

# 5. Document Object Model (DOM)

Pemrograman  
JavaScript

# Nasrul, S.Pd.I, S.Kom, M.Kom



nasrul99@gmail.com

•Dosen Tetap STT Nurul Fikri

Instruktur IT NF Computer

Instruktur IT Sekolah Programmer  
YBM PLN

Instruktur IT Fast Com

Programmer

# Tujuan

Setelah mengikuti materi ini peserta diharapkan memahami dan mengerti tentang Document Object Model (DOM) di dalam bahasa pemrograman JavaScript serta dapat mengimplementasikannya di sebuah program aplikasi

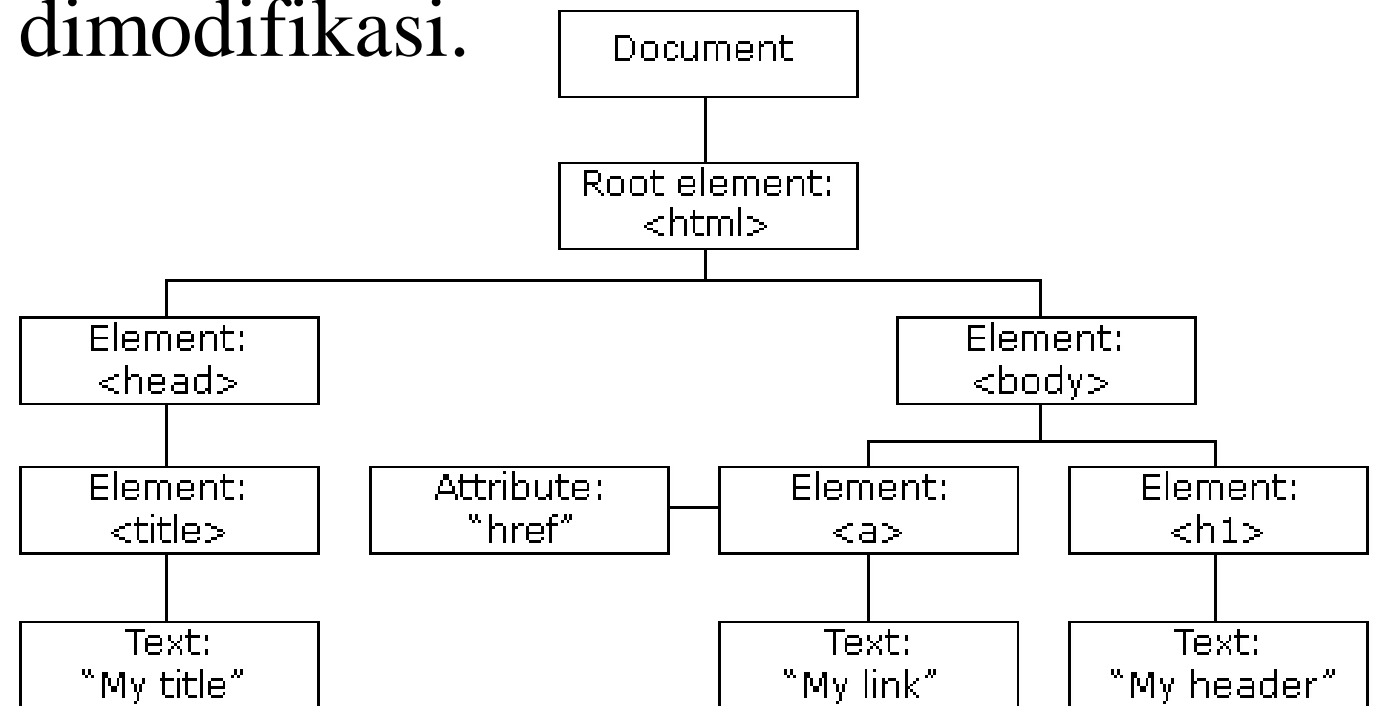
# Document Object Model (DOM)

DOM merupakan singkatan dari Document Object Model, yaitu model objek dari sebuah dokumen HTML yang dapat dimanfaatkan oleh JavaScript untuk mengolah dokumen HTML. Dengan memanfaatkan DOM, JavaScript dapat melakukan berbagai macam tindakan terhadap dokumen HTML.

