

## MODUL 5 PENGENALAN JAVASCRIPT

---

### 5.1. Deskripsi Singkat

Selain HTML dan CSS, JavaScript adalah salah satu dari 3 bahasa yang harus dipelajari semua pengembang web. HTML digunakan untuk mendefinisikan konten halaman web. CSS digunakan untuk mengatur tampilan dan menentukan tata letak konten pada halaman web. Adapun Javascript digunakan untuk memprogram perilaku halaman web dan untuk membuat konten yang dinamis pada suatu dokumen HTML.

### 5.2. Tujuan Praktikum

Setelah menyelesaikan praktikum pada modul ini, mahasiswa diharapkan dapat memahami konsep dasar javascript dengan membuat beberapa dokumen HTML dan menambahkan JavaScript di dalamnya.

### 5.3. Material Praktikum

Praktikum ini memerlukan aplikasi text editor seperti Notepad (bawaan jika anda menggunakan Microsoft Windows). Anda juga bisa mengunduh aplikasi text editor lainnya seperti Notepad++ atau Visual Studio Code. Jangan menggunakan rich-text editor seperti Microsoft Word atau sejenisnya. Selain itu, praktikum ini juga memerlukan web browser seperti Internet Explorer atau Google Chrome. Saat ini, anda belum memerlukan web server untuk membuat halaman web.

### 5.4. Kegiatan Praktikum

#### 5.4.1. Deklarasi Variabel

Mari kita mulai dengan membuat halaman web "Hello World!" dengan JavaScript.

1. Buka text editor lalu masukkan markup HTML dan kode JavaScript berikut ini.

```
<!DOCTYPE html>
<html lang="en-GB">

<head>
  <title>Hello World</title>
</head>

<body>
  <script id="s1"> // Script with id s1
  </script>
  <p>Our first JavaScript script</p>
  <script id="s2"> // Script with id s2
    user = "<your name>";
    document.writeln("<p><b>Hello " + user + "<br>\nHello
World!</b></p>");
  </script>
```

```

<script id="s3"> // Script with id s3
</script>
<p>This is the rest of the page</p>
<noscript>JavaScript not supported or disabled</noscript>
</body>

</html>

```

Tuliskan nama anda di *<your name>*. Perintah `document.writeln` menuliskan suatu string/teks ke dalam dokumen HTML. Perhatikan ada tiga blok kode JavaScript dalam dokumen ini, ditandai dengan elemen script dengan id s1, s2, dan s3. Ini akan kita lengkapi pada latihan-latihan berikut.

2. Simpan markup HTML dan kode JavaScript di atas dalam file bernama **jsDemo06A.html**.
3. Buka file jsDemo06A.html dengan web browser. Periksa dan pastikan tampilannya sesuai dengan yang diharapkan termasuk hasil eksekusi kode JavaScript. Pastikan bukan kode JavaScript yang ditampilkan di browser, melainkan hasil eksekusinya yang ditampilkan dalam web browser.
4. Jika tampilannya tidak sesuai harapan, khususnya pada hasil eksekusi kode JavaScript, anda bisa men-debug kode anda. Untuk itu, gunakan fitur 'console' atau 'JavaScript console' pada web browser, pada menu *Developer Tools*. Hampir semua web browser menyediakan fitur ini. Coba buka console pada browser, lalu muat kembali file jsDemo06A.html.
5. Untuk melihat bagaimana cara kerja console, tambahkan kode JavaScript berikut pada elemen script dengan id s2.

```

document.writeln("(1) The value of userAge is: " +
    userAge + "<br>");
document.writeln("(2) This statement is executed<br>");

```

Simpan file tersebut dan muat ulang browser. Pastikan bahwa tampilan tidak berubah dari sebelumnya, dan sepertinya, tidak ada efek dari skrip di atas. Jika anda lihat pada JavaScript console, anda akan melihat pesan kesalahan, bahwa variabel `userAge` belum dideklarasikan dan/atau diinisiasi.

6. Betulkan skrip sebelumnya dengan menambahkan deklarasi dan inisiasi terhadap variabel `userAge`. Pertama, tambahkan skrip berikut setelah skrip pada butir 5 di atas, yaitu setelah perintah `document.writeln` yang menggunakan variabel `userAge`.

```

var userAge = "27";
document.writeln("(3) The value of userAge is: " +
    userAge + "<br>");

```

Simpan kembali file dan muat ulang browser. Seharusnya sekarang tidak ada lagi pesan error pada console. Tampilan di web browser seharusnya seperti berikut ini.

```
(1) The value of userAge is: undefined  
(2) This statement is executed  
(3) The value of userAge is: 27
```

7. Pindahkan deklarasi berikut

```
var userAge = "27";
```

dari elemen script dengan id s2 ke elemen script dengan id s3.

