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)

Observe Attribute Value

Potongan kode untuk materi ini:
https://repl.it/@dicodingacademy/163-03-attribute-observe?lite=true

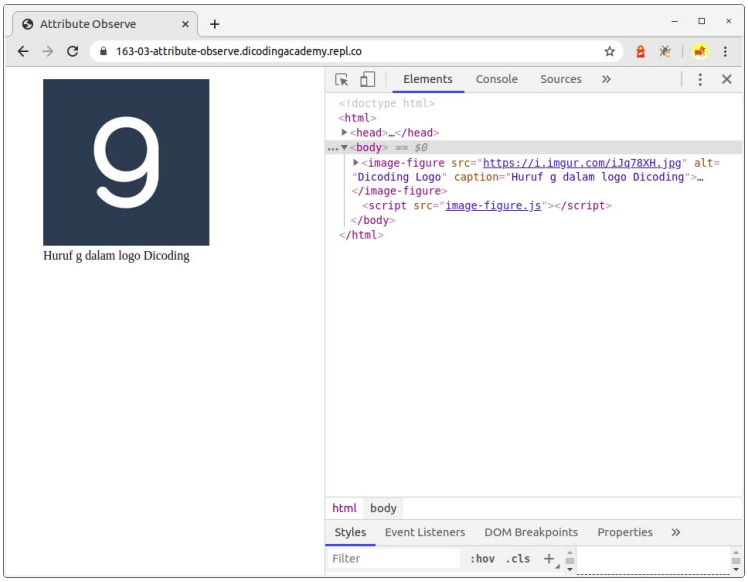
Ketika elemen sudah tampak pada DOM, tidak jarang kita mengganti nilai dari atributnya karena terdapat data yang perlu diperbaharui. Jika kita menggunakan elemen HTML standar, perubahan yang diterapkan akan langsung kita bisa lihat hasilnya pada browser. Namun bagaimana dengan custom element? Apakah sama? Mari kita coba bersama.

Pada contoh sebelumnya, kita telah membuat element `<image-figure>` yang berfungsi layaknya element `<figure>` dengan penggunaan yang lebih sederhana. Contohnya kita memiliki kode seperti ini:

index.html image-figure.js

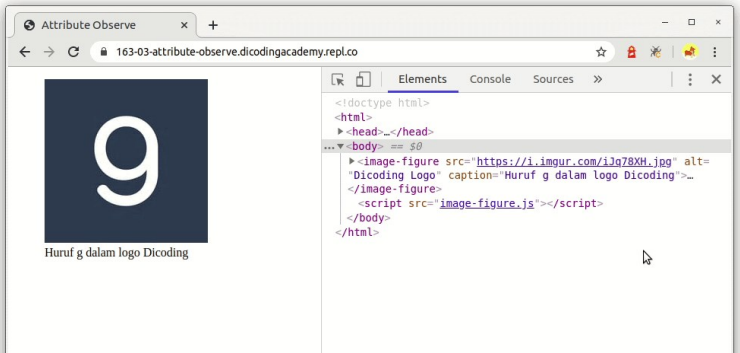
```
<!DOCTYPE html>
<html>
 <head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width">
   <title>Attribute Observe</title>
 </head>
<body>
   <image-figure
     src="https://i.imgur.com/iJq78XH.jpg"
     alt="Dicoding Logo"
     caption="Huruf g dalam logo Dicoding">
     </image-figure>
   <script src="image-figure.js"></script>
 </body>
</html>
```

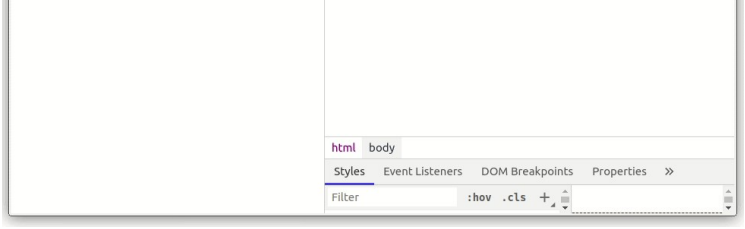
Lalu kita buka pada browser, maka akan tampak seperti ini:



Lalu mari kita coba mengubah nilai atribut ***caption*** dengan nilai baru menggunakan JavaScript melalui console browser.

```
const element = document.querySelector("image-figure");
element.setAttribute("caption", "Lorem ipsum dolor sit amet");
```





Kita bisa lihat pada gif di atas bahwa walaupun kita berhasil mengubah nilai atribut caption, pada browser nilai yang ditampilkan tersebut tidak berubah. Kok demikian? Pada custom element kita perlu mengimplementasi dua fungsi dalam kelasnya (`ImageFigure`) agar kita dapat mengobservasi nilai atribut yang berubah. Yang pertama fungsi `attributeChangedCallback` , dan yang kedua fungsi `static get observedAttributes.`