# Document Object Model (DOM)

Saat halaman web dibuka, browser membuat HTML DOM dari halaman tersebut berupa objek. DOM merepresentasikan semua konten halaman yang dapat dimodifikasi.

Dengan DOM, Javascript mampu membuat HTML yang dinamis



# Hal yang bisa dilakukan oleh DOM

- DOM JavaScript dapat mengubah semua elemen HTML di halaman

- DOM JavaScript dapat mengubah semua atribut HTML di halaman

- DOM JavaScript dapat mengubah semua gaya CSS di halaman

- DOM JavaScript dapat menghapus elemen dan atribut HTML yang ada

- DOM JavaScript dapat menambahkan elemen dan atribut HTML baru

- DOM JavaScript dapat bereaksi terhadap semua peristiwa HTML yang ada di halaman

- DOM JavaScript dapat membuat acara HTML baru di halaman

# Yang Akan dipelajari pada DOM

- Menemukan Elemen HTML
- Mengubah Elemen HTML
- Menambah atau Menghapus Elemen HTML
- Navigasi Elemen HTML
- Mengubah CSS HTML



# Mencari Elemen HTML Dengan DOM

Property/Method	Penjelasan
<code>document.getElementById(id)</code>	Menemukan elemen HTML dengan id
<code>document.getElementsByClassName(name)</code>	Menemukan beberapa elemen HTML dengan nama class
<code>document.getElementsByTagName(name)</code>	Menemukan beberapa elemen HTML dengan nama tag
<code>document.querySelectorAll(selector)</code>	Menemukan beberapa elemen HTML dengan selector CSS



# Mencari Elemen HTML

Beberapa method untuk mencari elemen HTML :

- document.getElementById(id)
- document.getElementsByTagName(name)
- document.getElementsByClassName(name)

Note, pencarian selain dengan Id menghasilkan collection of object HTML (lihat dari s pada getElements)

Bisa dipilih dengan kurung siku ke indexnya

```
> document.getElementsByTagName('li')
< ▼ HTMLCollection [li] ⓘ
  ▶ 0: li
    length: 1
    __proto__: HTMLCollection
> document.getElementsByTagName('li')[0]
< ▼ <li>
  ::marker
  "Information"
  </li>
>
```

# Contoh Mencari Elemen HTML Dengan DOM

```
<!DOCTYPE html>
<html>
  <head>
    <title>Belajar DOM Javascript</title>
  </head>
  <body>
    <p id="isi"></p>
  </body>
  <script>
    document.getElementById("isi").innerHTML = "Hello DOM JS";
  </script>
</html>
```

# Mengubah Elemen HTML Dengan DOM

Property/Method	Penjelasan
<code>element.innerHTML = "isi konten"</code>	Mengubah konten suatu elemen dengan konten yang baru
<code>element.nama_atribut = "nilai atribut"</code>	Mengubah nilai atribut dari sebuah elemen.
<code>element.style.property_css = nilai property</code>	Mengubah style CSS suatu elemen
<code>element.setAttribute(nama_atribut, nilai)</code>	Mengubah nilai atribut

# Mengubah Elemen HTML

- `element.innerHTML = <new html content>`
- `element.<attribute> = <new value>`
- `element.style.<property> = <new style>`
- `element.setAttribute(<attribute>, <value>)`

Begin

- Information

End

Elements Console Sources Network

top Filter

> info = document.getElementsByTagName('li')[0]

< ▶ <li>...</li>

> info.style.backgroundColor = "blue";

< "blue"

> |

# InnerHTML

Properti innerHTML mengizinkan kita mengakses isi HTMLnya dalam bentuk string. Kita juga bisa memodifikasinya.

Ini adalah cara paling powerful untuk mengubah konten halaman.

Contoh :

```
document.body.innerHTML = 'The new BODY!';
```

Akan mengubah teks di dalam document.body ke teks baru tersebut.

# textContent

Properti innerHTML menginput teks ke halaman sebagai objek HTML (masih memungkinkan ada tag di dalamnya). Lebih aman jika menggunakan properti textContent karena teksnya diperlakukan sebagai teks murni (pure).

