



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)



## Basic Usage

Potongan kode untuk materi ini:

- <https://repl.it/@dicodingacademy/163-03-shadow-dom-basic-usage?lite=true>
- <https://repl.it/@dicodingacademy/163-03-shadow-dom-styling?lite=true>

Untuk melampirkan Shadow DOM pada elemen penggunaan sangat mudah, yaitu dengan menggunakan properti `attachShadow` pada elemen-nya seperti ini:

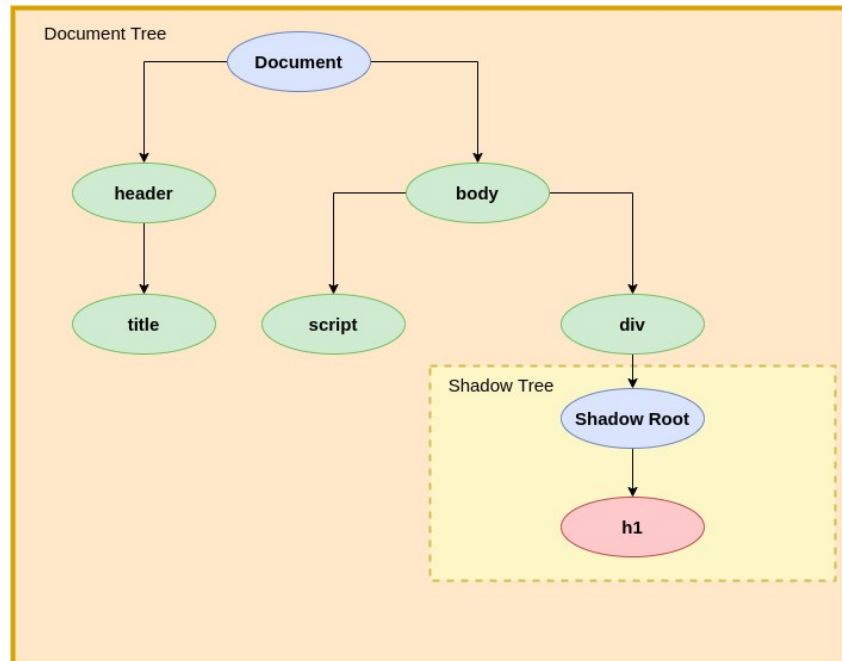


main.js index.html

```
1. // Shadow Host
2. const divElement = document.createElement("div");
3.
4.
5. // element yang berada di dalam Shadow DOM
6. const headingElement = document.createElement("h1");
7. headingElement.innerText = "Ini merupakan konten di dalam shadow DOM";
8.
9.
10. // Melampirkan shadow root pada shadow host
11. // Mengatur mode shadow dengan nilai open
12. const shadowRoot = divElement.attachShadow({mode: "open"});
13.
14.
15. // Memasukkan element heading ke dalam shadow root
16. shadowRoot.appendChild(headingElement);
17.
18.
19. // Memasukkan elemen shadow host ke regular DOM
20. document.body.appendChild(divElement);
```

Jika kita lihat pada browser, maka struktur HTML yang akan dihasilkan adalah seperti ini:

Dan struktur DOM tree yang terbentuk akan tampak seperti ini:

