

JavaScript

Pengenalan Javascript

JavaScript adalah bahasa pemrograman yang terutama digunakan untuk pengembangan web. JavaScript memungkinkan pengembang membuat halaman web yang interaktif dan dinamis. Bersama dengan HTML dan CSS, JavaScript adalah salah satu teknologi inti dalam pengembangan web.



Cara Menjalankan Javascript

Umumnya ada 4 cara menggunakan javascript

- Pada console di browser
- Internal javascript di html
- External javascript .js
- Dengan node js

Output

Output adalah sebuah tampilan program yang biasanya digunakan untuk memperlihatkan hasil akhir.

Pada javascript output dapat ditampilkan dengan 4 cara :

- Menggunakan Fungsi `console.log()`;
- Menggunakan Fungsi `alert()`;
- Menggunakan Fungsi `document.write()`;
- Menggunakan `innerHTML`.

Variabel

Variables are Containers for Storing Data

JavaScript Variables can be declared in 4 ways:

- Automatically
- Using **var**
- Using **let**
- Using **const**

Tipe Data

tipe data adalah jenis-jenis nilai yang bisa disimpan di dalam variabel di JavaScript. Dengan tipe data, kita tahu cara memperlakukan suatu nilai (apakah angka, teks, array, dll).

Tipe Data	Contoh
String	"Halo"
Number	123, 45.6
Boolean	true, false
Array	["a", "b", "c"]
Object	{ nama: "Budi" }
Null	null
Undefined	undefined
Symbol	Symbol("id")
BigInt	123n

Operator

Operator digunakan untuk operasi matematik. Terdapat 5 jenis operator : operator aritmatika, assignment, comparison, logical, dan bitwise.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (<u>ES2016</u>)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

Operator

Operator Assignment adalah operator yang digunakan untuk memberikan tugas kepada variabel. Biasanya digunakan untuk mengisi variabel.

Operator	Example	Same As
=	<code>x = y</code>	<code>x = y</code>
<code>+=</code>	<code>x += y</code>	<code>x = x + y</code>
<code>-=</code>	<code>x -= y</code>	<code>x = x - y</code>
<code>*=</code>	<code>x *= y</code>	<code>x = x * y</code>
<code>/=</code>	<code>x /= y</code>	<code>x = x / y</code>
<code>%=</code>	<code>x %= y</code>	<code>x = x % y</code>
<code>**=</code>	<code>x **= y</code>	<code>x = x ** y</code>

Operator

Operator comparison adalah operator yang digunakan untuk membandingkan dua nilai.

Operator perbandingan akan menghasilkan sebuah nilai boolean true dan false.

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

Operator

Logical Operator digunakan untuk melakukan operasi terhadap dua nilai boolean.

Operator	Description
&&	logical and
	logical or
!	logical not

Array

Array merupakan **struktur data** yang digunakan untuk menyimpan sekumpulan data dalam satu tempat.

Struktur data merupakan cara-cara atau metode yang digunakan untuk menyimpan data di dalam memori komputer.

Setiap data pada array memiliki indeks.

