

DISKUSIKAN MATERI

LAPORKAN MATERI

Daftar Modul

Masukkan kata kunci

Persetujuan Hak Cipta

Modul 1: Introduction
Course (Gratis)

Prasyarat Kelas (Gratis)

Apa yang Akan Kita
Pelajari? (Gratis)

Tools Requirement

Modul 2: ECMAScript 6
(Gratis) Running Snippet Code
(Gratis) Persiapan Project
Latihan (Gratis)

Life Cycle of Custom Element

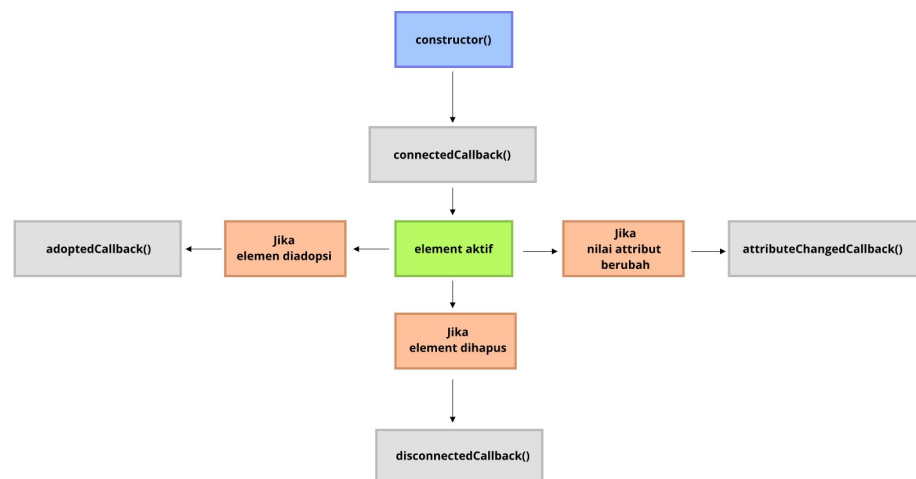
Potongan kode untuk materi ini:

<https://repl.it/@dicodingacademy/163-03-element-life-cycle?lite=true>

Ketika sebuah JavaScript class mewarisi sifat dari `HTMLElement` maka class tersebut akan memiliki siklus hidup layaknya sebuah elemen HTML. Kita dapat menerapkan logika pada setiap siklus hidup yang ada dengan memanfaatkan lifecycle callbacks yang ada. Berikut ini lifecycle callbacks yang ada pada `HTMLElement`:

- `connectedCallback()` : Akan terpanggil setiap kali elemen berhasil ditambahkan ke dokumen HTML (DOM). Callback ini merupakan tempat yang tepat untuk menjalankan konfigurasi awal seperti mendapatkan data, atau mengatur attribute.
- `disconnectedCallback()` : Akan terpanggil setiap kali elemen dikeluarkan (`remove()`) dari DOM. Callback ini merupakan tempat yang tepat untuk membersihkan data yang masih disimpan pada elemen. Bisa itu event, state, ataupun objek.
- `attributeChangedCallback()` : Akan terpanggil setiap kali nilai atribut yang di-observe melalui fungsi static `get observedAttributes()` diubah. Callback ini bisa kita manfaatkan untuk memuat ulang data yang ditampilkan oleh elemen.
- `adoptedCallback()` : Akan terpanggil setiap kali elemen dipindahkan pada dokumen baru. Kita relatif jarang menggunakan callback ini, namun jika kita memanfaatkan tag `<iframe>` maka callback ini akan terpanggil.

Untuk mempermudah memahami urutan siklus hidup element pada HTML kita bisa lihat pada ilustrasi berikut:



Walaupun sebenarnya `constructor()` bukan termasuk siklus hidup HTML Element, namun fungsi tersebut sering digunakan untuk melakukan konfigurasi awal ketika pertama kali element dibuat. Seperti menentukan event listener, atau menetapkan Shadow DOM.

Ketika kita mengimplementasikan constructor pada custom element, kita wajib memanggil method `super()`. Jika tidak, maka akan menghasilkan error:

```
1. ReferenceError: Must call super constructor in derived class before accessing 'this' or returning from derived constructor
```

Class yang merupakan custom element lebih ketat dibandingkan class lain. Kita tidak dapat membuat argument pada constructor class-nya. Karena instance dibuat tidak menggunakan keyword `new` seperti class JavaScript umumnya.

Terdapat dua cara membuat instance dari custom element. Yang pertama adalah menggunakan

nama tagnya langsung yang dituliskan pada kode HTML. Contohnya:

```
1. <body>
2.   <image-figure></image-figure>
3. </body>
```

Lalu cara kedua adalah dengan menggunakan sintaks JavaScript. Sama seperti membuat element HTML biasa, kita gunakan method `document.createElement` dalam membuat elemen baru.

```
1. const imageFigureElement = document.createElement("image-figure");
2. document.body.appendChild(imageFigureElement);
```

Kita bisa mencobanya sendiri dengan menuliskan kode-kode berikut dan menjalankannya pada browser. Kemudian lihat output yang dihasilkan pada browser. Output tersebut akan menunjukkan urutan siklus hidup yang terpanggil.



[index.html](#) [image-figure.js](#) [main.js](#) [output console](#) [html structure](#)

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <meta name="viewport" content="width=device-width">
6.     <title>Element life cycle</title>
7.   </head>
8.   <body>
9.     <!-- silakan hapus tag <image-figure> untuk membuatnya secara sintaks JavaScript -->
10.    <image-figure></image-figure>
11.    <script src="image-figure.js"></script>
12.    <script src="main.js"></script>
13.  </body>
14. </html>
```

Implementasi *lifecycle callback* pada custom element bersifat opsional. Kita tidak perlu menuliskannya jika memang tidak diperlukan.

[← KEMBALI KE MATERI SEBELUMNYA](#)

[LANJUTKAN KE MATERI BERIKUTNYA →](#)

dicoding



PERUSAHAAN

[Tentang Kami](#)

[Blog](#)

[Berita Terbaru](#)



PROGRAM

[Academy](#)

[Challenge](#)

[Event](#)

[Job](#)

[Rewards](#)

SUPPORT

[Bantuan](#)

[FAQ](#)

[Hubungi Kami](#)

