

JavaScript Document Object Model & Event Listener

PREPARED BY

MADE FOR

UNIVERSITAS PERTAMINA

PRAKTIKUM PEMROGRAMAN WEB

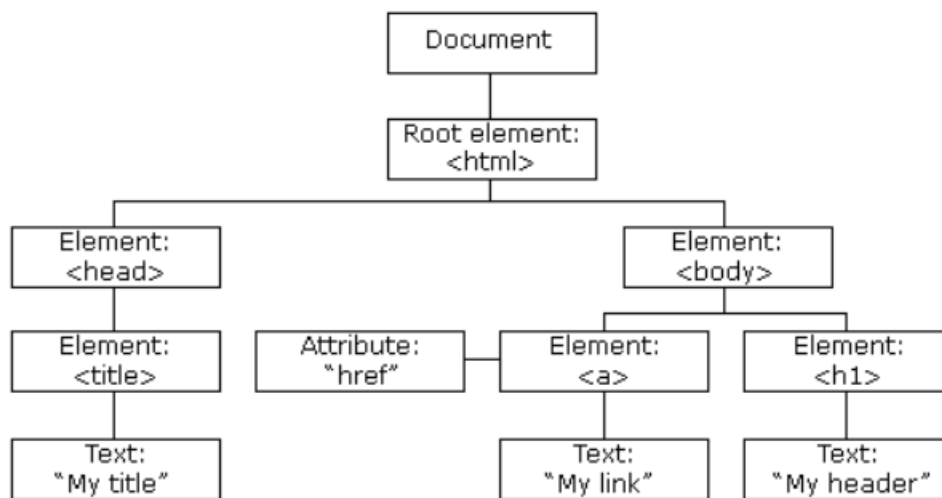
Pendahuluan

Pada beberapa kasus pengembangan website, banyak website yang tentunya dibangun dengan lebih interaktif kepada penggunanya. Interaktif disini adalah, terdapat banyak aksi yang terjadi ketika pengguna melakukan suatu kegiatan atau kejadian pada website yang dibangun. Contoh, ketika pengguna melakukan pengisian form, dan terdapat beberapa field kolom yang kosong, terdapat notifikasi bahwasanya kolom tersebut harus diisi. Atau contoh sederhananya, ketika pengguna mencoba untuk mengklik suatu tombol hapus data, maka akan muncul notifikasi untuk mengkonfirmasi penghapusan data dan notifikasi bahwasanya data berhasil dihapus.

Semua hal tersebut tentunya bukan menutup kemungkinan bahwasanya website dapat melakukan hal tersebut. Lantas, bagaimana hal tersebut dapat dilakukan? Hal ini dapat dilakukan dikarenakan adanya Javascript. Yang mana, javascript menyediakan Document Object Model (DOM) untuk dapat menyeleksi element HTML tertentu, untuk kemudian dapat diberikan Event Listener yang akan memberikan suatu aksi ketika pengguna melakukan suatu aksi pada elemen yang diseleksi. Lalu, apa itu DOM dan Event Listener?

Document Object Model (DOM)

Document Object Model, atau dikenal juga dengan DOM, merupakan interface dari pemrograman untuk dokumen HTML. Yang mana, peran DOM ini sebagai penerjemah antara Javascript dengan HTML (sama seperti CSS Selector). Fungsi dari DOM ini adalah untuk menjelaskan struktur dari HTML, memodifikasi tampilan, konten atau struktur dari website tanpa membuat website tersebut harus dimuat ulang. Secara umum, struktur DOM dari suatu website telah digambarkan seperti berikut:



Hal hal yang bisa dilakukan dengan DOM diantaranya adalah sebagai berikut:

1. Menyeleksi element

Untuk menyeleksi suatu element dengan DOM, kita dapat menggunakan beberapa cara. Secara umum, dapat dilakukan dengan menggunakan ID, Class, atau TagName.

