai properti id bersifat unik dalam satu dokumen. Jika terdapat lebih dari satu elemen dengan id yang sama, maka elemen yang pertama ditemukan akan dikembalikan.
- `document.getElementsByClassName`, seperti namanya digunakan untuk memanggil seluruh elemen DOM dengan nilai properti class yang diberikan.

Mengembalikan objek mirip array, yaitu *HTMLCollection* dari seluruh elemen DOM yang cocok.

- c) `document.getElementsByName`, mengambil elemen berdasarkan nilai *properti name*. Mengembalikan *HTMLCollection* seperti `getElementsByClassName`.
- d) `document.getElementsByTagName`, memilih elemen-elemen dengan tag HTML tertentu. Sama seperti semua fungsi yang mengembalikan banyak elemen DOM, memberikan nilai berupa *HTMLCollection*.
- e) `document.getElementsByTagNameNS`, sama seperti `getElementsByTagName`, hanya saja fungsi ini melakukan penyaringan tambahan berdasarkan namespace pada dokumen XHTML.
- f) `document.querySelector`, mencari elemen DOM pertama yang sesuai dengan aturan *selector* CSS yang diberikan ke fungsi.
- g) `document.querySelectorAll`, sama seperti *querySelector*, tetapi mengembalikan semua elemen yang memenuhi aturan (bukan hanya elemen pertama).

4. Menghapus Elemen DOM

Penghapusan elemen DOM dilakukan dengan sangat sederhana, hanya melalui satu fungsi saja: `removeChild`. Prosesnya sederhana: pilih elemen yang akan dihapus, kemudian panggil `removeChild` pada penampung elemen yang akan dihapus tersebut:

```
var hapus = document.querySelector("li");
ul.removeChild(hapus);
```

5. Mengganti Tampilan Elemen DOM

Salah satu cara yang paling sederhana untuk mengganti properti CSS adalah dengan memanfaatkan atribut `style` pada sebuah elemen. Karena atribut *style* akan menerapkan properti CSS sesuai dengan nilai yang diberikan. Contoh penggunaan :

```
var x = document.getElementsByTagName("li");
x.style.background = "Red";
```

D. Praktikum

Praktikum kali ini, kita akan membuat **playlist lagu** menggunakan HTML DOM.

Instruksi:

1. Buatlah Folder dengan nama **4digit-terakhir-NIM-anda**. Misalnya folder **1234**.
2. Buka text editor Notepad++.
3. Simpan terlebih dahulu file tersebut dengan nama project sesuai dengan nomor **4 digit NIM terakhir** anda masing-masing. Contoh : **1234.html** (*Note. Gunakan ketentuan ini untuk semua file praktikum ini, contoh : 1234.html , 1234.js dan 1234.css*)
4. Selanjutnya yang kita lakukan adalah membuat dokumen HTML dengan sebuah **form** dan elemen *list* yang nantinya akan menampung playlist lagu yang diinputkan, seperti yang terlihat pada Gambar 3 berikut.

```

<!DOCTYPE html>
<html>
<head>
  <title>Praktikum 7 - Playlist Lagu</title>
  <script src="1234.js"></script>
  <link rel="stylesheet" href="1234.css">
</head>
<body>
  <h2>Playlist Lagu</h2>
  <form>
    <input type="text" id="songTextInput" size="50" placeholder="Nama Lagu">
    <input type="button" id="addButton" value="Tambah Lagu">
  </form>
  <ul id="playlist"></ul>
</body>
</html>

```

Gambar 3. Html Source Code

Jadi kita akan mempraktikkan cara menambahkan elemen baru pada dokumen HTML dengan menggunakan *javascript*, yang membuat dokumen HTML menjadi lebih interaktif.

Di dalam terdapat sebuah element input dengan tipe text filed untuk mengetikkan judul lagu dan atribut placeholder yang menampilkan penjelasan dari kolom input. dan elemen input lainnya dengan tipe button untuk meng-submit judul lagu yang baru ditambahkan.

Playlist yang diinputkan akan muncul dalam bentuk list untuk itu kita membutuhkan elemen list ``, saat ini list lagu masih kosong tapi kita akan mengisinya dengan *Javascript* pada instruksi selanjutnya. Jika script diatas dijalankan akan terlihat seperti gambar dibawah ini.

