



Universitas Kristen Duta Wacana

Fakultas Teknologi Informasi

Prodi Informatika

Jln. Dr. Wahidin Sudiro Husodo No 5-25

Daerah Istimewa Yogyakarta 55224

Mata Kuliah	: Pemrograman Website
Kode Mata Kuliah	: TI0243
Periode	: Semester Genap 2020/2021
Dosen Pengampu	: Danny Sebastian S.Kom., M.M., M.T. : Maria Nila Anggia Rini S.T., M.Tl. : Agata Filiana S.Kom., M.Cs.

BAB 11

JAVASCRIPT (2)

A. CPL Pertemuan

- Menggunakan javascript untuk membuat interaktivitas halaman website

B. Alat & Bahan

Kegiatan praktikum pada bagian ini membutuhkan perangkat keras (*hardware*) atau komputer dengan spesifikasi minimum:

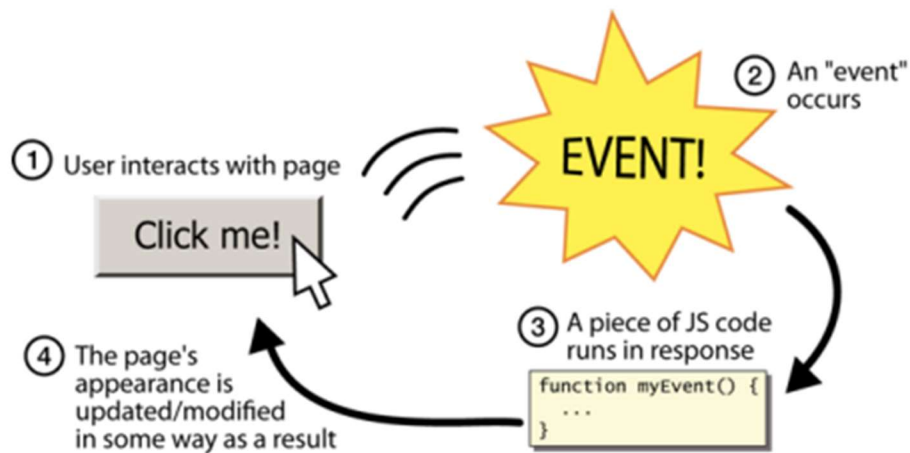
- Processor : Intel Core i3
- Memory : 2 GB
- Hard disk : kapasitas kosong penyimpanan sebesar 3 GB

Selain perangkat keras, kegiatan praktikum bagian ini juga membutuhkan perangkat lunak (*software*):

- Sistem Operasi (Windows, Linux, atau MacOSX)
- Text editor (Atom, Visual Studio Code, atau editor lain)
- Browser (Google Chrome, Opera, atau Mozilla Firefox)
- Xampp (Web Server Apache & Database MariaDB)

C. Event Handler pada Javascript

Pada JavaScript, kita bisa membuat sebuah fungsi, yaitu dengan mendeklarasikan “function” pada script yang telah dibuat dan fungsi tersebut dapat dipanggil dalam beberapa cara melalui event yang ada pada tag HTML.



Ketika pengguna memencet sebuah elemen html(1) maka *event* akan muncul(2), maka akan ada baris kode (yang biasanya berupa fungsi) yang akan merespon *event* tersebut(3), lalu halaman HTML akan melakukan sebuah perubahan(4). Contoh penggunaannya adalah sebagai berikut:

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Document</title>  
  </head>  
  <body>  
    <button onclick="myFunction()">Try it</button>  
    <p id="demo"></p>  
    <script>  
      function myFunction() {  
        alert("Halo anda sudah mentrigger sebuah event!")  
      }  
    </script>  
  </body>  
</html>
```

Event terletak pada tag `<button>` dan *event* tersebut adalah `onClick="myFunction()"` yang berarti ketika button tersebut dipencet maka akan memanggil fungsi `myFunction()`.

Berikut ini adalah beberapa event dalam JavaScript yang sering digunakan, selengkapnya dapat dilihat di http://www.w3schools.com/jsref/dom_obj_event.asp

- **onClick** : keadaan dimana elemen diklik oleh user (misalnya : link, button).
- **onLoad** : dipanggil saat elemen/dokumen berhasil di-load .
- **onUnload** : dipanggil saat meninggalkan halaman.
- **onFocus** : dipanggil saat suatu tag input di klik/difokus.
- **onBlur** : dipanggil saat suatu tag input tidak lagi difokus.
- **onChange** : dipanggil saat tag input mengalami perubahan value.
- **onSubmit** : dipanggil saat suatu form disubmit.
- **onMouseOver** : dipanggil saat suatu tag mengalami hover.
- **onMouseOut** : dipanggil saat suatu mouse keluar dari tag.

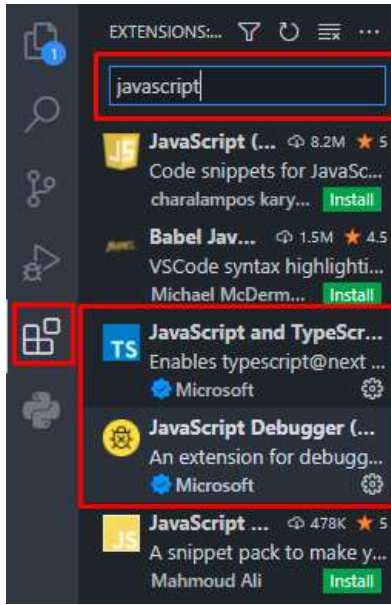
Contoh penggunaan event dan function pada JavaScript:

```
<!DOCTYPE html>
<html>
<head>
  <title>Test JavaScript - Event and Function</title>
  <script type="text/javascript">
    function test(){
      alert("fungsi test() ini dipanggil oleh event onLoad!!");
    }
  </script>
</head>
<body onLoad="test()">
  <a href="#" onMouseOver="alert('kena hover')">Hover me</a>
  <button onClick="alert('Button diklik')">Click Me</button>
</body>
</html>
```

D. Javascript Selector

Selector digunakan ketika kita ingin mengambil/memilih sebuah element yang nantinya kita dapat melakukan perubahan pada element tersebut. Contohnya kita memiliki sebuah element **<h1> Halo </h1>** dan kita akan mengubah value String “Halo” menjadi “Hello” menggunakan javascript melalui event onclick. Untuk melakukan select sebuah element dapat dilakukan dengan beberapa cara seperti berikut :

Sebelum memulai menulis baris kode javascript, disarankan untuk menginstal ekstensi di bawah agar memudahkan kita untuk menulis kode javascript. Instal ekstensi yang ada didalam persegi warna merah berikut :



D.1. Selector getElementById

Untuk melakukan select menggunakan metode ini dapat dilakukan dengan perintah berikut :

```
document.getElementById("demo");
```

Metode select di atas akan melakukan select element yang memiliki id dengan nama demo. Contohnya element h1 di bawah ini :

```
<h1 id="demo">Halo dunia!</h1>
```

Untuk cara penggunaan select by id ini dapat dilakukan langkah-langkah berikut.

1. Buat terlebih dahulu halaman website seperti berikut, dan buka halaman tersebut di browser Anda.

```
index.php X
index.php > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
```

2. Kemudian tambahkan element H1 beserta id nya seperti di bawah ini. (highlight persegi warna biru).

```
index.php X
index.php > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10      <h1 id="demo">Halo dunia!</h1>
11 </body>
12 </html>
```

3. Lalu kita buat tag <script> untuk memulai membuat baris kode javascript. Buat tag <script> di dalam tag <head> seperti di bawah ini.

```
index.php X
index.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8
9   <script>
10     // kita akan menuliskan baris kode
11     // javascript di sini
12
13   </script>
14 </head>
15 <body>
16   <h1 id="demo">Halo dunia!</h1>
17 </body>
18 </html>
```

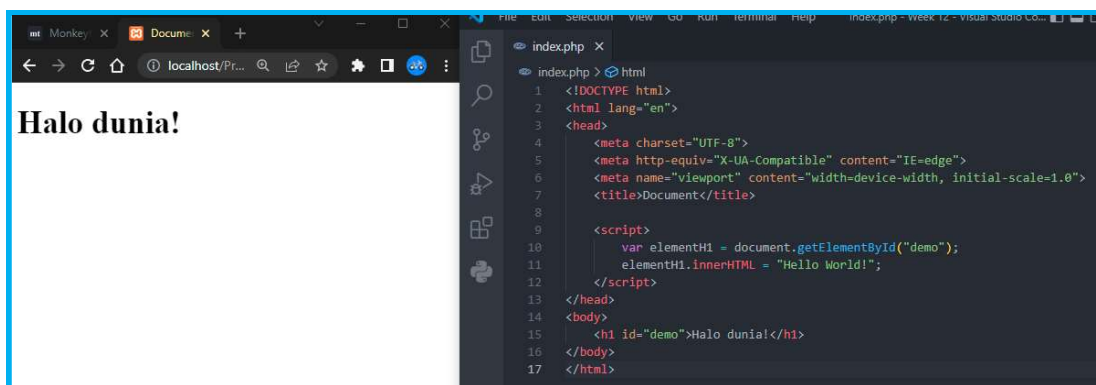
4. Kemudian untuk mengubah nilai string “Halo dunia!” yang ada di element H1. Kita perlu melakukan select terlebih dahulu. Baris kode select akan kita tampung dalam sebuah variabel agar bisa dilakukan perubahan pada element H1 yang di select tadi.