Simpan kembali file. Bersihkan layar console. Kemudian muat kembali ke dalam web browser.

Apakah kode JavaScript sekarang berjalan dengan tanpa error, atau ada error? Apa kesimpulan yang bisa diambil dalam kaitannya antara deklarasi variabel dengan cakupan variabel?

8. Pindahkan kembali deklarasi berikut

```
var userAge = "27";
```

dari elemen script dengan id s3 ke elemen script dengan id s1.

Simpan kembali file. Bersihkan layar console. Kemudian muat kembali ke dalam web browser.

Apakah kode JavaScript sekarang berjalan dengan tanpa error, atau ada error? Apakah hasilnya sama dengan butir 7? Kalau tidak, mengapa?

Apa kesimpulan yang bisa diambil dalam kaitannya antara deklarasi variabel dengan cakupan variabel?

#### 5.4.2. Error Handling

Pada contoh sebelumnya, yaitu pada **jsDemo06A.html**, dapat kita lihat bahwa saat terjadi error, JavaScript berhenti dieksekusi namun tidak ada pesan kesalahan yang ditampilkan di layar browser. Saat menjalankan kode JavaScript, berbagai kesalahan dapat terjadi. Kesalahan atau error dapat berupa kesalahan coding yang dilakukan oleh programmer, kesalahan karena input dari user yang salah, dan hal-hal lain yang tidak terduga. Kita dapat menangani error pada kode Javascript menggunakan statement **try-catch**, sehingga saat terjadi kesalahan atau error, kita dapat menampilkan pesan kesalahannya kepada user, menjalankan script alternatif untuk menangani error tersebut, atau mengatur penanganan-penanganan lainnya.

Untuk memahami cara kerja **try-catch**, kita gunakan kembali potongan program yang serupa dengan **jsDemo06A.html**.

1. Salin ulang potongan program di bawah ini, lalu simpan dengan nama **jsDemo06B.html**.

```
<!DOCTYPE html>  
<html lang="en-GB">
```

```

<head>
  <title>Hello World</title>
</head>

<body>
  <script id="s1"> // Script with id s1
  </script>
  <p>Our first JavaScript script</p>
  <script id="s2"> // Script with id s2
    user = "<your name>";
    document.writeln("<p><b>Hello " + user + "<br>\nHello
World!</b></p>");
    document.writeln("(1) The value of userAge is: "+userAge +
"<br>");
    document.writeln("(2) This statement is executed<br>");

  </script>
  <script id="s3"> // Script with id s3
  </script>
  <p>This is the rest of the page</p>
  <noscript>JavaScript not supported or disabled</noscript>
</body>

</html>

```

2. Kita akan menggunakan try-catch untuk menampilkan pesan error di layer browser. Tambahkan container untuk menampilkan pesan error tersebut. Sebagai contoh, kita akan menggunakan elemen paragraf. Tambahkan script html berikut setelah tag <body>.

```

<p id="alert" style="color: red;"></p>

```

3. Modifikasi baris script yang menyebabkan error, yaitu script yang mencoba menampilkan variabel userAge. Tambahkan statement try catch pada baris tersebut menjadi seperti ini:

```

try {
  document.writeln("(1) The value of userAge is: " + userAge +
"<br>");
} catch (error) {
  document.getElementById("alert").innerHTML = error.message;
}

```

Sekarang seharusnya pesan error sudah tampil pada layer browser Anda. Dengan cara seperti ini, kesalahan atau error yang terjadi pada script dapat terinfokan dengan baik kepada user. Anda dapat memodifikasi statement catch dengan penanganan-penanganan error lainnya.

### 5.4.3. Sifat Bilangan Pecahan

Presisi dari bilangan pecahan 64-bit sangat terbatas. Permasalahan ini dikarenakan representasi bilangan biner untuk bilangan pecahan ini memungkinkan terjadinya pecahan-pecahan biner, seperti pada  $1/8$  dan  $1/128$ , namun tidak pada bilangan pecahan desimal seperti  $1/10$  dan  $1/100$  yang biasa digunakan sehari-hari. Latihan berikut ini akan menunjukkan hal tersebut.