Playlist Lagu

5. Tambahkan aksi pada button “Tambah Lagu” dengan souce Code Javascript seperti berikut:

```

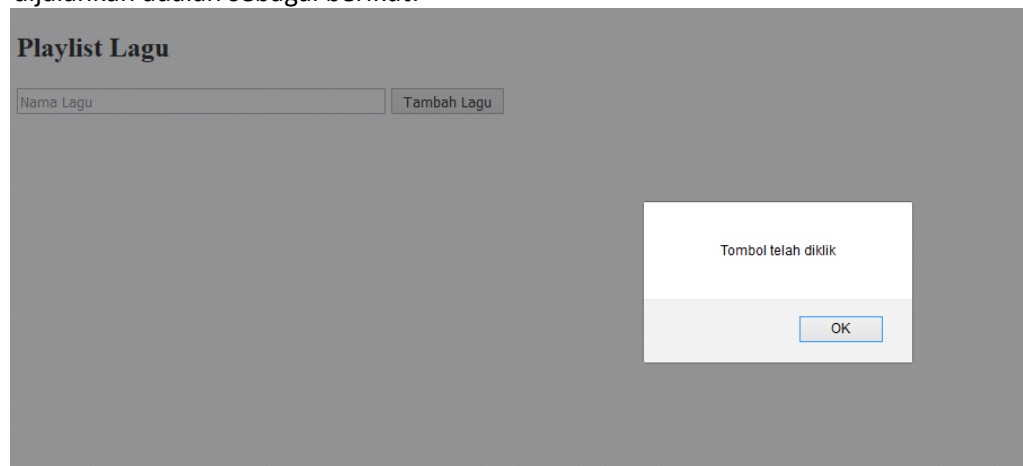
window.onload = init;

function init(){
  var button = document.getElementById("addButton");
  button.onclick=handleButtonClick;
}

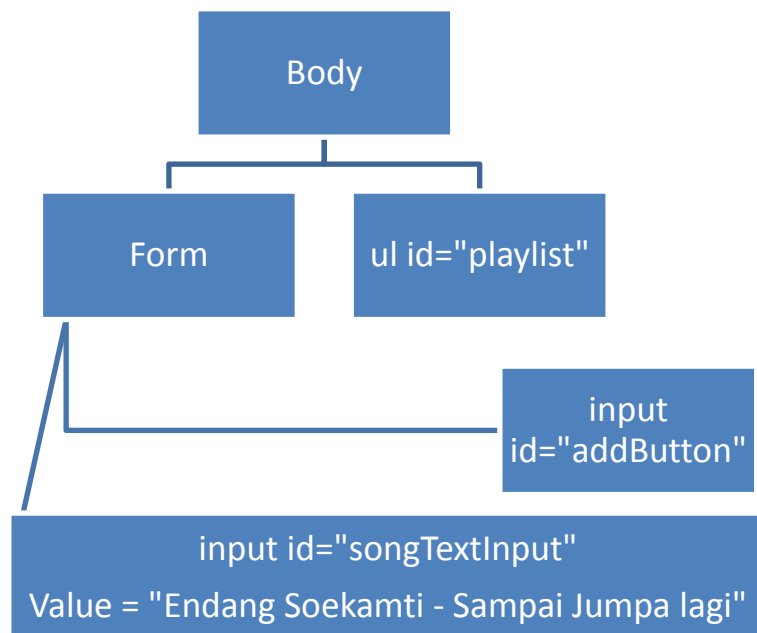
function handleButtonClick(){
  alert("Tombol telah diklik");
}

```

setelah tambahkan import SC tersebut ke file html. hasilnya ketika dijalankan adalah sebagai berikut:



6. Selanjutnya bagaimana menangkap text (judul lagu) yang diinputkan user dengan cara mengambil element input dari DOM dengan menggunakan method `getElementById` untuk mendapatkan nilai (*value property*) pada text elemen input berupa text yang diinputkan user pada kolom input, dengan cara sebagai berikut:

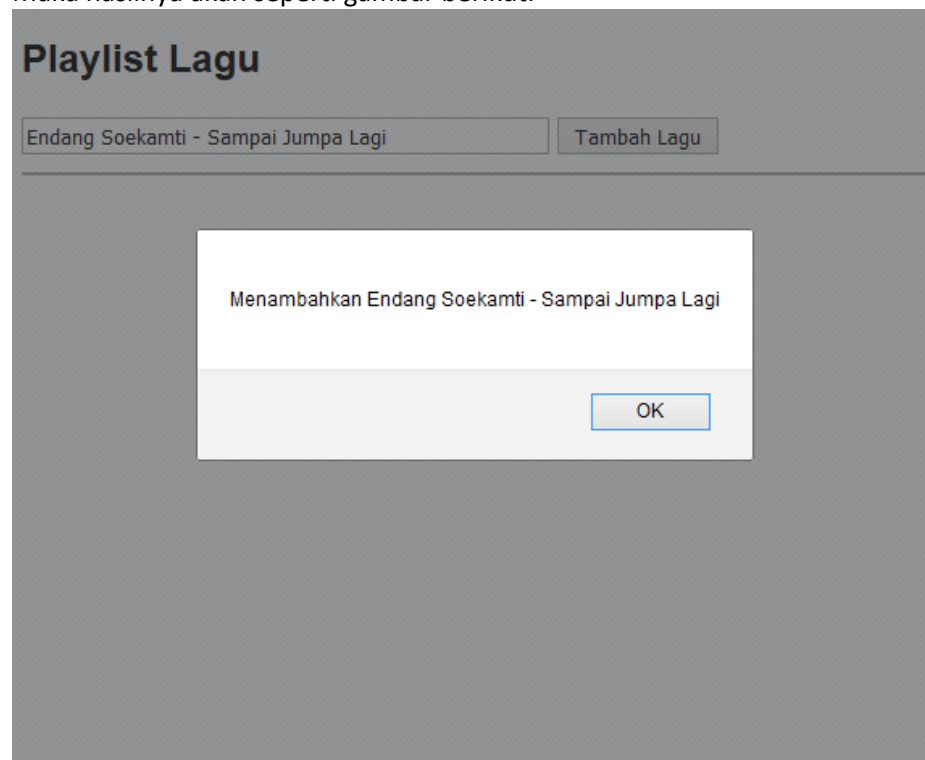


```
window.onload = init;

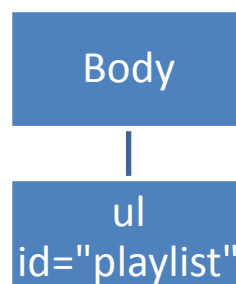
function init() {
    var button = document.getElementById("addButton");
    button.onclick=handleButtonClick;
}

function handleButtonClick() {
    var textInput = document.getElementById("songTextInput");
    var judulLagu = textInput.value;
    alert("Menambahkan " +judulLagu);
}
```