```
index.php X
index.php > html > head > script
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8
9   <script>
10     var elementH1 = document.getElementById("demo");
11
12   </script>
13 </head>
14 <body>
15   <h1 id="demo">Halo dunia!</h1>
16 </body>
17 </html>
```

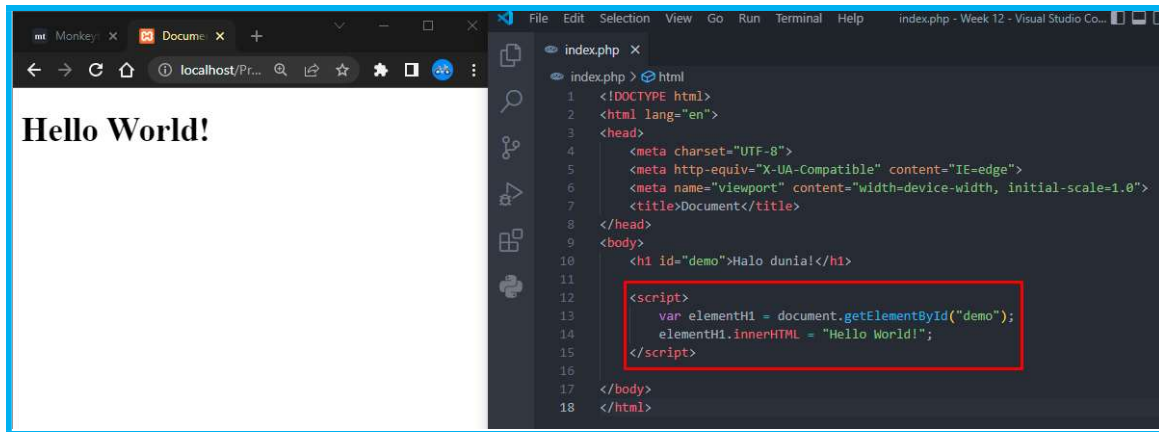
5. Lalu kita mengubah string “Halo dunia!” yang ada di element H1 dengan cara di bawah ini.

```
index.php X
index.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8
9   <script>
10     var elementH1 = document.getElementById("demo");
11     elementH1.innerHTML = "Hello World!";
12   </script>
13 </head>
14 <body>
15   <h1 id="demo">Halo dunia!</h1>
16 </body>
17 </html>
```

6. Mari kita lihat hasil nya di browser terlebih dahulu.



Ternyata element string H1 belum berubah. Hal dikarenakan ketika pembacaan baris kode urut dari atas, kita mencoba mengubah Element H1 yang belum terbaca (perintah Javascript dibaca lebih dahulu dari element H1), ini mengapa element string H1 tidak berubah. Agar kita bisa mengubah element H1, maka kita akan taruh <script> di bawah h1 seperti berikut.



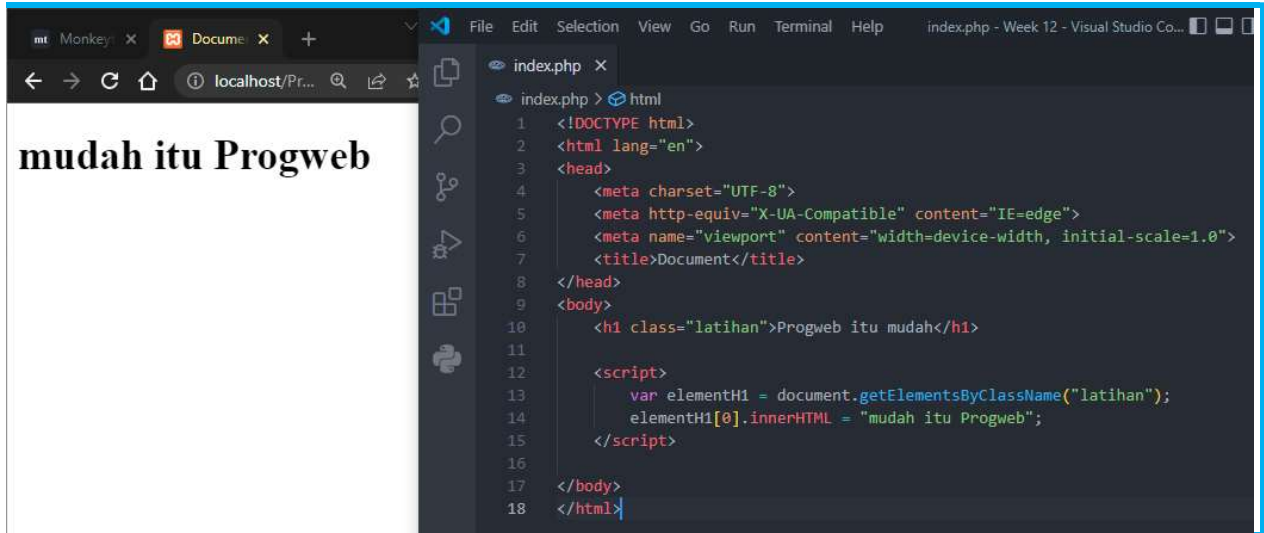
Nah, jika `<script>` di taruh di bawah H1, maka string "Halo dunia!" akan terganti menjadi "Hello World!" seperti yang ada di baris kode javascript.

D.2. Selector `getElementsByClassName`

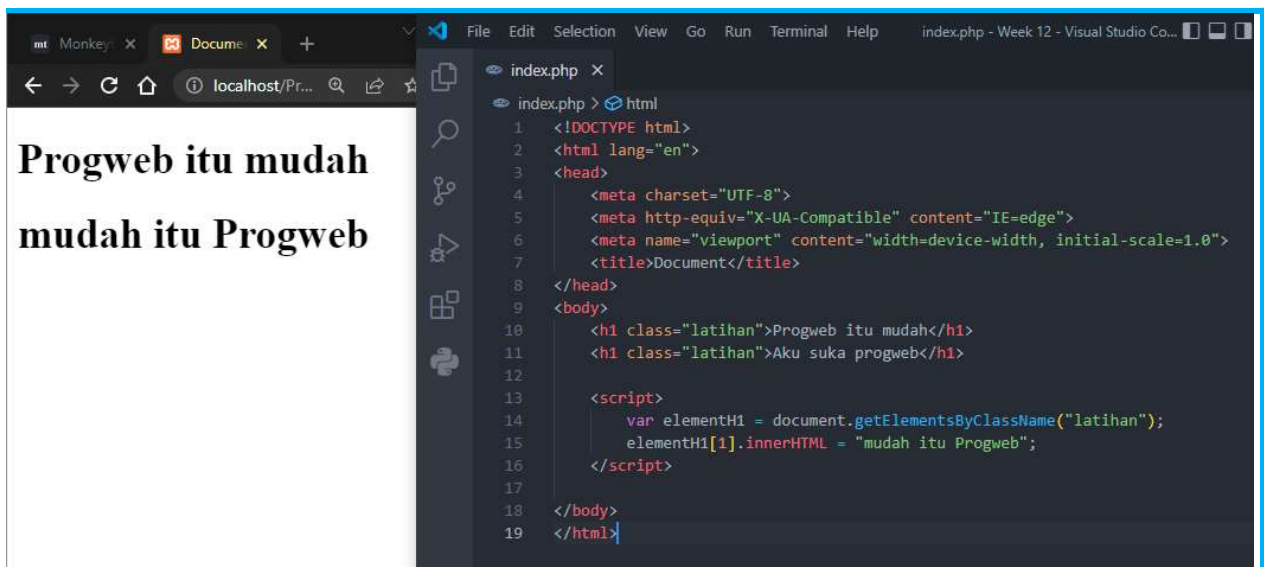
Pada metode select berikut memiliki konsep yang hampir sama seperti sebelumnya, kali ini yang membedakan terdapat pada cara melakukan select element dimana element tersebut memiliki sebuah class. Untuk perintah select methode ini dapat dilakukan dengan cara di bawah ini.