Misalkan ada menu input seperti ini

```
let name = prompt("What's your name?", "");  
elem.textContent = name;
```

Jika user memasukkan tag html, maka tag itu akan terakomodir di halaman kita (di objek elem). Tetapi jika menggunakan elem.textContent = name; tagnya akan dianggap sebagai bagian dari string.

# OuterHTML

Properti outerHTML tidak mengubah elemennya, tetapi mengganti elemen tersebut di DOM (elemen lamanya tetap ada)

```
<div>Hello, world!</div>
<script>
  let div = document.querySelector('div');
  // ganti div.outerHTML dengan <p>...</p>
  div.outerHTML = '<p>A new element</p>'; // (*)
  // ternyata div yang lama tetap bisa diakses
  alert(div.outerHTML); // <div>Hello, world!</div> (**)
</script>
```



# Menambah atau Menghapus Elemen HTML Dengan DOM

Property/Method	Penjelasan
<code>document.createElement(element)</code>	Membuat elemen HTML baru
<code>document.removeChild(element)</code>	Menghapus elemen HTML
<code>document.appendChild(element)</code>	Menambahkan elemen HTML
<code>document.replaceChild(kata baru, kata lama)</code>	Mengubah sebuah kata dalam satu elemen
<code>document.write(text)</code>	Menuliskan teks pada dokumen HTML

# Menambah dan Menghapus Elemen HTML

- `document.createElement(element)`
- `document.removeChild(element)`
- `document.appendChild(element)`
- `document.replaceChild(new, old)`
- `document.write(text)`

Begin

End

1. Information

```
Elements Console Sources Network Performance Memory
top Filter Default levels ▼
> unorderedlist = document.getElementsByTagName("ul")[0]
< ▶ <ul>...</ul>
> parent = unorderedlist.parentElement
< ▶ <body>...</body>
> child = unorderedlist.children[0]
< ▼ <li>
  ::marker
  "Information"
  </li>
> orderedlist = document.createElement("ol")
< <ol></ol>
> parent.append(orderedlist)
< undefined
> orderedlist.append(child)
< undefined
```