1. Buka text editor lalu masukkan markup HTML dan kode JavaScript berikut ini.

```
<!DOCTYPE html>
<html lang="en-GB">

<head>
  <title>Math Example 2</title>
</head>

<body>
  <script>
    var x = 0.3 - 0.2
    var y = 0.2 - 0.1
    document.writeln("x = " + x + "<br>")
    document.writeln("y = " + y + "<br>")
    document.writeln("(x == y) = " + (x == y) + "<br>")
    document.writeln("(x == 0.1) = " + (x == 0.1) + "<br>")
    document.writeln("(y == 0.1) = " + (y == 0.1) + "<br>")
  </script>
</body>

</html>
```

Simpan dalam file **jsDemo06C.html**.

2. Buka file jsDemo06C.html dengan web browser.
3. Amati hasil eksekusi kode JavaScript yang ditampilkan di web browser. Mengapa terdapat perbedaan pada hasil operasi  $0.3 - 0.2$  dan  $0.2 - 0.1$ ?
4. Gunakan converter online untuk bilangan desimal menjadi bilangan pecahan seperti

[http://www.binaryconvert.com/convert\\_double.html](http://www.binaryconvert.com/convert_double.html)

<http://www.exploringbinary.com/floating-point-converter/>

untuk memahami bagaimana representasi bilangan pecahan 0.3, 0.2, 0.1 di dalam bilangan biner/komputer pada 64-bit floating point.

5. Tambahkan kode JavaScript berikut ke dalam file.

```
x = 1.275
y = 1.27499999999999991118
document.writeln(x + " and " + y + " are " +
  ((x == y) ? "equal" : "not equal") + "<br>")
```

6. Simpan file lalu buka/muat kembali ke dalam web browser. Output pada baris terakhir seperti ini.

1.275 and 1.275 are equal

Walaupun pernyataan di atas benar, namun ada yang aneh terjadi pada nilai dari variabel y. Apa yang terjadi?

#### 5.4.4. Operasi Matematika

Mari kita lihat lebih lanjut bagaimana operasi matematika dalam JavaScript. Sekaligus, kita akan mencoba untuk lebih baik memisahkan kode JavaScript dari markup HTML.

1. Buka text editor lalu masukkan markup HTML berikut ini lalu simpan dalam file **jsDemo07A.html**.

```
<!DOCTYPE html>
<html lang="en-GB">

<head>
  <title>JavaScript 07A</title>
</head>

<body>
  <p>Our third JavaScript script</p>
  <script src="jsDemo07A.js"></script>
</body>

</html>
```

Kemudian, dalam file terpisah, masukkan kode JavaScript berikut ini lalu simpan dalam file jsDemo07A.js. Perhatikan ekstensi filenya adalah **.js**.

```
for (var ii = 4, jj = 3; jj >= 0; ii++, jj--) {
  document.writeln(ii + " * " + jj + " = " + ii * jj + "<br>")
  document.writeln(ii + " / " + jj + " = " + ii / jj + "<br>")
  document.writeln("log(" + jj + ") = " +
    Math.log(jj) + "<br>")
  document.writeln("sqrt(" + (jj - 1) + ") = " +
    Math.sqrt(jj - 1) + "<br><br>")
}
```

2. Pastikan kedua file di atas telah tersimpan dan ada di direktori yang sama.
3. Buka file jsDemo07A.html dalam web browser lalu amati hasil eksekusi kode JavaScript di atas. Pastikan anda memahami bagaimana hasil eksekusi seperti NaN, Infinity dan -Infinity bisa keluar.
4. Empat baris pertama dari output adalah sebagai berikut:

```
4 * 3 = 12
4 / 3 = 1.3333333333333333
log(3) = 1.0986122886681098
```

```
sqrt(2) = 1.4142135623730951
```

Seperti yang anda lihat, hasil dari operasi matematika di atas menghasilkan bilangan desimal yang banyak. Ubah kode JavaScript agar menampilkan hanya tiga digit di belakang koma, seperti berikut.

```
4 * 3 = 12.000  
4 / 3 = 1.333  
log(3) = 1.099  
sqrt(2) = 1.414
```

Kisi-kisi: Cari di Internet fungsi apa dalam JavaScript untuk membatasi jumlah digit di belakang koma pada **konversi dari bilangan menjadi string**, bukan pembulatan.