Untuk melakukan hal ini, dapat dilakukan dengan cara sebagai berikut:

- `document.getElementById("nama-id")`: akan menyeleksi elemen yang memiliki id "nama-id". Nilai yang dikembalikan pada perintah ini adalah 1 element id yang pertama kali ditemukan oleh javascript.
- `document.getElementsByTagName("jenis-tag-html")`: akan menyeleksi semua element yang memiliki jenis tag htmlnya sesuai dengan yang ditulis pada parameter. Nilai yang akan dikembalikan pada perintah ini adalah array dari semua tag html yang ditulis.
- `document.getElementsByClassName("nama-class")`: akan menyeleksi semua element yang memiliki nama class "nama-class". Nilai yang dikembalikan pada perintah ini adalah semua element yang memiliki class "nama-class" dalam bentuk array.
- `document.querySelector("#id.class")`: akan menyeleksi element yang memiliki id "id" dan class "class". Query yang dikirimkan pada parameter ini sama seperti perlakuan pada CSS Selector. Nilai yang dikembalikan dalam bentuk 1 element yang pertama kali ditemukan oleh javascript.

- `document.querySelectorAll("#id.class")`: akan menyeleksi semua element yang memiliki id "id" dan class "class". Query yang dikirimkan pada parameter ini sama seperti perlakuan pada CSS Selector. Nilai yang dikembalikan dalam bentuk array yang berisi element element HTML yang memiliki kriteria sesuai dengan query yang telah dikirimkan.

2. Membuat element:

Dengan DOM, tentunya kita dapat membuat suatu element HTML. Pada kasus tertentu, membuat element ini dilakukan ketika developer menginginkan membuat suatu element ketika suatu kondisi yang diinginkan tercapai. Lantas, bagaimana membuat suatu element pada javascript?

Untuk membuat suatu element pada javascript dapat dilakukan dengan membuat element HTML terlebih dahulu, kemudian menambahkan nama id dan / atau nama class (optional). Kemudian, kita dapat menambahkan child didalam element yang telah dibuat atau menambahkan text didalamnya (optional). Hal hal tersebut dapat dilakukan seperti contoh berikut:

```
const div = document.createElement('div');
div.setAttribute('id', 'nama-id');
div.setAttribute('class', 'nama-class');
div.innerHTML = "Element Div Baru";
const main = document.createElement('main');
main.appendChild(div);
document.body.appendChild(main);
```

pada contoh diatas, tahap tahapan yang dilakukan untuk membuat element adalah dengan membuat elemen div, yang atributnya dengan id 'nama-id', class 'nama-class', yang berisi teks element div baru. Element div ini akan menjadi child atau berada didalam tag html 'main'. Dan main sendiri akan diposisikan pada tag body pada html. Untuk menambahkan teks pada suatu element, dapat dilakukan tidak hanya dengan innerHTML, tapi dapat dilakukan juga dengan menggunakan textContent.

3. Menghapus element

Dalam beberapa kasus, terdapat beberapa hal yang membuat developer membuat suatu kondisi yang dapat membuat suatu element dihilangkan dari websitenya. Hal ini dapat dilakukan dengan DOM. Bagaimana hal ini dapat dilakukan? Hal yang

dilakukan adalah kita perlu menyeleksi element yang ingin kita hapus. Hal tersebut dapat dilakukan seperti contoh berikut:

```
const div = document.querySelector('div#nama-id.nama-class');  
div.remove();
```

pada contoh diatas, kita menyeleksi element div dengan id 'nama-id' dan class 'nama-class'. Ketika element telah ditemukan, maka div akan dihapus dengan perintah method remove.

Pada beberapa kasus, kita juga dapat menghapus element child dari suatu element. Hal ini dapat dilakukan seperti contoh berikut:

```
const div = document.getElementById('nama-id');  
const childDiv = document.getElementById('id-child');  
div.removeChild(childDiv);
```

pada contoh diatas, kita menyeleksi element dengan id 'nama-id' dan 'id-child', Kemudian, child dari id 'nama-id', yaitu 'id-child' dihapus melalui perintah removeChild.