# Contoh Menambah Elemen HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Belajar DOM Javascript</title>
  </head>
  <body>
    <script>
      let div = document.createElement('div');
      div.id = 'content';
      div.innerHTML = '<p>Contoh Menambah Element</p>';
      document.body.appendChild(div);
    </script>
  </body>
</html>
```

# Navigasi HTML DOM

Sebenarnya sebuah dokumen HTML itu tersusun secara hirarki. Elemen yang berada di level paling atas disebut **root**. Setiap elemen pasti mempunyai induk elemen, kecuali root. Beberapa elemen kadang kala memiliki induk yang sama. hubungan antara elemen yang memiliki induk sama disebut **sibling**.

- <html> adalah root



- <html> tidak memiliki orang tua



- <html> adalah induk dari <head> dan <body>



- <head> adalah anak pertama dari <html>



- <body> adalah anak terakhir dari <html>

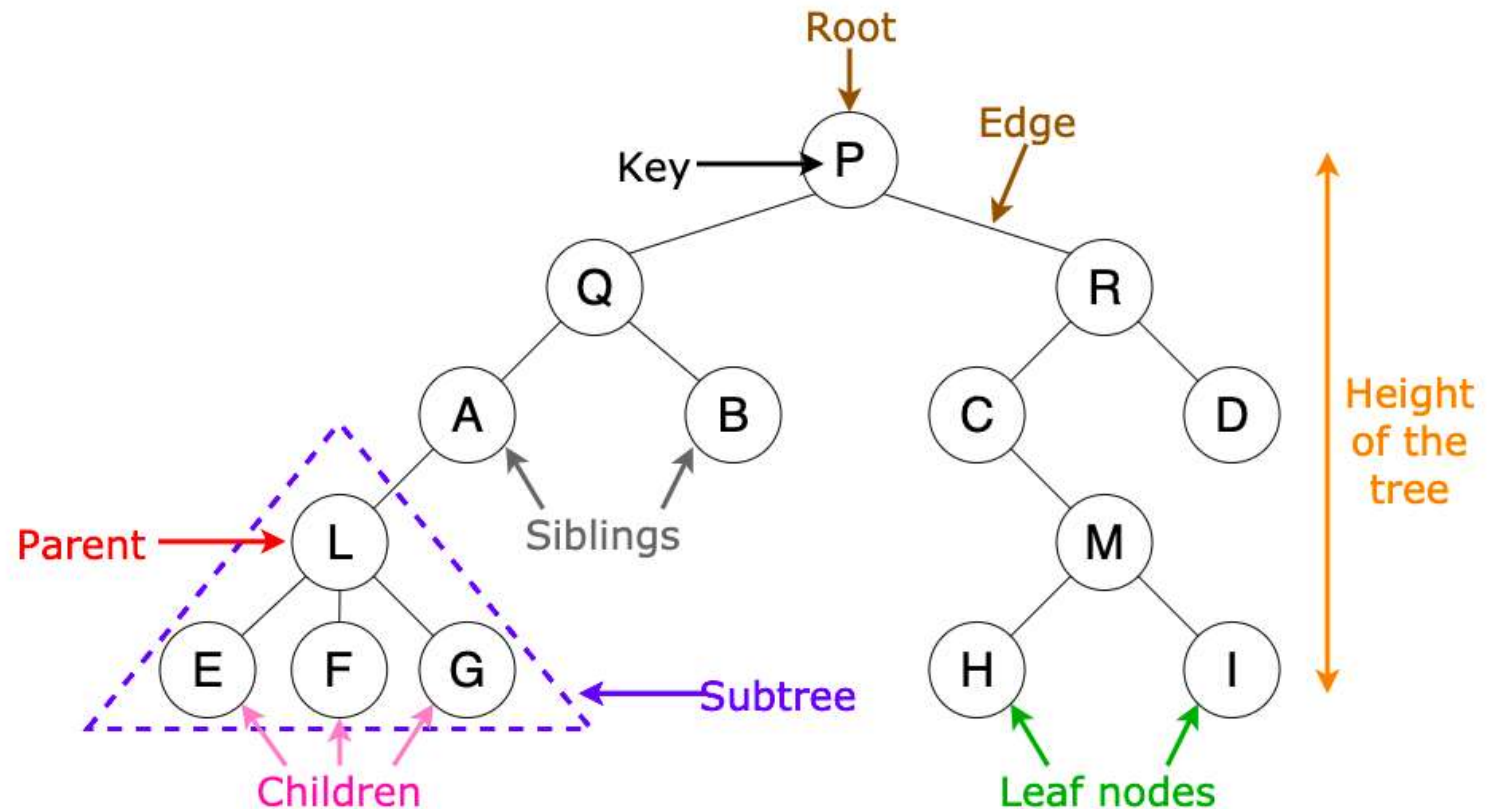


# Navigasi HTML DOM

Property/Method	Penjelasan
parentNode	Mendapatkan elemen induk
childNodes[urutan]	Mendapatkan elemen yang menginduk. Urutan diisi nomor urut elemen dimulai dari nol.
firstChild	Mendapatkan elemen urutan pertama yang menginduk
lastChild	Mendapatkan elemen urutan terakhir yang menginduk
nextSibling	Mendapatkan elemen setelahnya
previousSibling	Mendapatkan elemen sebelumnya

# Navigasi HTML DOM

- Children
  - childNodes
  - firstChild
  - lastChild
- Ancestors
- Descendants
- Sibling
  - nextSibling
  - previousSibling
- Parent



# Navigasi HTML DOM

## Kode HTML

```
<html>
<body>
  <div>Begin</div>
  <ul>
    <li>Information</li>
  </ul>
  <div>End</div>
</body>
</html>
```

## Tampilan di browser

Begin

- Information

End

## Mengambil body sebagai object JS

```
> body = document.body
< ▼ <body>
    <div>Begin</div>
    ▼ <ul>
        ► <li>...</li>
    </ul>
    <div>End</div>
  </body>

> typeof body
< "object"

> |
```



# Ubah CSS dengan DOM

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Belajar DOM Javascript</title>
  </head>

  <body>
    <p id="isi">Ubah CSS Warna menggunakan DOM</p>
    <script>
      document.getElementById("isi").
        style.color = "blue";
    </script>
  </body>
</html>
```

# Referensi

- <https://www.w3schools.com/>
- <https://www.javapoint.com/>
- <https://www.tutorialspoint.com/>