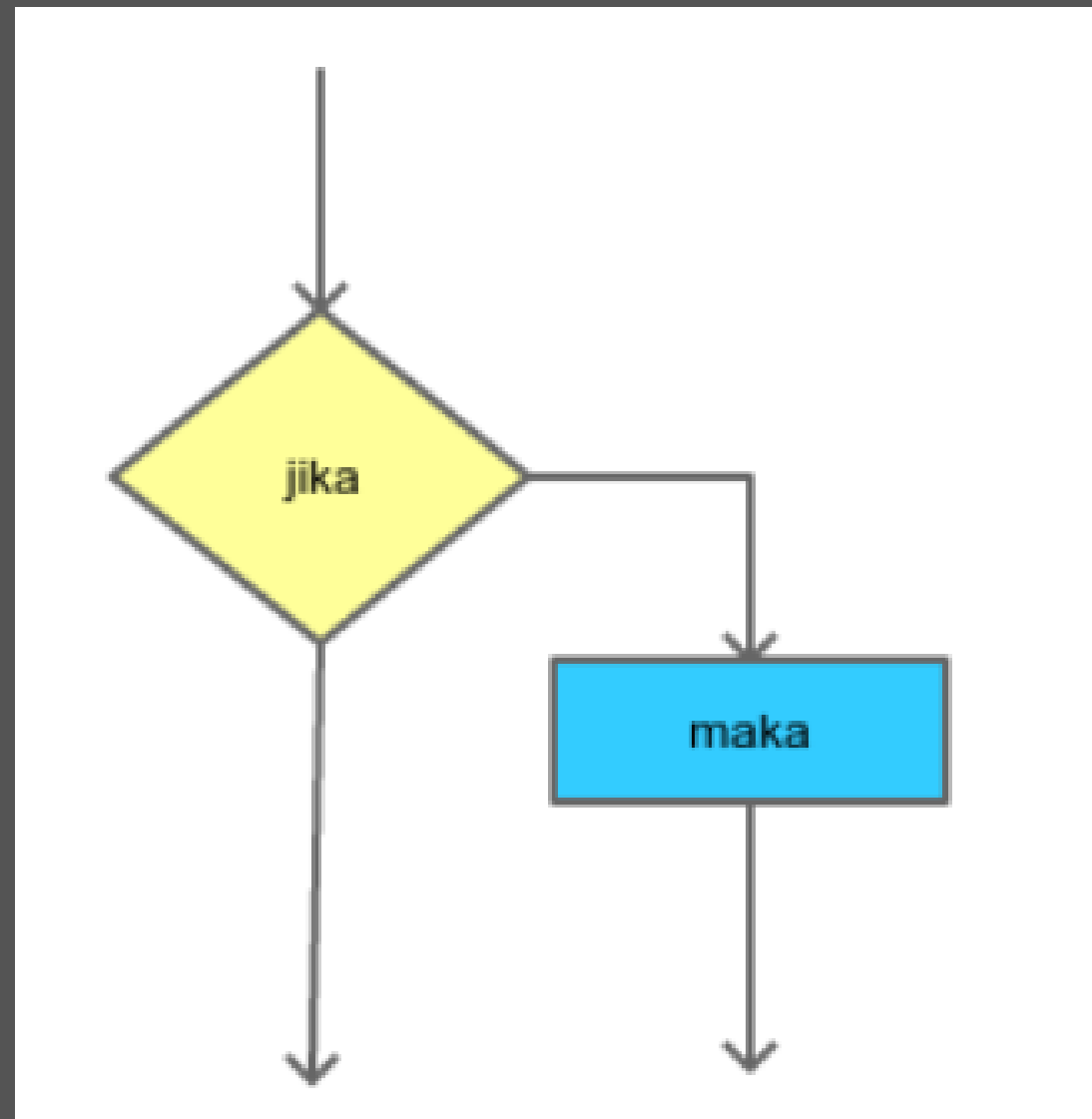
"Hardisk 2TB"	"Flashdisk 32GB"	"Modem"
0	1	2

Property Array

Properti	Keterangan	Contoh
<code>.length</code>	Menghitung jumlah elemen	<code>buah.length</code> hasilnya <code>3</code>
<code>.push()</code>	Menambahkan data di akhir array	<code>buah.push('nanas')</code>
<code>.pop()</code>	Menghapus data terakhir	<code>buah.pop()</code>
<code>.unshift()</code>	Menambahkan data di awal array	<code>buah.unshift('anggur')</code>
<code>.shift()</code>	Menghapus data pertama	<code>buah.shift()</code>
<code>.join()</code>	Menggabungkan isi array jadi string	<code>buah.join(',')</code>
<code>.slice()</code>	Mengambil sebagian isi array	<code>buah.slice(1, 3)</code>
<code>.splice()</code>	Menambah/Menghapus/Mengganti isi array	<code>buah.splice(1, 1, 'kiwi')</code>
<code>.forEach()</code>	Looping untuk setiap elemen array	<code>buah.forEach(item => console.log(item));</code>
<code>.map()</code>	Membuat array baru dari array lama	<code>const baru = angka.map(x => x*2)</code>