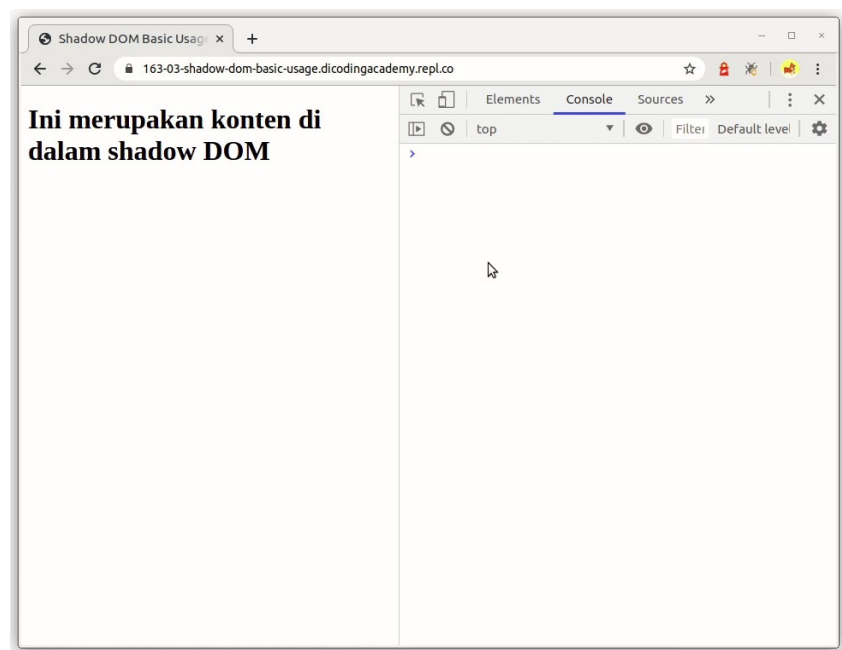


Dalam penggunaan `attachShadow()` kita melampirkan objek dengan properti `mode` yang memiliki nilai `'open'`. Sebenarnya terdapat dua opsi nilai yang dapat digunakan dalam properti mode, yaitu `"open"` dan `"closed"`.

Menggunakan nilai open berarti kita memperbolehkan untuk mengakses properti `shadowRoot` melalui elemen yang melampirkan Shadow DOM.

```
1. divElement.attachShadow;
```

properti `shadowRoot` mengembalikan struktur DOM yang berada pada shadow tree.

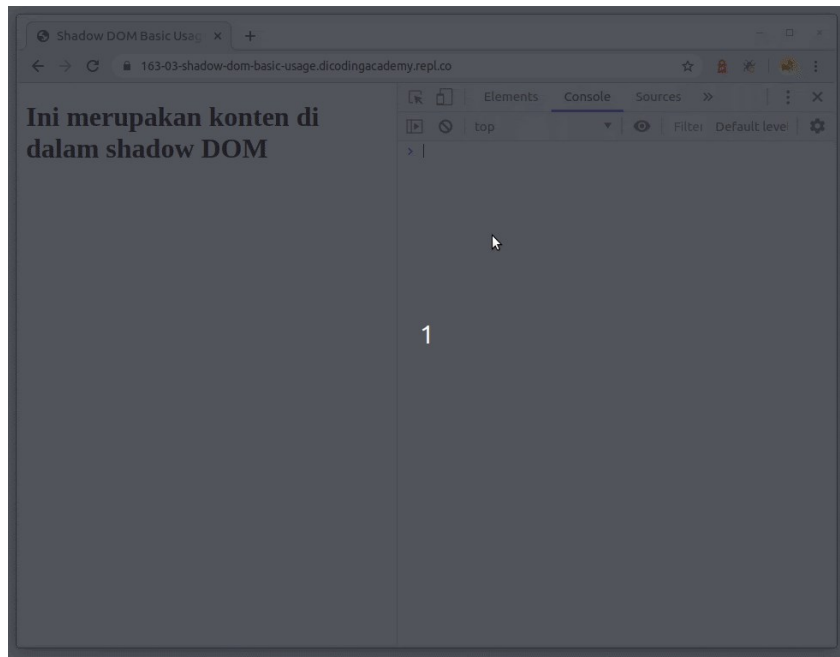


Namun jika kita menggunakan nilai `closed` maka properti `shadowRoot` akan mengembalikan nilai `null`.

```
1. const shadowRoot = divElement.attachShadow({mode: "closed"});
2. divElement.shadowRoot // null;
```

Hal ini berarti kita sama sekali tidak dapat mengakses Shadow Tree selain melalui variabel yang kita definisikan ketika melampirkan Shadow DOM.