```
document.getElementsByClassName("namaKelas");
```

Untuk praktiknya kita akan membuat sebuah element H1 lagi dengan string "Progweb itu mudah", dan memberikan class pada element ini, lalu kita akan mengubah nilainya. Perhatikan baris kode di bawah ini.



Nah, string "Progweb itu mudah" sudah berubah menjadi seperti di atas. Namun jika Anda lihat lagi, mengapa ada indeks [0] pada saat akan mengubah string seperti diatas. Hal ini dikarenakan ketika kita melakukan select menggunakan methode by class, maka baris kode ini akan mengembalikan sebuah list/array yang berisi kumpulan element yang menggunakan class tersebut (akan mengembalikan semua element yang menggunakan class="latihan" dalam bentuk list). Dan karena list yang kita terima hanya memiliki 1 item, maka kita dapat mengakses item tersebut dengan memanggilnya menggunakan indeks ke [0]. Untuk lebih jelasnya perhatikan gambar di bawah ini yang mengubah element indeks ke 1, sementara element indeks ke 0 (**<h1 class="latihan">Progweb itu mudah</h1>**) tidak akan berubah.



D.3. Selector `getElementsByName`

Seperti yang kita tahu HTML memiliki beberapa tag seperti :

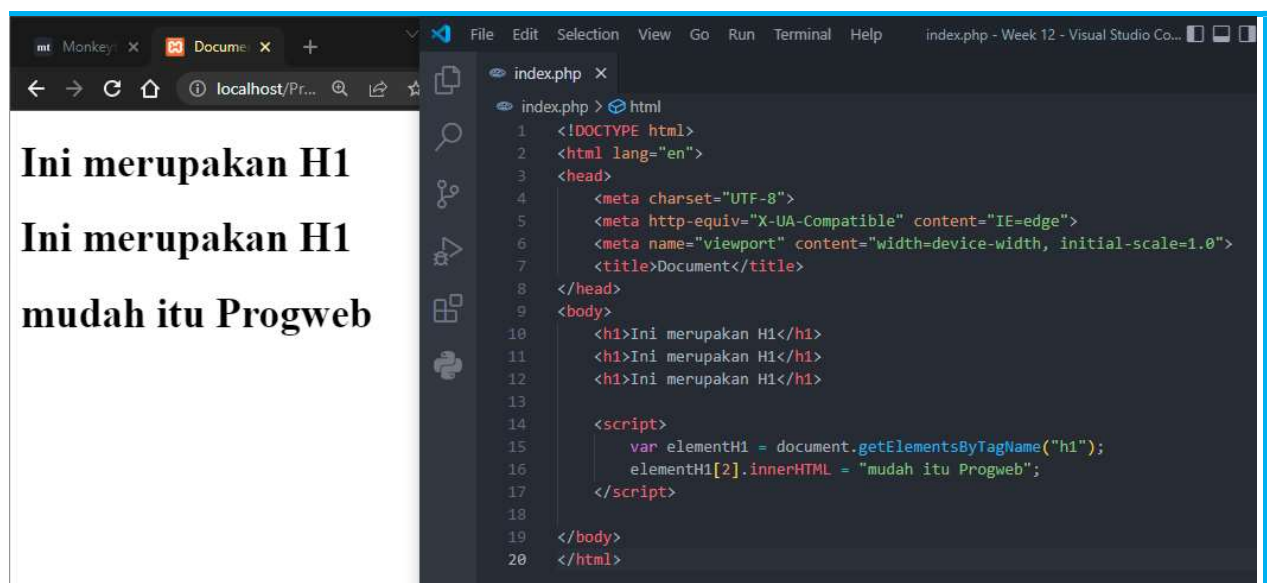
<code><address></code>	<code><article></code>	<code><aside></code>	<code><blockquote></code>	<code><canvas></code>	<code><dd></code>	<code><div></code>
<code><dl></code>	<code><dt></code>	<code><fieldset></code>	<code><figcaption></code>	<code><figure></code>	<code><footer></code>	<code><form></code>
<code><h1>-<h6></code>	<code><header></code>	<code><hr></code>	<code></code>	<code><main></code>	<code><nav></code>	<code><noscript></code>
<code></code>	<code><p></code>	<code><pre></code>	<code><section></code>	<code><table></code>	<code><tfoot></code>	<code></code>
<code><video></code>						
<code><a></code>	<code><abbr></code>	<code><acronym></code>	<code></code>	<code><bdo></code>	<code><big></code>	<code>
</code>
<code><button></code>	<code><cite></code>	<code><code></code>	<code><dfn></code>	<code></code>	<code><i></code>	<code></code>
<code><input></code>	<code><kbd></code>	<code><label></code>	<code><map></code>	<code><object></code>	<code><output></code>	<code><q></code>
<code><samp></code>	<code><script></code>	<code><select></code>	<code><small></code>	<code></code>	<code></code>	<code><sub></code>
<code><sup></code>	<code><textarea></code>	<code><time></code>	<code><tt></code>	<code><var></code>		

Nah kali ini kita akan melakukan select berdasarkan nama tag di atas.

Konsep select nya mirip seperti saat kita melakukan select menggunakan `getElementByClassName` yang mengembalikan kumpulan element, namun pada metode kali ini yang akan dihasilkan adalah sebuah kumpulan element tag yang memiliki `tagName` yang sesuai. Untuk perintah metode select ini dapat dilakukan dengan cara di bawah :

```
document.getElementsByTagName("tagName_yang_digunakan");
```

Kemudian untuk praktiknya dapat dilakukan dengan cara berikut.



E. More of Javascript

Javascript dapat kita gunakan untuk memodifikasi element HTML yang kita buat misalnya melakukan tambah, hapus, ubah element HTML sesuai event yang diperlukan. Javascript berinteraksi dengan HTML, CSS dan data untuk mengatur warna, ukuran, dan bentuk. Bahasa tersebut sering digunakan untuk menciptakan efek khusus pada tampilan web. Contohnya dapat kita temui pada website dengan pendekatan native app disebut single page application (SPA), yang bergantung pada JavaScript untuk update tampilan web setiap terjadi interaksi dengan user, tanpa mengharuskan adanya pengiriman request atau data ke server.

E.1. Memperoleh Data

.value digunakan untuk memperoleh value konten dari suatu element.

```
var a = document.getElementById("idName").value;
var b = document.getElementsByClassName("className").value;
var c = document.getElementsByTagName("p").value;
```

.getAttribute digunakan untuk memperoleh value dari attribute yang diminta pada suatu element.

```
var a = document.getElementById("idName").getAttribute("class");
var b =
document.getElementsByClassName("className").getAttribute("id");
var c = document.getElementById("idName").getAttribute("href");
```

.tagName digunakan untuk memperoleh tag dari suatu element.

```
var a = document.getElementById("idName").tagName;
```

.textContent, .innerText, dan innerHTML dapat digunakan untuk memperoleh text dari suatu element.

```
var a = document.getElementById("idName").textContent;
var b = document.getElementById("idName").innerText;
var c = document.getElementById("idName").innerHTML;
```

E.2. Penambahan Konten

E.2.1. Konten HTML

Berikut beberapa langkah-langkah yang digunakan untuk menambah konten element HTML :

.createTextNode() digunakan untuk membuat text node pada HTML. Sehingga value dari variabel a berupa node "SomeText".

```
var a = document.createTextNode("Some Text");
```

.createElement() digunakan untuk membuat tag element pada HTML seperti p, label, input, button, div, span, dll. Sehingga value variabel b berupa tag adalah <p></p>.

```
var b = document.createElement("p");
```

.appendChild() digunakan untuk menambahkan text ke element yang telah dibuat. Value dari variabel b menjadi <p>Some Text</p>.

```
b.appendChild(a);
```

Lakukan langkah ini untuk menambahkan element ke element lain sesuai id yang ditentukan. Maka <p>Some Text</p> ditambahkan pada element HTML yang memiliki id= "name".

```
var c = document.getElementById("idName");  
c.appendChild(b);
```

E.2.2. Attribute

.setAttribute untuk mengeset attribute seperti id, class, src, dan href pada suatu element HTML.

```
document.getElementById("idName").setAttribute("class", "kiri");  
document.getElementsByTagName("p")[0].setAttribute("id", "t1p");
```

E.2.3. Event

.addEventListener digunakan untuk menambahkan event seperti click, mouseover, keyup, dan keydown pada suatu element.

```
document.getElementById("idName").addEventListener("click",  
functionName);
```

E.3. Penghapusan Konten

E.3.1. Konten HTML

Menghapus element HTML dapat dilakukan dengan cara menghapus child dari parent-nya dengan perintah sebagai berikut:

```
parent.removeChild(node);
```

Contoh:

```
<div id="makanan">
  <p>Bakso</p>
  <p>Mie Ayam</p>
  <p>Nasi Goreng</p>
</div>
<script>
  var makanan = document.getElementById("makanan");
  var m = document.getElementsByTagName("p");
  makanan.removeChild(m[1]);
</script>
```

E.3.2. Attribute

.removeAttribute adalah perintah untuk menghapus attribute pada suatu element.

```
document.getElementById("idName")[1].removeAttribute("class");
document.getElementsByTagName("div")[0].removeAttribute("id");
```

E.3.3. Event

.removeEventListener digunakan untuk menghapus event pada suatu element.