Event Listener

Event Listener merupakan suatu fungsi atau prosedur yang ada dalam program komputer yang akan menunggu terjadinya suatu kejadian untuk kemudian akan diberikan respon. Yang mana, event listener ini akan memungkinkan pengguna untuk dapat membuat respon khusus dari suatu peristiwa yang telah ditentukan.

Pembentukan event listener ini erat kaitannya dengan Document Object Model. Yang mana, Document Object Model ini akan melakukan seleksi element yang akan diberikan event listener, dan event listener yang akan menjalankan eksekusi pada element tersebut. Yang mana, seperti yang dilihat pada sub materi sebelumnya, yaitu DOM, kita dapat melihat bagaimana DOM akan menyeleksi element atau membuat element (yang dilakukan oleh developer pada suatu kondisi). Hal ini dapat diterapkan melalui EventListener.

Hal yang perlu dilakukan untuk memberikan event listener adalah dengan menyeleksi suatu element dengan menggunakan Javascript DOM, kemudian menjalankan method `addEventListener('event yang diinginkan', hal yang akan terjadi)`. Contoh nya seperti berikut:

```
const div = document.getElementById('button');
```

```
div.addEventListener('click', () => {  
  
    const p = document.createElement('p');  
  
    p.setAttribute('class', 'nama-class');  
  
    p.innerHTML = "Paragraf dari div";  
  
    div.appendChild(p);  
  
});
```

pada contoh diatas, Javascript akan mencari element div dengan id 'button'. Yang mana, div dengan id 'button' ini, ketika di klik, akan menambahkan element p dengan class 'nama-class' dan berisi paragraf bertuliskan 'Paragraf dari div'.

Event yang diberikan pada addEventListener ini tidak hanya berupa klik pada suatu item, tapi dapat berupa banyak hal. Adapun event yang dapat ditambahkan adalah sebagai berikut:

1. click: event yang dijalankan ketika suatu element di klik
2. mouseover: event yang dijalankan ketika suatu element berada dibawah kursor mouse
3. mouseout: event yang dijalankan ketika kursor mouse keluar dari jangkauan element.
4. resize: event yang dijalankan ketika ukuran layar berubah
5. mousemove: event yang dijalankan ketika pada suatu element, terdapat pergerakan baju.
6. dblclick: event yang dijalankan ketika suatu element di klik sebanyak dua kali dalam waktu yang sama.

Dan masih banyak lagi event yang dapat diberikan selain yang telah disebutkan diatas. Namun, pada beberapa kondisi, beberapa developer tentunya ingin menghapus event listener. Yang mana, ketika suatu kondisi yang diinginkan tercapai, maka event listener ini dapat dihapus atau tidak akan dijalankan. Hal ini dapat dilakukan dengan cara menambahkan atribut removeEventListener. Yang mana, perintah ini akan membuat suatu element tidak akan menjalankan suatu event walaupun terjadi suatu kejadian yang kita inginkan pada element tersebut. Hal itu dapat diterapkan seperti contoh berikut:

```
const div = document.getElementById('button');
```

```
const events = () => {  
  
    const p = document.createElement('p');  
  
    p.setAttribute('class', 'nama-class');  
  
    p.innerHTML = "Paragraf dari div";  
  
    div.appendChild(p);  
  
}  
  
div.addEventListener('click', events);  
  
div.addEventListener('dblclick', () => {  
  
    div.removeEventListener('click', events);  
  
})
```

LATIHAN

1. Bos perusahaan ingin anda membuat website dengan fitur pengisian form data pegawai. Yang mana, semua data diatas harus diisi. Jika form tidak diisi, maka akan muncul notifikasi error dibawah field form bersangkutan. Buatlah website tersebut untuk memenuhi permintaan bos. Terapkan Javascript DOM dan Event Listener untuk menyelesaikan permasalahan ini.

TAKE HOME TASK

1. Update Web Portofolio anda. Terapkan beberapa hal yang ada pada modul ini pada website yang anda bangun.