5. Kode JavaScript yang anda buat bisa diakses dan dibaca oleh semua user yang bisa mengakses melalui web browser. Kalau ingin membuat kode JavaScript sedikit lebih sulit untuk dibaca, disalin atau diubah, anda bisa membuatnya menjadi sedikit **obscure/tersembunyi**, misalnya dengan tool **uglifyjs**. Hasil program **uglifyjs** terhadap **jsDemo07A.js** seperti berikut.

```
for(var ii=4,jj=3;0<=jj;ii++,jj--)document.writeln(ii+" * "+jj+" =  
"+ii*jj+"<br>"),document.writeln(ii+" / "+jj+" =  
"+ii/jj+"<br>"),document.writeln("log("+jj+") =  
"+Math.log(jj)+"<br>"),document.writeln("sqrt("+jj-1+") =  
"+Math.sqrt(jj-1)+"<br><br>");y=x=1.275,document.writeln(x+" and  
"+y+" are "+(x==y?"equal":"not equal")+"<br>");
```

Bisa dilihat, walaupun hasil eksekusi kode JavaScript ini masih sama, namun penulisan kode-nya menjadi sedikit sulit untuk dipahami.

#### 5.4.5. Operasi Logika

Pada latihan berikutnya, kita akan melihat bagaimana cara kerja tipe, persamaan dan perbandingan dalam JavaScript.

1. Buka text editor lalu masukkan markup HTML berikut ini lalu simpan dalam file **jsDemo07C.html**.

```
<!DOCTYPE html>  
<html lang="en-GB">  
  
<head>  
  <title>JavaScript 07C</title>  
</head>  
  
<body>  
  <script>  
    Array.prototype.toString = function () {  
      var arrStr = "[";  
      for (var i = 0; i < this.length; i++) {  
        arrStr += this[i];  
        if (i < this.length - 1) arrStr += ", "  
      }  
    }  
  }  
</script>  
</body>  
</html>
```

```

        return (arrStr += "]);
    }
    var values = [0, 123, 1.23e2, "12.3e1", true, false, NaN,
        Infinity, undefined, null, [0]];
    for (var i = 0; i < values.length; i++) {
        document.writeln(i + ") The type of " + values[i] +
            " is: " + typeof (values[i]) + "<br>");
    }
    document.writeln(i + ") The type of " + values + " is: " +
        typeof (values) + "<br>");
</script>
<noscript>JavaScript not supported or disabled</noscript>
</body>

</html>

```

2. Buka file jsDemo07C.html menggunakan web browser dan amati hasil eksekusi kode JavaScript. Pastikan anda memahami mengapa tiap elemen array ditampilkan seperti itu.
3. Kembangkan kode di atas dengan menambahkan kode berikut:

```
document.writeln("Elements of values are: " + values.join()+ "<br>")
```

Hasil dari kode di atas adalah sebagai berikut.

```
Elements          of          values          are:
0,123,123,12.3e1,true,false,NaN,Infinity,,,[0]
```

Mengapa 'undefined' dan 'null' ditampilkan menjadi string kosong?

Lihat <https://262.ecma-international.org/9.0/>, Bab 22.1.3.13, untuk memahami cara kerja fungsi join.

4. Kembangkan kode JavaScript di atas dengan menambahkan kode program yang membandingkan tiap elemen dengan tiap elemen yang lain menggunakan empat operator, yaitu ==, ===, > dan <.

Sembilan baris pertama dari output kode program yang baru seperti berikut ini:

```

(number)0 == (number)0: true
(number)0 === (number)0: true
(number)0 < (number)0: false
(number)0 > (number)0: false

(number)0 == (number)123: false
(number)0 === (number)123: false
(number)0 < (number)123: true
(number)0 > (number)123: false

```



5. Ubah kode pada latihan sebelumnya pada butir d agar hasilnya ditampilkan dalam bentuk tabel menggunakan elemen tabel HTML. Misalnya potongan output untuk sembilan baris di atas ditampilkan seperti ini:

(number)0 == (number)0: true	(number)0 === (number)0: true	(number)0 < (number)0: false	(number)0 > (number)0: false
(number)0 == (number)123: false	(number)0 === (number)123: false	(number)0 < (number)123: true	(number)0 > (number)123: false

### 5.5. Penugasan

Screenshot potongan program kegiatan praktikum yang telah Anda kerjakan pada kegiatan (jsDemo06A.html, jsDemo06B.html, jsDemo06C.html, jsDemo07A.html, jsDemo07A.js, jsDemo07C.html) beserta screenshot outputnya masing-masing pada file word. Kemudian simpan dan kumpulkan dalam format pdf.