Kedua fungsi tersebut saling terhubung. Fungsi `attributeChangedCallback` akan terpanggil ketika nilai atribut yang diamati pada fungsi `observedAttributes` diubah nilainya. Kemudian pada callback fungsi `attributeChangedCallback` inilah kita menetapkan logika perubahan. Pada fungsi ini juga terdapat 3 (tiga) argument fungsi yang bisa dimanfaatkan yaitu:

- `name` : Nama dari atribut yang berubah
- `oldValue` : Nilai pada atribut sebelum diubah
- `newValue` : Nilai baru yang ditetapkan pada atribut.

Urutan dari ketiga argumen tersebut sangatlah penting, jadi jangan sampai tertukar. Sebenarnya kita dapat memberikan nama apa saja untuk ketiga argumennya namun lebih baik gunakan `name` , `oldValue` , `newValue` guna memudahkan kita dalam penggunaannya.

Berikut contoh implementasi dari kedua fungsi tersebut:

```
class ImageFigure extends HTMLElement {

  connectedCallback() {
    this.src = this.getAttribute("src") || null;
    this.alt = this.getAttribute("alt") || null;
    this.caption = this.getAttribute("caption") || null;
    this.render();
  }

  render() {
    this.innerHTML = `
      <figure>
        <img src="${this.src}"
            alt="${this.alt}">
        <figcaption>${this.caption}</figcaption>
      </figure>
    `;
  }

  attributeChangedCallback(name, oldValue, newValue) {
    this[name] = newValue;
    this.render();
  }
```

Mari kita telaah kodenya satu persatu. Di dalam fungsi `attributeChangedCallback()` , kita tuliskan kode untuk mengubah nilai properti `this[name]` dengan nilai baru yang ditetapkan. `this[name]` ini merupakan cara dinamis untuk mengubah nilai properti sesuai dengan nama atribut yang diubah. Misalkan jika kita mengubah atribut `“caption”` maka akan mengubah nilai `this[“caption”]` , jika kita mengubah atribut `“alt”` maka akan mengubah nilai `this[“alt”]` .

Setelah mengubah nilainya lalu kita panggil fungsi `render()` . Perhatikan juga bahwa kita perlu memisahkan kode rendering HTML di browser pada fungsi yang terpisah (tidak dilakukan di `connectedCallback`). Tujuannya agar kita dapat memanggil fungsi tersebut tidak hanya sekali tapi setiap kali terdapat perubahan data.

Lalu terdapat juga `static get observedAttributes()` . Apa fungsinya? Fungsi *getter* statis ini berperan sebagai “seseorang” yang mengamati perubahan nilai pada atribut yang ditentukan. Jika terjadi perubahan, ia akan memanggil `attributeChangedCallback` dengan memberitahu nama atribut apa yang berubah, nilai sebelum perubahan, serta nilai baru yang akan ditetapkan. `observedAttributes` tidak akan mengamati seluruh atribut yang diterapkan pada custom element, hanya atribut yang dituliskan pada nilai kembaliannya yang akan diamati.

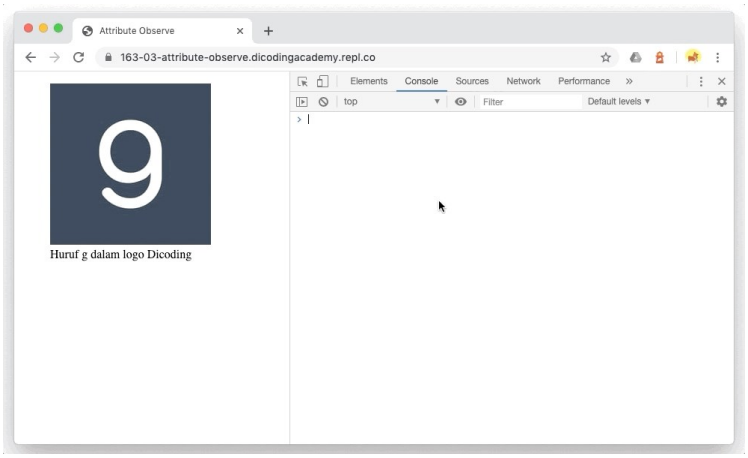
```
  return ["caption"];
```

Nilai kembalian dari `observedAttributes` merupakan array. Jika kita ingin mengamati lebih dari satu atribut, kita dapat menuliskannya layaknya array literals.

```
  return ["caption", "src", "alt"];
```

Setelah mengimplementasi kedua fungsi tadi seharusnya custom element sudah dapat bereaksi

ketika terjadi perubahan nilai atribut.



← KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA →