SHADOW DOM

Dengan SHADOW DOM, element mendapatkan sebuah konten baru yang berhubungan dengan element tersebut.
konten baru tersebut kita sebut shadow root, sedangkan element yang didalamnya mengandung konten / kontent baru bisa kita sebut shadow host.
Dengan konsep SHADOW DOM, konten dasar di dalam shadow root tidak akan dirender, melainkan konten baru yang sengaja kita buat konsep SHADOW DOM

SIMPLE WORKSHOP

1. Buat file html baru

```
<!DOCTYPE html>
    <head>
        <title>SHADOW DOM</title>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link href="./css/***.css" rel="stylesheet">
        <script src="./app/webcomponents-lite.js"></script>
    </head>
    <body>

    </body>
</html>
```

2. Tambahkan tag element baru

```
<p>hi, saya tag html biasa</p>
```

3. Cara 1 : Aktifasi SHADOW DOM untuk elemen “p”

```
var host = document.querySelector(‘p’);
var root = host.createShadowRoot();
root.textContent = ‘bohong, saya adalah tag html dengan shadow dom’;
```

4. Apa yang tertulis pada halaman website anda ?? Ternyata browser merender kaliman “bohong, saya adalah tag html dengan shadow dom” dibandingkan kalimat dasar tag p “hi, saya tag html biasa”

Tidak hanya mengganti konten di dalam element, jika dalam javascript textContent kita letakkan sebuah perintah / logic, maka kita bisa menggunakan SHADOW HOST sebagai sebuah widget interaktif

Contoh :

1. Buat tag tambahan

```
<div class=”nama”>
    <p>Mifan Twan Ardana</p>
</div>
```

2. Buat tag template tambahan yang akan menggantikan tag dengan class “nama”

```
<template class=”nama-template”>
    <h1>MIFAN TWAN ARDANA</h1>
    <p>nah sekarang kenal kan?</p>
</template>
```

3. Buat file javascript internal maupun external

```
var shost = document.querySelector('.nama');
var sroot = shost.createShadowRoot();
var stemplate = document.querySelector('.nama-template');
root.appendChild(document.importNode(stemplate.content, true));
```

4. Sekarang elemen anda digantikan dengan elemen dengan dokumen baru dengan template yang berbeda
5. Secara keseluruhan, bisa kita lihat kode full dari file html SHADOW DOM yang kita buat :

```
<!DOCTYPE html>
    <head>
        <title>SHADOW DOM</title>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link href="./css/***.css" rel="stylesheet">
        <script src="./app/webcomponents-lite.js"></script>
    </head>
    <body>
        <p>hi, saya tag html biasa</p>
        <div class="nama">
            <p>Mifan Twan Ardana</p>
        </div>
        <template class="nama-template">
            <h1>MIFAN TWAN ARDANA</h1>
            <p>nah sekarang kenal kan?</p>
```

```
        </template>
        <script>
            var host = document.querySelector('p');
            var root = host.createShadowRoot();
            root.textContent = 'bohong, saya adalah tag html dengan shadow dom';
        </script>
        <script>
            var shost = document.querySelector('.nama');
            var sroot = shost.createShadowRoot();
            var stemplate = document.querySelector('.nama-template');
            root.appendChild(document.importNode(stemplate.content, true));
        </script>
    </body>
</html>
```

HTML IMPORT

Html import adalah bagian dimana kita bisa menambahkan file .html lain kedalam file .html utama yang sedang kita kerjakan, kita bisa menambahkan file .html sesuka hati dengan paket bersama css, javascript maupun file lain. Dengan kata lain, import adalah tool utama kita untuk menambahkan file html/css/js/dll dengan mengubahnya menjadi bagian bagian kecil yang mudah dikembangkan.

SIMPLE WORKSHOP

1. Tambahkan tag link di element head

```
<head>
    <link rel=”import” href=”./html/header.html”>
</head>
```

2. Buat element baru dalam file baru sesuai dengan struktur folder yang kita import dalam tag baru tadi

```
<header>
    <h1>ini file header saya</h1>
</header>
```

3. Html belum bisa di’import sampai langkah ini, untuk bisa merender file header.html kita perlu menambahkan file javascript internal maupun external seperti ini

```
var link = document.querySelector(‘link[rel=”import”]’);
var content = link.import;
var el = content.querySelector(‘header’);
```

```
document.body.appendChild(el.cloneNode(true));
```

4. Secara keseluruhan, bisa kita lihat kode full .html seperti ini

```
<!DOCTYPE html>
    <head>
        <title>HTML IMPORT</title>
        <meta name=”viewport” content=”width=device-width, initial-scale=1”>
        <link href=”./css/***.css” rel=”stylesheet”>
        <script src=”./app/webcomponents-lite.js”></script>
        <link href=”./html/header.html” rel=”import”>
    </head>
    <body>
        <script type=”text/javascript”>
            var link = document.querySelector(‘link[rel=”import”]’);
            var content = link.import;
            var el = content.querySelector(‘header’);
            document.body.appendChild(el.cloneNode(true));
        </script>
    </body>
</html>
```

5. Perhatikan yang terjadi pada browser

HTML TEMPLATE

Konsep HTML Template adalah fitur paling menarik dari semua web framework yang tersedia bebas di internet, konsep ini sudah lama dan mengalami banyak perubahan untuk mendapatkan penggunaan paling mudah sejak munculnya framework MVC. Setiap framework menyediakan fitur logic “VIEW” yang tugasnya me-render halaman basic presentasi pada browser.

SIMPLE WORKSHOP

1. Untuk membuat sebuah konten template, deklarasi kan element “template” pada file html

```
<template>
    <div>
        Template yang saya gunakan : <span>0</span>
    </div>
    <script>
        alert(‘Hei, template nya berhasil ditambahkan !’);
    </script>
</template>
```

elemen “template” tidak akan tampil pada halaman browser sampai diaktifasi, elemen ini dianggap sebagai hidden DOM dan tidak akan direnderoleh browser pada kondisi default

semua konten dalam elemen “template” baik berupa text, multimedia file seperti audio, video, dll tidak akan mempengaruhi kinerja file .html sampai elemen ini benar benar digunakan.

konten di dalam elemen “template” dianggap tidak ada dalam document, menggunakan document.getElementById() atau querySelector() untuk konten di dalam elemen “template” tidak akan berpengaruh pada html render sampai elemen “template” di aktivasi

elemen “template” boleh ditambahkan di bagian mana saja dalam file html, boleh di <head></head>, </body></body>, dll. Artinya element “template” bisa diletakkan di tempat secure dimana HTML parser tidak diijinkan.
*html parse = mengubah text baku menjadi kode unik untuk di render HTML

2. Buat sebuah button yang tugasnya menambahkan element “template” pada file html yang sedang kita kelola.

```
<button onclick=”buatTemplate()”>tambahkan konten HTMK Template</button>
```

3. Tambahkan element host yang akan menjadi tempat untuk menampikan / render elemen “template”

```
<section></secttion>
```

4. Aktivasi element “template” dengan bantuan argumen javascript

```
<script type=”text/javascript”>
    function buatTemplate(){
        var content = document.querySelector(‘template’).content;
        var span = content.querySelector(‘span’);
        span.textContent = parseInt(span.textContent) + 1;
        document.querySelector(‘section’).appendChild(document.importNode(content, true));
    }
</script>
```

5. contoh file .html dengan fitur HTML Template :

```
<!DOCTYPE html>
```

```
<head>
    <title>HTML IMPORT</title>
    <meta name=”viewport” content=”width=device-width, initial-scale=1”>
    <link href=”./css/***.css” rel=”stylesheet”>
    <script src=”./app/webcomponents-lite.js”></script>
</head>
<body>
    <button onclick=”buatTemplate()”>
        Tambahkan konten HTML Template
    </button>
    <section></section>
    <script>
        function buatTemplate() {
            var content = document.querySelector(‘template’).content;
            var span = content.querySelector(‘span’);
            span.textContent = parseInt(‘section’).appendChild(
                document.importNode(content, true)
            );
        }
    </script>
    <template>
        <script>
            <div>
                Konten template yang digunakan : <span>0</span>
            </div>
            alert(‘Hei, HTML Template nya berhasil’)
```

```
            </script>
        </template>
    </body>
</htm>
```

6. Perhatikan yang terjadi pada browser