```
document.getElementById("idName").removeEventListener("click",
functionName);
```

E.4. Pengubahan Konten

Ada beberapa cara yg digunakan untuk mengubah element HTML sesuai kebutuhan, seperti :

.innerHTML digunakan untuk mengubah konten suatu element HTML.

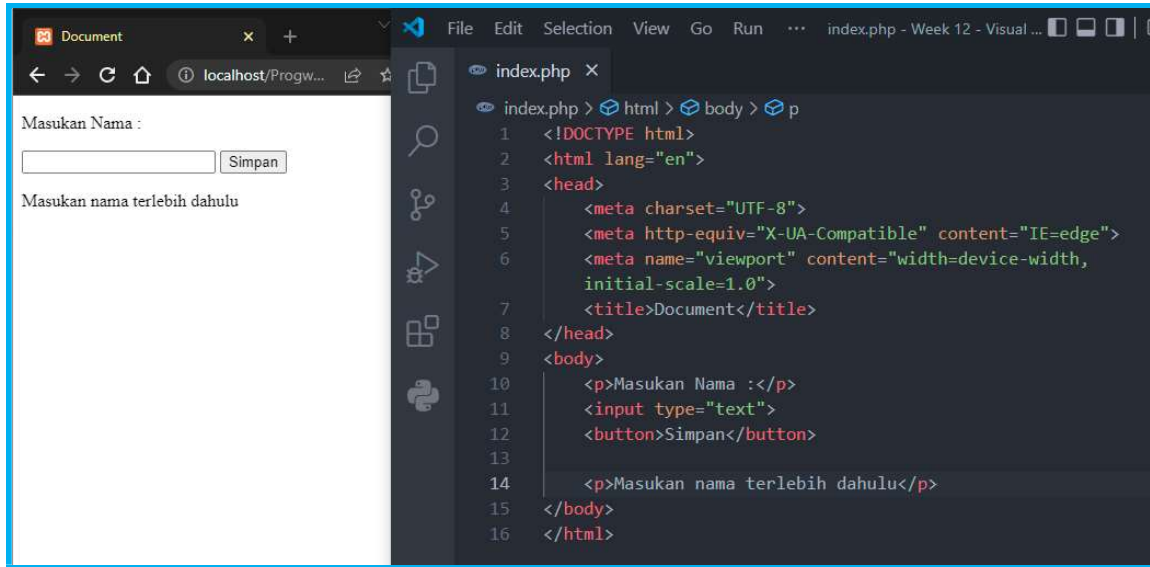
```
<div id="idName"></div>
<script>
  document.getElementById("idName").innerHTML = "text";
</script>
```

F. Guided

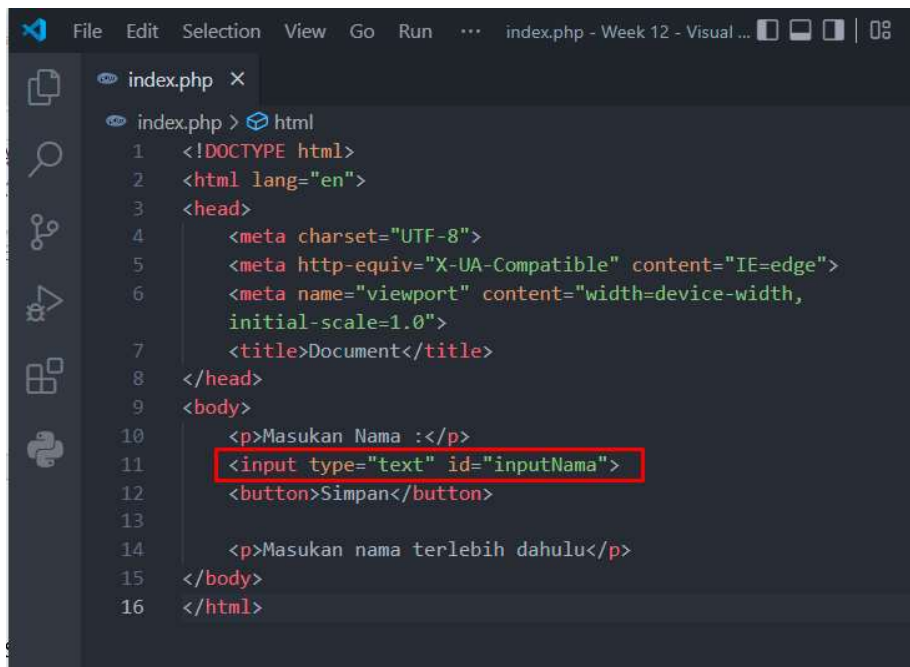
F.1. Latihan Selector 1

Kali ini kita akan mencoba menampilkan nama dari sebuah textfield ke alert juga mengubah string dari sebuah element.

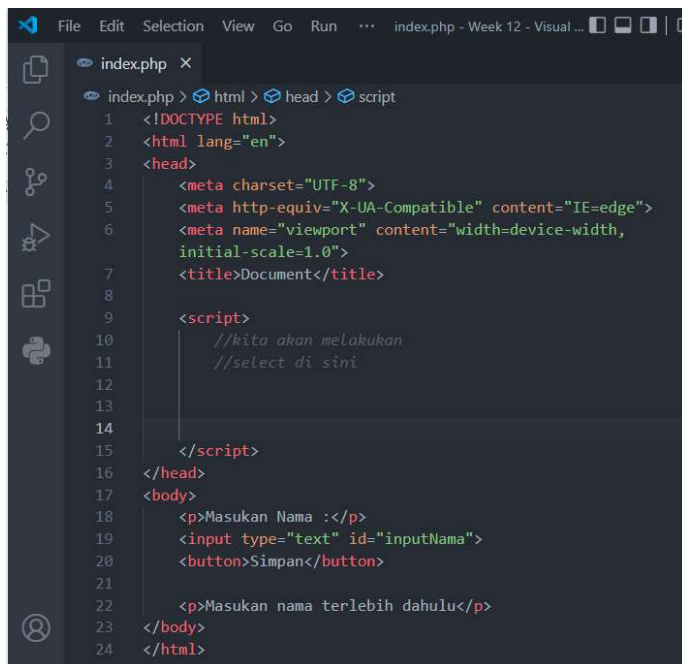
1. Pertama kita buat dahulu sebuah halaman website seperti berikut.



2. Setelah itu kita akan melakukan select terhadap nama yang dimasukan ketika tombol simpan di klik.
Pertama buat id di inputan nama dan lakukan select by id.

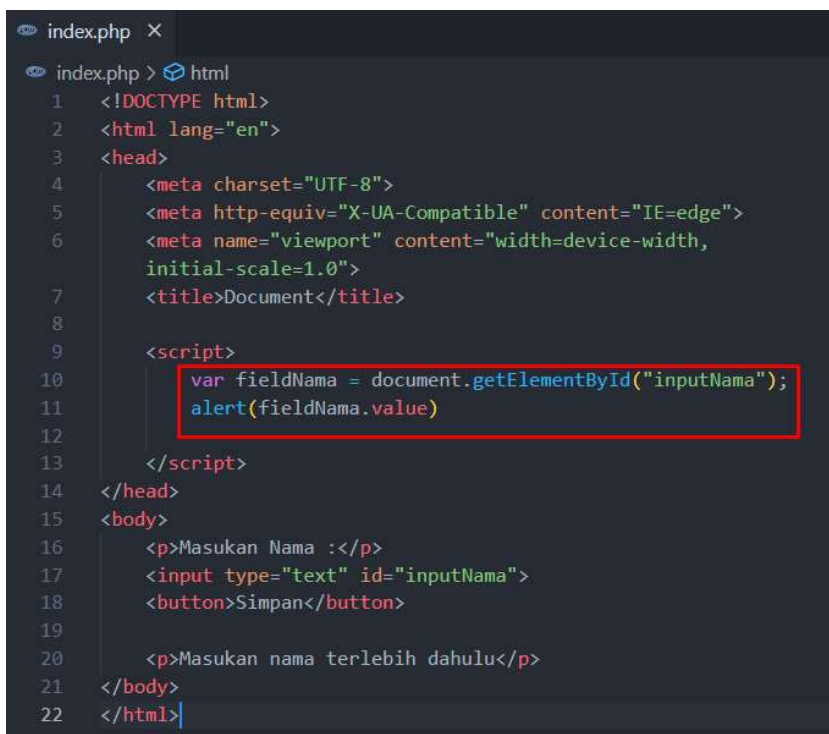


3. Pastikan `<script>` berada di dalam tag `<head>` karena kita akan mengubah string setelah menekan tombol nya dahulu.



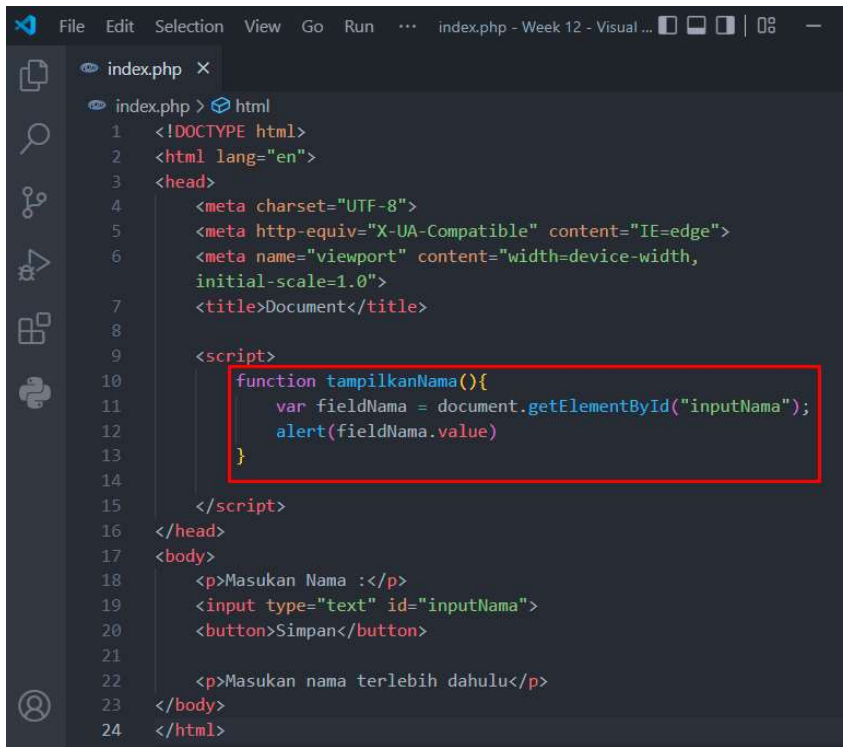
```
index.php X
index.php > html > head > script
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,
7     initial-scale=1.0">
8   <title>Document</title>
9
10  <script>
11    //kita akan melakukan
12    //select di sini
13
14  </script>
15
16 </head>
17 <body>
18   <p>Masukan Nama :</p>
19   <input type="text" id="inputNama">
20   <button>Simpan</button>
21
22   <p>Masukan nama terlebih dahulu</p>
23 </body>
24 </html>
```

4. Kemudian kita akan melakukan select inputan nama tersebut dan menampilkannya di window alert dengan cara seperti di bawah ini. Pertama kita select inputan nama dan ambil stringnya (nama user). Untuk mengambil string dari sebuah tag <input> kita menggunakan .value, tidak seperti tag HTML sebelumnya yang menggunakan .innerHTML .