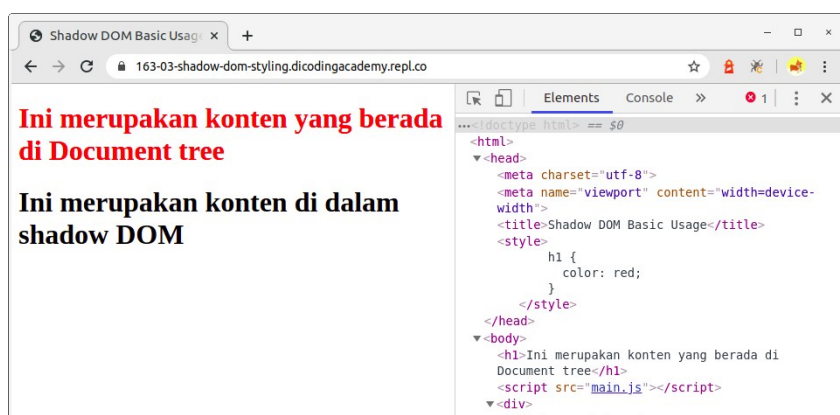
```
1. const shadowRoot = divElement.attachShadow({mode: "closed"});
2. divElement.shadowRoot // null;
3. shadowRoot // # shadow-root (closed)
```

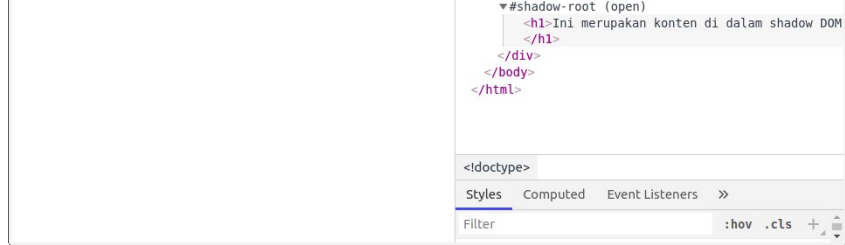


Karena Shadow DOM terisolasi dari document tree maka element yang terdapat di dalamnya pun tidak akan terpengaruh oleh styling yang berada diluar dari shadow root-nya.



Jika dilihat pada browser maka hasilnya akan seperti ini:



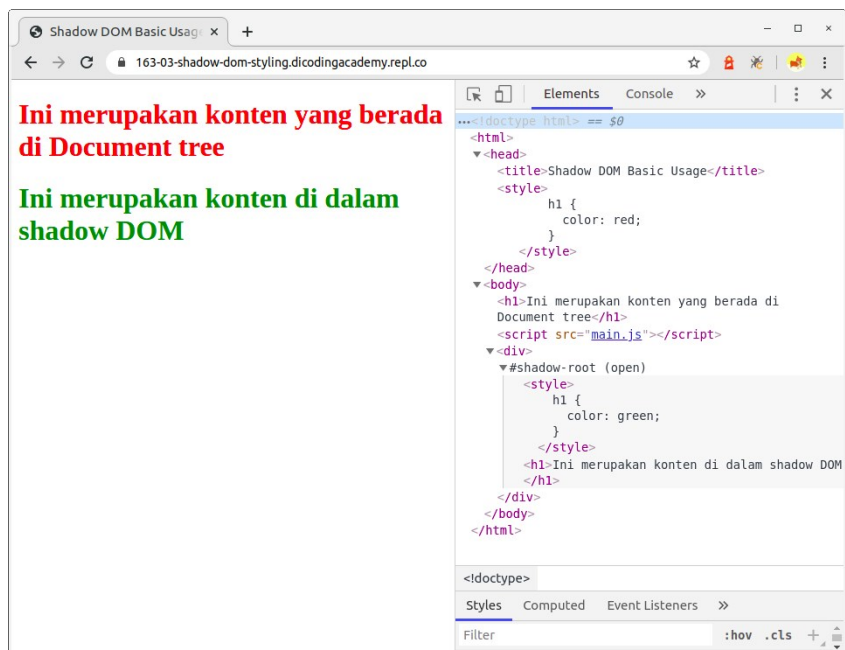


Berdasarkan hasil di atas, styling hanya akan diterapkan pada elemen `<h1>` yang berada di document tree. Sedangkan elemen `<h1>` yang berada pada shadow dom tidak akan terpengaruh dengan styling tersebut. Lantas, bagaimana caranya kita melakukan styling pada Shadow DOM?

Kita dapat melakukannya dengan menambahkan template `<style>` di dalam `shadowRoot.innerHTML`. Contohnya seperti ini:



Maka element `<style>` tersebut akan berada di dalam shadow tree dan akan berdampak pada elemen yang ada di dalamnya.



[← KEMBALI KE MATERI SEBELUMNYA](#)

[LANJUTKAN KE MATERI BERIKUTNYA →](#)



Job

Rewards