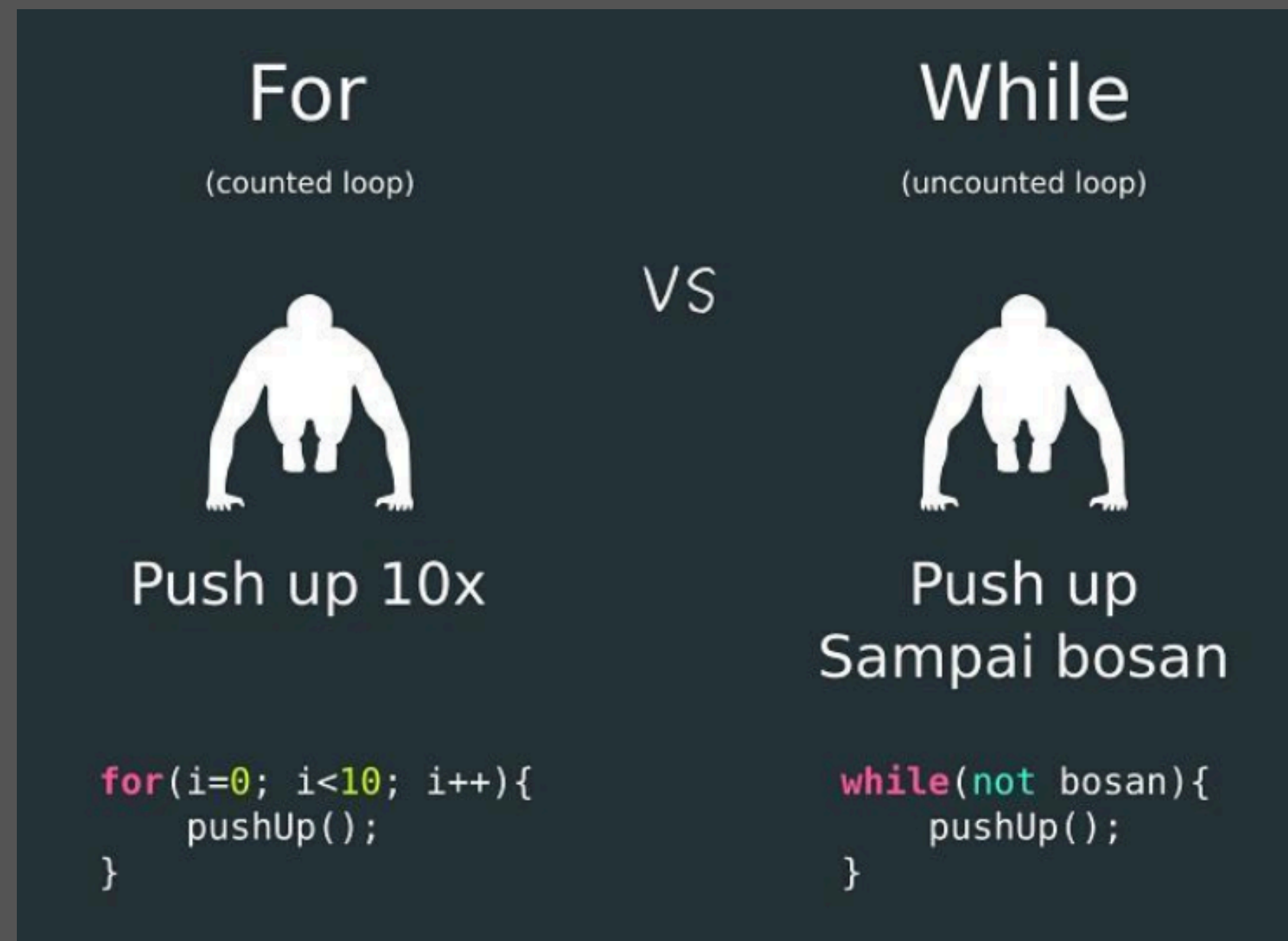
Percabangan

Percabangan adalah alur program yang bercabang, memilih jalan mana yang diambil berdasarkan hasil kondisi. Menggunakan operator perbandingan dalam pengkondisiannya. Terdapat 2 syntax untuk percabangan : if else dan switch



Perulangan

Perulangan adalah proses menjalankan kode berulang kali selama kondisi tertentu masih terpenuhi. Ada 2 jenis perulangan yaitu for loop dan while loop.



Fungsi

Fungsi adalah sub-program yang bisa digunakan kembali baik di dalam program itu sendiri, maupun di program yang lain.

Fungsi merupakan struktur dasar dari javascript itu sendiri.

Tujuan Fungsi:

1. Mengorganisir kode supaya lebih rapi dan modular.
2. Menghindari pengulangan kode (kode DRY — Don't Repeat Yourself).
3. Mempermudah pemeliharaan kode karena tiap fungsi punya tugas spesifik.
4. Mempercepat pengembangan karena fungsi bisa dipakai ulang.

Object

Object di JavaScript adalah struktur data yang digunakan untuk menyimpan kumpulan data dan/atau fungsi dalam satu entitas. Data dalam objek disimpan dalam bentuk key-value pair (pasangan kunci-nilai).

Object dipakai hampir di semua hal di JavaScript (misalnya JSON, DOM, dll).