```
index.php X
index.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,
7     initial-scale=1.0">
8   <title>Document</title>
9
10  <script>
11    var fieldNama = document.getElementById("inputNama");
12    alert(fieldNama.value)
13  </script>
14 </head>
15 <body>
16   <p>Masukan Nama :</p>
17   <input type="text" id="inputNama">
18   <button>Simpan</button>
19
20   <p>Masukan nama terlebih dahulu</p>
21 </body>
22 </html>
```


5. Kemudian kita perlu membuat sebuah function berisi baris kode di atas, agar fungsi ini dapat dipanggil ketika sebuah tombol diklik.

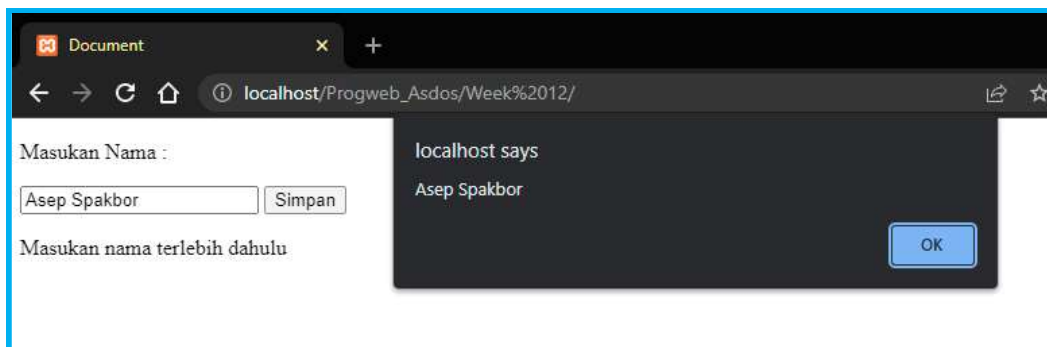


```
index.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,
7     initial-scale=1.0">
8   <title>Document</title>
9
10  <script>
11    function tampilkanNama(){
12      var fieldNama = document.getElementById("inputNama");
13      alert(fieldNama.value)
14    }
15  </script>
16 </head>
17 <body>
18   <p>Masukan Nama :</p>
19   <input type="text" id="inputNama">
20   <button>Simpan</button>
21
22   <p>Masukan nama terlebih dahulu</p>
23 </body>
24 </html>
```

6. Kemudian kita akan pasang fungsi ini pada sebuah tombol simpan.

```
index.php x
index.php > html > body > button
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,
7     initial-scale=1.0">
8   <title>Document</title>
9
10  <script>
11    function tampilkanNama(){
12      var fieldNama = document.getElementById("inputNama");
13      alert(fieldNama.value)
14    }
15  </script>
16 </head>
17 <body>
18   <p>Masukan Nama :</p>
19   <input type="text" id="inputNama">
20   <button onclick="tampilkanNama()">Simpan</button>
21
22   <p>Masukan nama terlebih dahulu</p>
23 </body>
24 </html>
```

7. Jika sudah maka kita akan lakukan pengujian. Coba masukan nama dan klik simpan. Maka nama akan ditampilkan di Window Alert.



8. Kita telah berhasil menampilkan nama dari inputan pengguna, sekarang kita akan mengubah string "Masukan nama terlebih dahulu" menjadi "Selamat Datang User, Asep Spakbor" ketika tombol simpan diklik. Pertama kita akan select tag <p> terlebih dahulu menggunakan by tag name 'p'. Lalu kita ubah string <p> seperti dibawah ini.

```
index.php X
index.php > html > head > script > tampilkanNama
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8
9   <script>
10     function tampilkanNama(){
11       var fieldNama = document.getElementById("inputNama");
12       alert(fieldNama.value)
13
14       var welcome = document.getElementsByTagName('p')
15       var namaUser = fieldNama.value
16       welcome[1].innerHTML = "Selamat Datang User, " + namaUser
17     }
18   </script>
19 </head>
20 <body>
21   <p>Masukan Nama :</p>
22   <input type="text" id="inputNama">
23   <button onclick="tampilkanNama()">Simpan</button>
24
25   <p>Masukan nama terlebih dahulu</p>
26 </body>
27 </html>
```

9. Seperti yang kita tahu karena get element by tag name mengembalikan list/array, maka kita perlu menambahkan indeks mana yang akan kita ubah. Karena element yang akan kita ubah berada di indeks ke 1 maka kita sesuaikan seperti pada baris kode diatas.

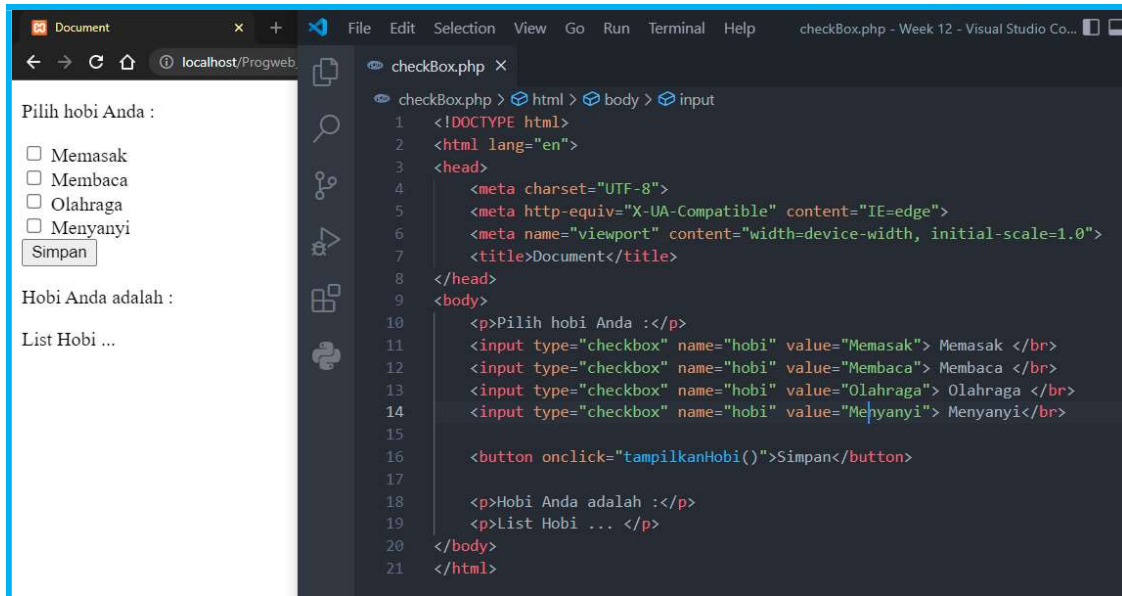
```
<body>
  <p>Masukan Nama :</p> => Indeks ke - 0
  <input type="text" id="inputNama">
  <button onclick="tampilkanNama()">Simpan</button>

  <p>Masukan nama terlebih dahulu</p> => Indeks ke - 1
</body>
</html>
```

F.2. Latihan Selector 2

Pada latihan kali ini kita akan membuat sebuah checkbox berisi hobi, dan kita akan menampilkan list hobi yang dipilih ke window alert. Dan menampilkan nya di element “Hobi Anda adalah :”.

1. Pertama kita buat terlebih dahulu halaman webiste seperti di bawah ini.



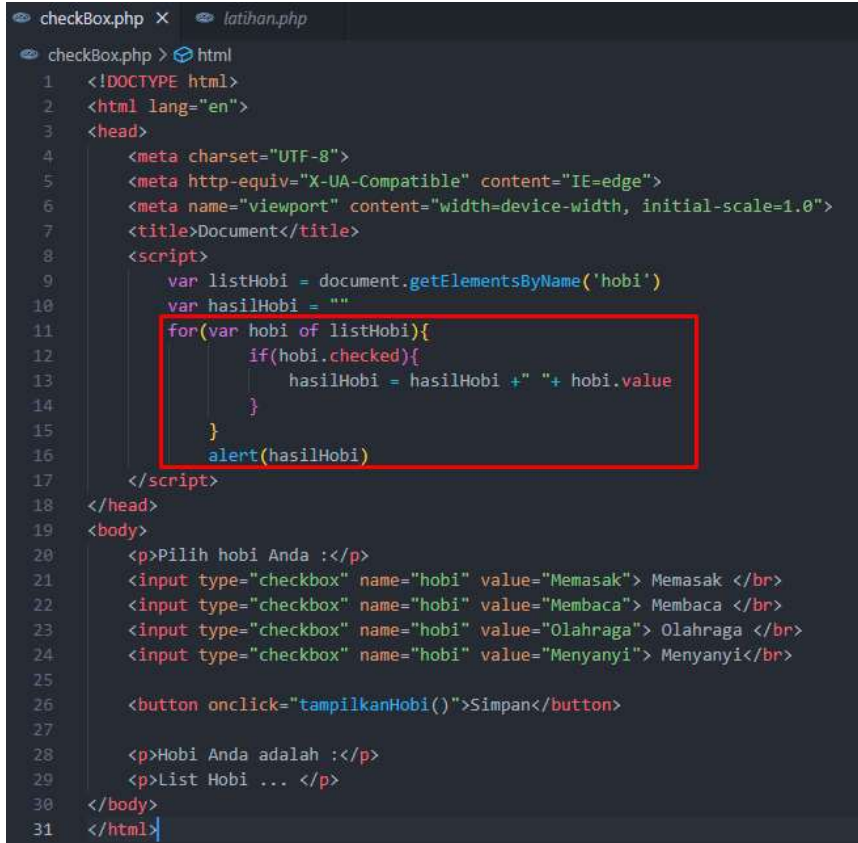
2. Kemudian kita akan melakukan select kepada checkbox menggunakan cara di bawah ini.

```
checkbox.php x latihan.php
checkbox.php > html > head > script
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <script>
9     var listHobi = document.getElementsByName('hobi')
10
11   </script>
12 </head>
13 <body>
14   <p>Pilih hobi Anda :</p>
15   <input type="checkbox" name="hobi" value="Memasak"> Memasak </br>
16   <input type="checkbox" name="hobi" value="Membaca"> Membaca </br>
17   <input type="checkbox" name="hobi" value="Olahraga"> Olahraga </br>
18   <input type="checkbox" name="hobi" value="Menyanyi"> Menyanyi</br>
19
20   <button onclick="tampilkanHobi()">Simpan</button>
21
22   <p>Hobi Anda adalah :</p>
23   <p>List Hobi ... </p>
24 </body>
25 </html>
```

3. Kemudian kita membuat sebuah variabel string untuk wadah list hobi yang di pilih pengguna.

```
checkbox.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <script>
9     var listHobi = document.getElementsByName('hobi')
10    var hasilHobi = ""
11
12   </script>
13 </head>
14 <body>
15   <p>Pilih hobi Anda :</p>
16   <input type="checkbox" name="hobi" value="Memasak"> Memasak </br>
17   <input type="checkbox" name="hobi" value="Membaca"> Membaca </br>
18   <input type="checkbox" name="hobi" value="Olahraga"> Olahraga </br>
19   <input type="checkbox" name="hobi" value="Menyanyi"> Menyanyi</br>
20
21   <button onclick="tampilkanHobi()">Simpan</button>
22
23   <p>Hobi Anda adalah :</p>
24   <p>List Hobi ... </p>
25 </body>
26 </html>
```

4. Kemudian kita akan membuat sebuah perulangan untuk mengecek apakah sebuah item di check/dipilih oleh pengguna atau tidak. Kemudian jika dicheck maka masukan kedalam var hasilHobi dan jika semua perulangan selesai, tampilkan hasilHobi di alert.



```
checkbox.php x latihan.php
checkbox.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <script>
9     var listHobi = document.getElementsByName('hobi')
10    var hasilHobi = ""
11    for(var hobi of listHobi){
12      if(hobi.checked){
13        hasilHobi = hasilHobi + " " + hobi.value
14      }
15    }
16    alert(hasilHobi)
17  </script>
18 </head>
19 <body>
20   <p>Pilih hobi Anda :</p>
21   <input type="checkbox" name="hobi" value="Memasak"> Memasak </br>
22   <input type="checkbox" name="hobi" value="Membaca"> Membaca </br>
23   <input type="checkbox" name="hobi" value="Olahraga"> Olahraga </br>
24   <input type="checkbox" name="hobi" value="Menyanyi"> Menyanyi</br>
25
26   <button onclick="tampilkanHobi()">Simpan</button>
27
28   <p>Hobi Anda adalah :</p>
29   <p>List Hobi ... </p>
30 </body>
31 </html>
```

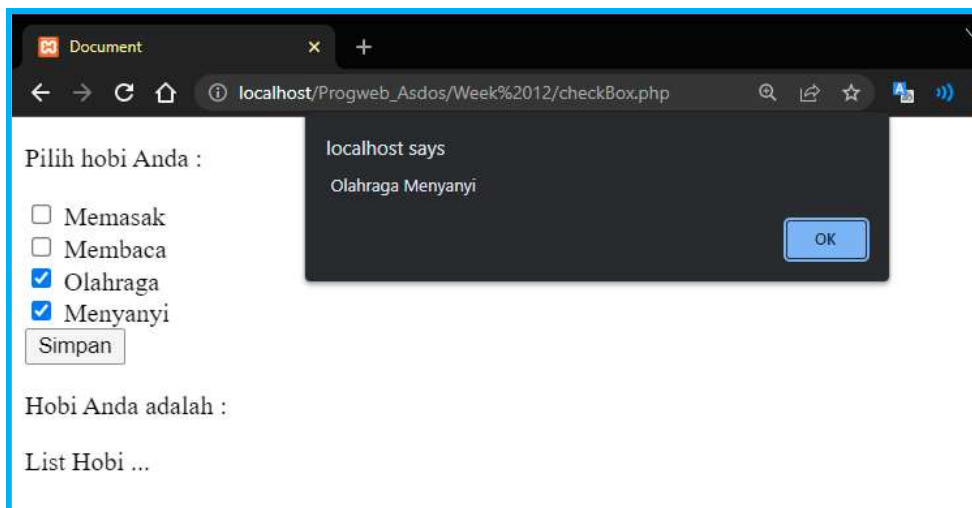
5. Selanjutnya karena kita akan menggunakan baris kode ini dalam sebuah tombol Simpan, maka kita pindahkan baris kode yang kita tulis ini ke dalam functions tampilkanHobi.


```

checkbox.php > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <script>
9          function tampilkanHobi(){
10             var listHobi = document.getElementsByName('hobi')
11             var hasilHobi = ""
12             for(var hobi of listHobi){
13                 if(hobi.checked){
14                     hasilHobi = hasilHobi + " " + hobi.value
15                 }
16             }
17             alert(hasilHobi)
18         }
19     }
20 </script>
21 </head>
22 <body>
23     <p>Pilih hobi Anda :</p>
24     <input type="checkbox" name="hobi" value="Memasak"> Memasak </br>
25     <input type="checkbox" name="hobi" value="Membaca"> Membaca </br>
26     <input type="checkbox" name="hobi" value="Olahraga"> Olahraga </br>
27     <input type="checkbox" name="hobi" value="Menyanyi"> Menyanyi</br>
28
29     <button onclick="tampilkanHobi()">Simpan</button>
30
31     <p>Hobi Anda adalah :</p>
32     <p>List Hobi ... </p>
33 </body>

```

6. Jika sudah maka kita uji coba memilih hobi. Jika berhasil maka akan terlihat seperti pada gambar di bawah ini.



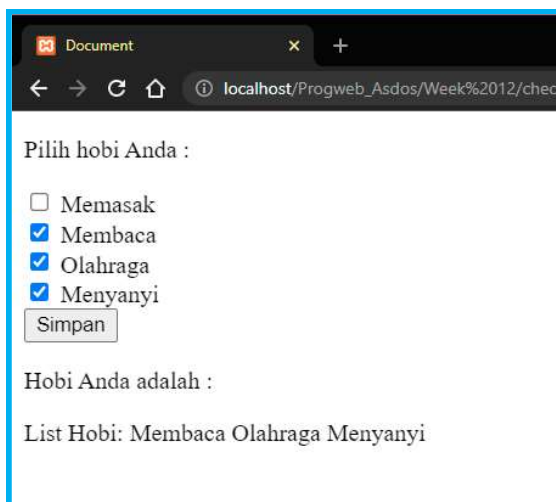
7. Sekarang tinggal kita masukan data list hobi kedalam "List Hobi ..."


