



DISKUSIKAN MATERI

LAPORKAN MATERI

Daftar Modul

Persetujuan Hak Cipta ✓

Modul 1: Introduction Course (Gratis) ✓

Prasyarat Kelas (Gratis) ✓

Apa yang Akan Kita Pelajari? (Gratis) ✓

Tools Requirement ✓

Modul 2: ECMA Script 6 (Gratis) ✓

Running Snippet Code (Gratis) ✓

Persiapan Project Latihan (Gratis) ✓

Write your first Custom Element

Potongan kode untuk materi ini: <https://repl.it/@dicodingacademy/163-03-first-custom-element?lite=true>

Dalam membuat custom element, kita menuliskannya dengan menggunakan JavaScript class. Class tersebut mewarisi sifat dari `HTMLElement`. `HTMLElement` merupakan interface yang merepresentasikan element HTML. `Interface` ini biasanya diterapkan pada class JavaScript sehingga terbentuklah element HTML baru melalui class tersebut (custom element).

Berikut contoh penulisan dalam membuat custom element:

```
1. class ImageFigure extends HTMLElement {
2.
3. }
```

Yey! `ImageFigure` sekarang merupakan sebuah HTML element baru. Namun tunggu dulu. Untuk menggunakannya pada berkas HTML, kita perlu menetapkan nama tag yang nantinya digunakan pada HTML. Caranya dengan menggunakan variabel `customElements` seperti ini:

```
1. customElements.define("image-figure", ImageFigure);
```

`customElements` merupakan *global variable* yang digunakan untuk mendefinisikan *custom element* dan memberitahu bahwa terdapat HTML tag baru. Di dalam `customElements` terdapat method yang bernama `define()`. Di sinilah kita meletakkan tag name baru kemudian diikuti dengan JavaScript class yang menerapkan sifat `HTMLElement`.

“Dalam penamaan tag untuk custom element, nama tag harus terdiri dari dua kata yang dipisahkan oleh dash (-). Jika tidak, pembuatan custom element tidak akan berhasil. Hal ini diperlukan untuk memberi tahu browser perbedaan antara elemen asli HTML dan custom element.”

Setelah mendefinisikan custom element, barulah ia siap digunakan pada berkas HTML. Kita cukup menuliskan tagnya layaknya elemen HTML biasa.

```
1. <image-figure></image-figure>
```

Jangan lupa lampirkan script pada berkas yang digunakan untuk menuliskan class `ImageFigure`.

```
1. <script src="image-figure.js"></script>
```

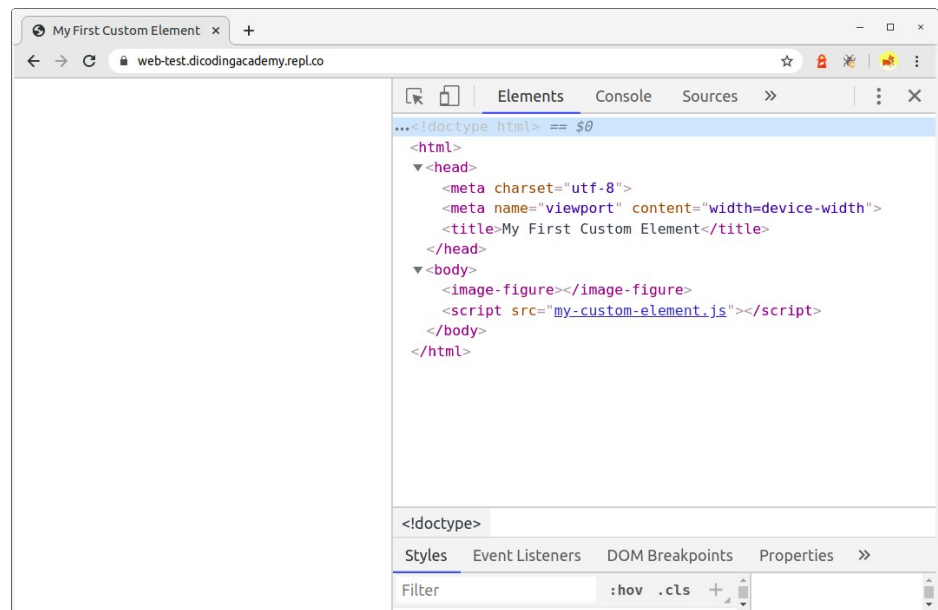
Berikut kode lengkapnya:

index.html image-figure.js

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <meta name="viewport" content="width=device-width">
6.     <title>My First Custom Element</title>
7.   </head>
8.   <body>
9.     <image-figure></image-figure>
10.    <script src="image-figure.js"></script>
```

```
11. </body>
12. </html>
```

Coba jalankan kode di atas pada browser, kita tidak akan mendapatkan apapun. Sampai saat ini, element `<image-figure>` berperan layaknya element `<div>` ataupun `` yang tidak memiliki fungsi khusus. Karena kita belum menetapkan seperti apa jadinya element baru ini.



Untuk menetapkan seperti apa fungsi dari elemen baru, kita lakukan semuanya dengan menggunakan kode JavaScript yang dituliskan di dalam class `ImageFigure`. Tapi sebelum itu, kita pelajari dulu siklus hidup (*life cycle*) dari elemen HTML.

[← KEMBALI KE MATERI SEBELUMNYA](#)

[LANJUTKAN KE MATERI BERIKUTNYA →](#)



PERUSAHAAN

Tentang Kami

Blog

Berita Terbaru



PROGRAM

Academy

Challenge

Event

Job

Rewards

SUPPORT

Bantuan

FAQ

Hubungi Kami