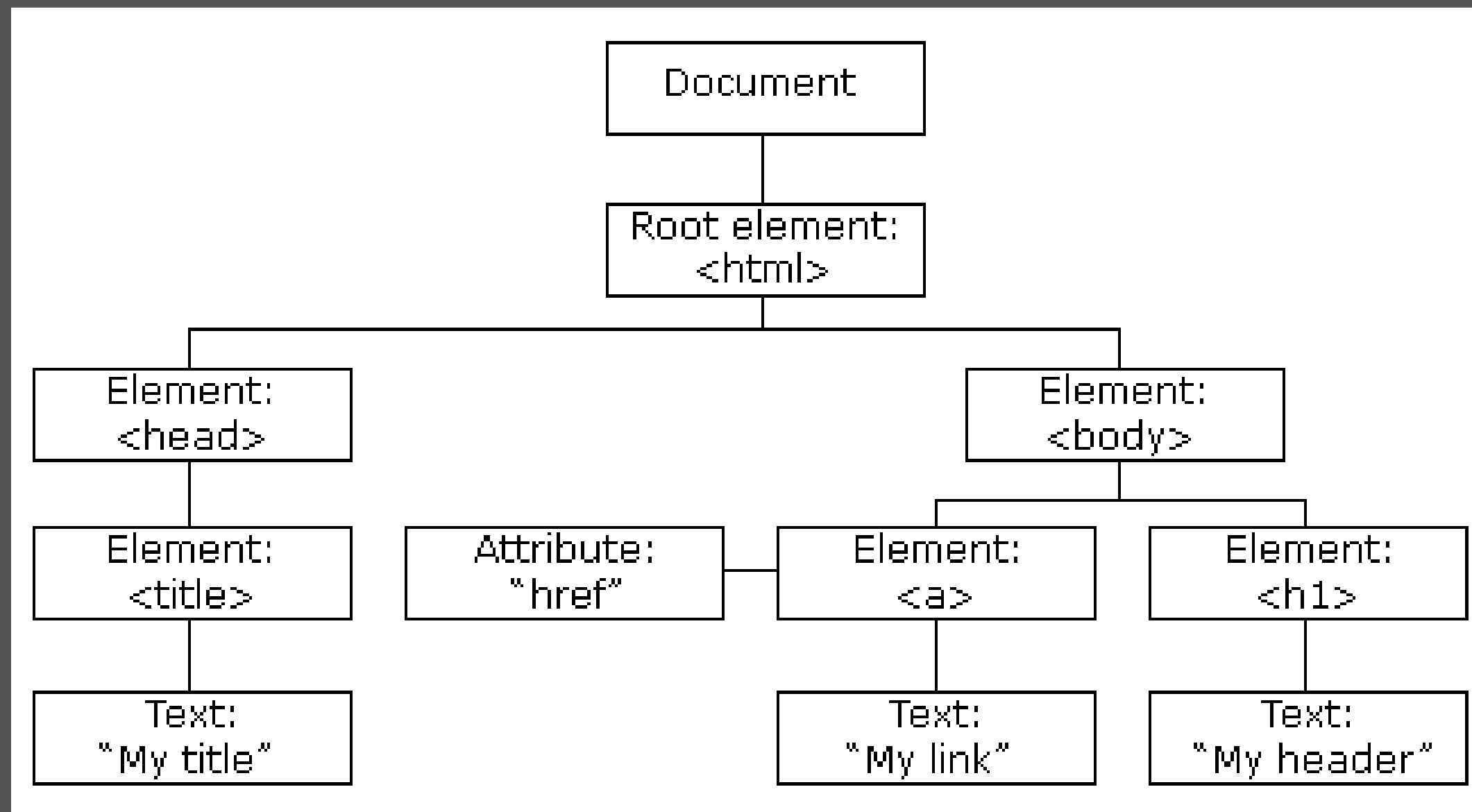
DOM

DOM (Document Object Model) adalah sebuah standar atau antarmuka pemrograman yang digunakan untuk merepresentasikan dokumen berbasis teks seperti HTML atau XML menjadi struktur pohon (tree structure) yang bisa dimanipulasi oleh bahasa pemrograman seperti JavaScript.

Dengan DOM, developer dapat membaca, mengubah, menambahkan, atau menghapus elemen dan konten dalam dokumen web secara dinamis.

DOM

Objek document adalah model dari dokumen HTML. Objek ini berisi kumpulan fungsi dan atribut berupa objek dari elemen HTML yang bisa digambarkan dalam bentuk pohon seperti ini



DOM

In the DOM, all HTML elements are defined as objects

A property is a value that you can get or set (like changing the content of an HTML element).

A method is an action you can do (like add or deleting an HTML element).

Fungsi	Keterangan
<code>getElementById()</code>	Mengambil elemen berdasarkan id
<code>getElementsByClassName()</code>	Mengambil elemen berdasarkan class
<code>querySelector()</code>	Mengambil elemen dengan selector CSS
<code>createElement()</code>	Membuat elemen baru
<code>appendChild()</code>	Menambahkan elemen sebagai anak
<code>removeChild()</code>	Menghapus elemen anak
<code>setAttribute()</code>	Menambahkan/mengubah atribut
<code>addEventListener()</code>	Menambahkan event handler

DOM Selection

DOM Selection merupakan method dari object elemen. Kita bisa memilih elemen html, menambahkan atau mengurangi elemen apa saja yang kita inginkan pada javascript. Berikut method DOM selection yang umum dipakai :

Metode	Keterangan	Contoh
<code>getElementById()</code>	Mengambil elemen berdasarkan ID	<code>document.getElementById('judul')</code>
<code>getElementsByClassName()</code>	Berdasarkan class	<code>document.getElementsByClassName('menu')</code>
<code>getElementsByTagName()</code>	Berdasarkan tag	<code>document.getElementsByTagName('p')</code>
<code>querySelector()</code>	CSS selector pertama yang cocok	<code>document.querySelector('.menu')</code>
<code>querySelectorAll()</code>	Semua elemen yang cocok selector	<code>document.querySelectorAll('li')</code>

DOM Manipulation

DOM Manipulation adalah proses membaca, mengubah, menambahkan, atau menghapus elemen dan konten dalam dokumen HTML secara dinamis melalui JavaScript.

Dengan manipulasi DOM, kita bisa membuat halaman web yang lebih interaktif tanpa harus me-reload halaman. Ini salah satu dasar utama dalam membuat website modern. Ada beberapa syntax untuk memanipulasi elemen:

- `inner.HTML` = untuk mengubah isi dari elemen yang kita pilih
- `<element>.style.<property>` = mengubah style dari elemen
- `<element>.get/set/removeAttribute` =
mendapatkan/menambahkan/menghilangkan attribute dari elemen
- `element.classList. ...` = mengelola kelas

element.classList

element.classList.add()

element.classList.remove()

element.classList.toggle()

element.classList.item()

element.classList.contains()

element.classList.replace()

DOM Manipulation

Method	Description
<code>document.createElement(<i>element</i>)</code>	Create an HTML element
<code>document.removeChild(<i>element</i>)</code>	Remove an HTML element
<code>document.appendChild(<i>element</i>)</code>	Add an HTML element
<code>document.replaceChild(<i>new</i>, <i>old</i>)</code>	Replace an HTML element
<code>document.write(<i>text</i>)</code>	Write into the HTML output stream

DOM Events

DOM Events adalah cara untuk menangkap dan merespon interaksi pengguna atau aktivitas lain yang terjadi di halaman web. Cara mengakses event yang terjadi adalah dengan fungsi `.addEventListener()`.

```
element.addEventListener(event, function, useCapture);
```

Selanjutnya kita membuat fungsi callback yang dijalankan saat event terjadi. Macam-macam callback yang umum digunakan :

Kategori	Nama Event	Kapan Terjadi	Contoh Penggunaan
◆ Mouse Events	click	Saat elemen diklik	<code>element.addEventListener('click', fn)</code>
	dblclick	Klik dua kali	<code>element.addEventListener('dblclick', fn)</code>
	mouseover	Kursor masuk ke elemen	<code>element.addEventListener('mouseover', fn)</code>
	mouseout	Kursor keluar dari elemen	<code>element.addEventListener('mouseout', fn)</code>
◆ Keyboard Events	keydown	Tombol ditekan (awal tekan)	<code>element.addEventListener('keydown', fn)</code>
	keyup	Tombol dilepas	<code>element.addEventListener('keyup', fn)</code>
◆ Form Events	submit	Form disubmit	<code>form.addEventListener('submit', fn)</code>
	change	Nilai input berubah	<code>input.addEventListener('change', fn)</code>
	input	Saat pengguna mengetik	<code>input.addEventListener('input', fn)</code>
◆ Window Events	load	Halaman selesai dimuat	<code>window.addEventListener('load', fn)</code>
	resize	Ukuran jendela berubah	<code>window.addEventListener('resize', fn)</code>
	scroll	Saat halaman discroil	<code>window.addEventListener('scroll', fn)</code>

Makasih