```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <script>
9     function tampilkanHobi(){
10       var listHobi = document.getElementsByName('hobi')
11       var hasilHobi = ""
12       for(var hobi of listHobi){
13         if(hobi.checked){
14           hasilHobi = hasilHobi + " " + hobi.value
15         }
16       }
17       alert(hasilHobi)
18       var ringkasanHobi = document.getElementsByTagName('p')
19       ringkasanHobi[2].innerHTML = "List Hobi: " + hasilHobi
20     }
21   }
22 </script>
23 </head>
24 <body>
25   <p>Pilih hobi Anda :</p>
26   <input type="checkbox" name="hobi" value="Memasak"> Memasak </br>
27   <input type="checkbox" name="hobi" value="Membaca"> Membaca </br>
28   <input type="checkbox" name="hobi" value="Olahraga"> Olahraga </br>
29   <input type="checkbox" name="hobi" value="Menyanyi"> Menyanyi</br>
30   <button onclick="tampilkanHobi()">Simpan</button>
31
32
33

```

8. Jika berhasil maka akan terlihat seperti pada gambar di bawah ini.



F.3. Validasi Teros

Pertama-tama kita buat *form sign up* terlebih dahulu pada *file* dengan nama **form_signup.php** sebagai berikut :

```
<?php
    if (($_POST)) {
        var_dump($_POST);
    }
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Validasi Form Sign up</title>
</head>
<style>
    .information-d{
        color: red;
    }
</style>
<body>
    <h1>Form Sign Up Sederhana</h1>
    <form method="POST" action="form_signup.php">
        <table>
            <tr>
                <td><label for="nama">Nama </label></td>
                <td><input type="text" id="idNama" name="nama"> </td>
            </tr>
            <tr>
                <td><label for="email">Email </label></td>
                <td><input type="email" id="idEmail" name="email"> </td>
            </tr>
            <tr>
                <td><label for="password">Password </label></td>
                <td><input type="password" id="idPassword" name="password"> </td>
                <td><span class="information-d"></span></td>
            </tr>
            <tr>
                <td><label for="confirmPassword">Confirm Password </label></td>
                <td><input type="password" id="idConfirmPassword" name="confirmPassword"></td>
            </tr>
            <tr>
                <td></td>
                <td><input type="submit" name="submit" value="Submit"></input></td>
            </tr>
        </table>
        <br>
        <br>
    </form>
</body>
</html>
```

Form telah mampu mengirimkan segala data yang diisikan. Data yang dikirimkan akan diterima pada *file* `form_signup.php` yang kemudian akan ditampilkan. Tetapi belum ada proses validasi pada *input*, sehingga segala *input* akan langsung dikirim. Padahal terdapat beberapa validasi yang secara umum kita temui pada suatu *form sign up*, seperti :

- Nama, *Email*, *Password* harus diisi sebelum *form* dikirim
- *Password* dan *Confirm Password* harus sama
- *Password* wajib berisikan minimal 1 huruf besar, 1 huruf kecil dan 1 angka.

Tiga validasi di atas dapat dibuat menggunakan *Javascript* karena proses validasi akan dapat berjalan ketika *user* sedang mengisi *form*. Hal ini sering dijumpai pada saat mengisi *password*. Ketika mengisi *password* terkadang terdapat syarat yang belum dipenuhi sehingga muncul peringatan. Hal seperti itu mampu dilakukan jika menggunakan *Javascript*.

1. Sekarang akan dibuat validasi “Nama, Email, Password harus diisi sebelum form dikirim”. Terdapat 2 cara untuk melakukan validasi ini.

Cara pertama adalah sebagai berikut :

```
</body>

<script>
    // Variabel
    var inpName = document.getElementById('idNama')
    var inpEmail = document.getElementById('idEmail')
    var inpPassword = document.getElementById('idPassword')
    var inpConfirmPassword = document.getElementById('idConfirmPassword')

    // Validasi 1 ( Required || Pada form tidak ada required, akan ditambahkan dengan JS)
    inpName.required = true
    inpEmail.required = true
    inpPassword.required = true

</script>

</html>
```

Perlu diperhatikan pada *form* di awal tidak ada atribut ‘required’ pada setiap inputan. Sehingga pada cara ini akan ditambahkan atribut tersebut menggunakan *Javascript*. Cara ini akan memanggil setiap

elemen input menggunakan ID yang dimiliki pada setiap inputan. Setiap inputan yang wajib diisi dapat ditambahkan atribut 'Required'. Penambahan atribut tersebut membuat inputan harus diisi sebelum *form* bisa dikirim.

Cara kedua adalah sebagai berikut :

```
</body>

<script>
    // Variabel
    var inpName = document.getElementById('idNama')
    var inpEmail = document.getElementById('idEmail')
    var inpPassword = document.getElementById('idPassword')
    var inpConfirmPassword = document.getElementById('idConfirmPassword')

    // Validasi 1 ( onclick submit baru dicek )
    function validationRequired(){
        if (inpName.value=== "" || inpEmail.value=== "" || inpPassword.value=== "") {
            window.alert('Terdapat Kolom Inputan yang kosong !')
            event.preventDefault()
        }
    }
</script>

</html>
```

```
<tr>
    <td><label for="confirmPassword">Confirm Password </label></td>
    <td><input type="password" id="idConfirmPassword" name="confirmPassword"></td>
</tr>
<tr>
    <td></td>
    <td><input type="submit" name="submit" value="Submit" onclick="validationRequired()"></input></td>
</tr>
```

Cara ini berbeda dengan cara pertama, dimana inputan akan dicek apakah berisi atau tidak ketika tombol 'submit' diklik. Proses ini dapat terjadi karena terdapat :

- **function validationRequired()**

Fungsi ini berisi proses pengecekan *value* pada *inputan* (Nama, Email, dan Password) apakah kosong atau tidak. Pengaksesan nilai inputan dapat dilakukan dengan cara "elemenInputan.value".

Jika terdapat inputan yang kosong maka akan ditampilkan informasi menggunakan 'window.alert("isi pesan...")' dan terdapat "event.preventDefault()" yang pada kasus ini akan membatalkan *event* pengiriman *form*, karena terdapat inputan yang tidak terisi.

- **onclick**

Atribut ini memungkinkan untuk mengeksekusi suatu fungsi yang sebelumnya telah dibuat. Fungsi akan dieksekusi ketika suatu elemen diklik. Pada kasus ini tombol 'Submit' akan mengeksekusi fungsi 'validationRequired()' ketika tombol 'Submit' diklik.

2. Kemudian akan dibuat validasi "Password dan Confirm Password harus sama".

```
<script>
// Variabel
var inpName = document.getElementById('idNama')
var inpEmail = document.getElementById('idEmail')
var inpPassword = document.getElementById('idPassword')
var inpConfirmPassword = document.getElementById('idConfirmPassword')

// Validasi 1 ( onclick submit baru dicek )
function validationRequired(){
    if (inpName.value=== "" || inpEmail.value=== "" || inpPassword.value=== "") {
        window.alert('Terdapat Kolom Inputan yang kosong !')
        event.preventDefault()
    }else{
        // Validasi 2 ( onclick submit baru dicek kesamaan password dengan confirm password )
        if (inpPassword.value !== inpConfirmPassword.value) {
            window.alert('Password dan Confirm Password tidak sama !')
            event.preventDefault()
        }
    }
}
</script>
```

Pada cara ini akan dilakukan pengecekan apakah 'password' dan 'confirm password' sama. Proses pengecekan dilakukan ketika *form* akan di kirim, oleh karena itu proses pengecekan akan terjadi ketika tombol 'submit' diklik. Sehingga validasi ini dapat dimasukan ke dalam fungsi validationRequired(). Secara garis besar proses validasi ini membandingkan nilai inputan 'password' dengan 'confirm password' dengan mengakses 'value' pada setiap inputan dan membandingkannya. Jika hasil perbandingan adalah tidak sama, maka akan ditampilkan informasi menggunakan 'window.alert("isi pesan...")' serta terdapat "event.preventDefault()" yang pada kasus ini akan membatalkan *event* pengiriman *Form*, karena 'password' tidak sama dengan 'confirm password'.

3. Kemudian akan dibuat validasi “Password wajib berisikan minimal 1 huruf besar, 1 huruf kecil dan 1 angka”.