Maka hasilnya akan seperti gambar berikut.



7. Sekarang kita akan menampilkan playlist pada halaman web. Sebelumnya kita telah meletakkan list kosong pada dokumen HTML ().



Lagu yang diinputkan akan masuk kedalam list (playlist) dengan cara menambahkan elemen didalam elemen setiap kali lagu baru dimasukkan.



Tambahkan SC javascript berikut ini untuk menambahkan elemen `` dalam elemen ``.

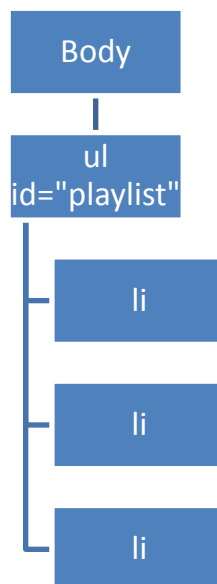
```

window.onload = init;

function init(){
    var button = document.getElementById("addButton");
    button.onclick=handleButtonClick;
}

function handleButtonClick(){
    var textInput = document.getElementById("songTextInput");
    var judulLagu = textInput.value;
    var li = document.createElement("li");
    li.innerHTML = judulLagu;
    var ul = document.getElementById("playlist");
    ul.appendChild(li);
    alert("Menambahkan " +judulLagu);
    return(0);
}
  
```

Kemudian susunan DOM yang baru adalah:



Kemudian Hasilnya adalah

Playlist Lagu

Endang Soekamti - Sampai Jumpa Lagi

Tambah Lagu

- Endang Soekamti - Sampai Jumpa Lagi

8. Tambahkan Script CSS berikut agar tampilan lebih enak dipandang. Beri nama 4 digit nim terakhir anda:

```
body{
    font-family:arial, sans-serif;
}
ul#playlist{
    border: 1px solid #a9a9a9;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius:5px;
    margin-top:10px;
    padding:0px;
    list-style-type:none;
}
ul#playlist li{
    border-bottom: 1px solid #a9a9a9;
    padding: 10px;
    background-image: -webkit-gradient(linear, left top, left bottom, from(#f9f9f9), to(#e3e3e3));
    background-image: -moz-linear-gradient(#f9f9f9, #e3e3e3);
    background-image: -ms-linear-gradient(#f9f9f9, #e3e3e3);
    background-image: -o-linear-gradient(#f9f9f9, #e3e3e3);
    background-image: -webkit-linear-gradient(#f9f9f9, #e3e3e3);
    background-image: linear-gradient(#f9f9f9, #e3e3e3);
}
ul#playlist li:last-child{
    -webkit-border-bottom-right-radius: 5px;
    -webkit-border-bottom-left-radius: 5px;
    -moz-border-radius-bottomright: 5px;
    -moz-border-radius-bottomleft: 5px;
    border-bottom:none;
    border-bottom-right-radius:5px;
    border-bottom-left-radius:5px;
}
ul#playlist li:first-child{
    -webkit-border-top-right-radius: 5px;
    -webkit-border-top-left-radius: 5px;
    -moz-border-radius-topright: 5px;
    -moz-border-radius-topleft: 5px;
    border-bottom-right-radius:5px;
    border-bottom-left-radius:5px;
}
```

Hasil akhirnya adalah sebagai berikut:

Playlist Lagu

Endang Soekamti - Sampai Jumpa Lagi

Labrynth - Jealous

E. Tugas

1. Print Screen semua point-point instruksi pekerjaan praktikum anda. Sertakan source code dan hasil running programnya.
2. Tuliskan dan jelaskan variabel-variabel dan methods yang digunakan pada source code javascript DOM HTML pada Praktikum.
3. Lampirkan laporan dengan nama file 4 digit nim terakhir anda dan source code yang sudah dibuat.

F. References

Freeman, Eric dan Robson, Elisabeth. 2011. *Head First HTML 5 Programming*. United States of America: O'Reilly Media.
W3school, http://www.w3schools.com/js/js_htmlDOM.asp