```
<tr>
  <td><label for="password">Password </label></td>
  <td><input type="password" id="idPassword" name="password" onkeyup="validasiPassword()"> </td>
  <td><span class="information-d"></span></td>
</tr>
```

```
// Validasi 3 (Password harus memiliki 1 huruf besar, 1 angka, 1 huruf kecil)
function validasiPassword(){
  let areaPeringatan = document.getElementsByClassName('information-d')[0]
  let regularExpression = /^(?=.*[0-9])(?=.*[A-Z])(?=.*[a-z])[a-zA-Z0-9]{3,16}$/;
  if(!regularExpression.test(inpPassword.value)) {
    areaPeringatan.textContent='Password mengandung 1 huruf besar, 1 huruf kecil dan 1 angka';
  }else{
    areaPeringatan.textContent='Okay';
  }
}
</script>
```

Validasi ini bekerja dengan cara melakukan pengecekan setiap kali *user* memberikan nilai terhadap *password*. Pengecekan akan menggunakan ReGex. Ketika ‘password’ yang diberikan tidak memenuhi syarat maka akan terdapat pesan berwarna merah seperti berikut :

Form Sign Up Sederhana

Nama	<input type="text"/>	
Email	<input type="text"/>	
Password	<input type="password"/>	Password mengandung 1 huruf besar, 1 huruf kecil dan 1 angka
Confirm Password	<input type="password"/>	
<input type="button" value="Submit"/>		

Seluruh proses validasi ini dapat berjalan karena terdapat :

- **function validasiPassword()**

Pada proses ini memanggil `` yang menampung pesan berwarna merah dengan menggunakan nama *class* nya. Kemudian dibuat ReGex yang mampu mewajibkan terdapat 1 huruf besar, 1 huruf kecil dan 1 angka. ReGex tersebut dapat diujikan terhadap ‘password’ yang diberikan menggunakan ‘.test()’. Jika pengujian ReGex gagal maka `` yang

sebelumnya telah dipanggil akan diatur isi teksnya menjadi “Password mengandung 1 huruf besar, 1 huruf kecil dan 1 angka”. Tetapi jika pengujian ReGex berhasil maka akan diubah menjadi “Okay”.

- **Onkeyup**

Atribut ‘onkeyup’ pada *input* ‘password’ untuk melakukan pengecekan setiap kali suatu *key* terangkat. Oleh karena itu proses pengecekan apakah suatu ‘password’ telah memenuhi syarat akan terjadi secara dinamis ketika *user* memasukan *password*. Proses pengecekan akan menggunakan fungsi bernama `validasiPassword()`.

F.4. Kasta Copier

Pada kasus ini, kita akan membuat satu halaman dengan 2 input text field yang salah satunya akan **menduplikasi** value. Sebagai contoh, kita memiliki input text field A dan B. Ketika kita **mengetik (onkeyup)** di input text field A, maka value / text dari A akan **diduplikasi / dicopy** ke input text field B.

JS#2 Copier

Input

Copier

Berikut ini langkah - langkah untuk membuatnya:

1. Buatlah file html beserta struktur htmlnya. Lalu tambahkan 2 input text pada body & berikan id pada masing - masing input. ID tersebut akan digunakan sebagai selector untuk javascript.


```

<h1>JS#2 Copier</h1>

<div>
  <h2>Input</h2>
  <input type="text" id="input">
</div>

<div>
  <h2>Copier</h2>
  <input type="text" id="copier">
</div>

```

2. Buatlah tag script setelah tag body. Lalu kita buat sebuah function untuk melakukan copy value. Function ini akan dipanggil setiap kita mengetik (on key up).

```

<script>
  function copier() {
    document.getElementById("copier").value =
    document.getElementById("real").value;
  }
</script>

```

Mari kita breakdown satu per satu setiap fungsi dari function tersebut:

```
document.getElementById("copier").value =
```

Statement tersebut akan **mengambil elemen dengan id copier** yaitu input text B yang value / text nya akan diisi oleh input A. Lalu untuk mengisi value dari sebuah elemen input, kita dapat menggunakan **.value** = .

```
document.getElementById("real").value;
```

Statement tersebut akan mengambil **elemen dengan id real** yaitu input text A dan kita ambil value nya dengan **.value** .

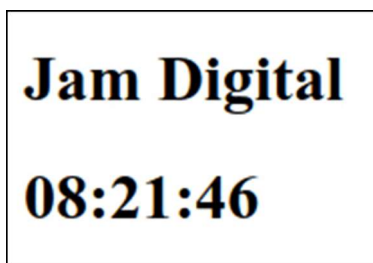
Sehingga dapat kita simpulkan bahwa **.value** dapat digunakan untuk **mengambil** sekaligus **mengisi** value dari input field.

- Langkah terakhir yaitu memanggil function `copier()` tersebut ketika sedang mengetik. Caranya adalah dengan **menambahkan attribute onkeyup** pada tag input A dan kita isi dengan **function copier()**.

```
<div>
  <h3>Input</h3>
  <input type="text" name="real" onkeyup="copier()" id="real">
</div>
```

F.5. Dikasih Waktu

Kita akan membuat jam digital yang memiliki format HH:MM:SS (23:59:59). Seperti jam digital pada umumnya, jam ini **akan selalu terupdate** setiap 1 detik.



Berikut ini langkah - langkah untuk membuatnya:

- Buatlah file html beserta struktur htmlnya. Lalu kita buat **struktur untuk jam digital** menggunakan **span** disertai dengan **id** untuk selector pada javascript.

```
<h1>Jam Digital</h1>
<h2>
  <span id="jam">00</span>:<span id="menit">00</span>:<span id="detik">00</span>
</h2>
```

Jika sudah maka hasil sementara akan seperti ini



- Buatlah file javascript dengan nama `jam_digital.js`. Lalu kita buat sebuah function bernama `setTime()`.

```

JS jam_digital.js > setTime
1 function setTime() {
2
3 }

```

Pada function tersebut kita harus mengambil waktu saat ini, lalu mengambil setiap jam, menit, dan detik. Berikut ini caranya

```

function setTime() {
    const date = new Date();
    let hour = date.getHours();
    let minute = date.getMinutes();
    let sec = date.getSeconds();
}

```

- Setelah itu, kita akan menambahkan jam, menit, dan detik ke dalam struktur html yang sudah kita buat. Kita dapat memanfaatkan id sebagai selector. Lalu mengisi text nya menggunakan innerHTML. Berikut ini caranya

```

document.getElementById("jam").innerHTML = hour;
document.getElementById("menit").innerHTML = minute;
document.getElementById("detik").innerHTML = sec;

```

- Jika sudah, kita dapat mencoba fungsi kita dengan menambahkan jam_digital.js diakhir body html.

```

17 </body>
18
19 <script src="jam_digital.js"></script>
20
21 </html>

```

- Lalu panggil method setTime() di dalam file javascript yang sudah kita buat.

```

1 > function setTime() { ...
10 }
11
12 setTime();

```

- Maka tampilan jam digital kita akan seperti berikut

Jam Digital

9:30:23

7. Namun Jam Digital yang kita buat **masih statis** atau **belum bisa melakukan update** setiap detiknya. Maka kita akan lanjutkan untuk membuat jam digital ini menjadi dinamis.

Pada function setTime(), kita tambahkan **code untuk melakukan delay** waktu selama **1 detik**. Delay ini berfungsi akan proses **update tidak terlalu cepat**. Ada sebuah function bawaan dari javascript untuk melakukan delay yaitu **setTimeout()**. Berikut ini cara penggunaannya.

8. Tambahkan setTimeout di dalam function setTime()

```
function setTime() {  
    setTimeout(function () {  
  
    }, 1000);  
}
```

Semua code yang ada di dalam block function() dari **setTimeout** akan di delay selama 1 detik sehingga masukkan code yang sebelumnya kita buat ke dalamnya. Maka secara function setTime() secara keseluruhan akan sebagai berikut

```
function setTime() {  
    setTimeout(function () {  
        const date = new Date();  
        let hour = date.getHours();  
        let minute = date.getMinutes();  
        let sec = date.getSeconds();  
  
        document.getElementById("jam").innerHTML = hour;  
        document.getElementById("menit").innerHTML = minute;  
        document.getElementById("detik").innerHTML = sec;  
    }, 1000);  
}
```

9. Setelah kita menambahkan delay, maka kita akan **melakukan refresh** untuk jam digital kita. Caranya cukup mudah yaitu dengan memanfaatkan **fungsi rekursif** yaitu memanggil function setTime(). Pemanggilan setTime() ini akan **dilakukan setelah waktu sudah ditampilkan** pada struktur html (innerHTML).

```
document.getElementById("jam").innerHTML = hour;  
document.getElementById("menit").innerHTML = minute;  
document.getElementById("detik").innerHTML = sec;
```

```
setTime());
```

10. Jam digital kita sudah berhasil berjalan dengan baik. Namun ada 1 hal yang masih belum tepat yaitu **format jam**. Agar jam digital kita lebih rapi, kita dapat menambahkan baris code berikut ini yang berfungsi untuk **menambahkan angka 0** jika angka pada jam **hanya 1 digit**.

```
6      let sec = date.getSeconds();  
7  
8      if (hour < 10) {  
9          hour = "0" + hour;  
10     }  
11  
12     if (minute < 10) {  
13         minute = "0" + minute;  
14     }  
15  
16     if (sec < 10) {  
17         sec = "0" + sec;  
18     }  
19  
20     document.getElementById("jam").innerHTML = hour;  
21     document.getElementById("menit").innerHTML = minute;  
22     document.getElementById("detik").innerHTML = sec;
```