

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. Pada praktikum kali ini, anda akan mencoba menggunakan Javascript untuk membuat konten yang dinamis pada suatu dokumen HTML.

5.2. Tujuan Praktikum

Setelah menyelesaikan praktikum pada modul ini, mahasiswa diharapkan dapat membuat beberapa dokumen HTML dan menambahkan JavaScript untuk membuat konten yang dinamis.

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 coba 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 tampilannya sesuai dengan yang diharapkan termasuk hasil eksekusi kode JavaScript. Pastikan bukan kode JavaScript 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. Hampir semua web browser menyediakan fitur ini. Coba buka console ini, 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, kemudian muat kembali ke dalam web browser sambil menampilkan JavaScript console. Tampilannya sama dengan sebelumnya, namun di console, seharusnya muncul pesan error yang menunjukkan bahwa variabel `userAge` belum didefinisikan/dideklarasikan.

6. Untuk memperbaiki kesalahan di atas, perbaiki kode JavaScript dengan menambahkan deklarasi variabel `userAge`. Pertama, tambahkan kode JavaScript berikut setelah kode yang ditambahkan pada butir e di atas.

```

var userAge = "27";

```

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

Simpan kembali file. Bersihkan layar console. Kemudian muat kembali ke dalam web browser. Seharusnya sekarang tidak ada lagi pesan error. 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
```

Bisa kita lihat di sini bahwa deklarasi variabel tidak harus mendahului penggunaannya untuk menghindari error; walaupun ini tidak termasuk inisialisasinya yang akan dieksekusi kemudian.

Namun ini hanya berlaku dalam konteks tertentu dalam kode JavaScript, yang dalam contoh ini yang diapit dengan elemen script. Kita akan lihat tentang ini setelah ini.

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? Kesimpulan apa yang dapat diambil berkaitan dengan scope/cakupan deklarasi 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 f? Kalau tidak, mengapa?

9. Kita perlu mempertimbangkan satu kemungkinan lagi tentang bagaimana scope/cakupan deklarasi variabel. Dalam banyak bahasa pemrograman, cakupan variabel hanya efektif di dalam blok di mana variabel tersebut dideklarasikan. Misalnya, variabel yang dideklarasikan dalam blok if-then hanya berlaku dalam blok itu. Variabel tersebut tidak efektif dalam blok else

walaupun dalam blok if yang sama. Mari kita lihat, apakah hal ini berlaku juga pada JavaScript.

Tambahkan kode berikut tepat sebelum elemen noscript.

```
<!-- Exercise 1j -->
<script id="s4">
  if (false) {
    var x = 519;
  } else {
    document.writeln("The value of x is " + x + "<br>");
  }
</script>
```

Simpan kembali file. Bersihkan layar console. Kemudian muat kembali ke dalam web browser. Seharusnya sekarang tidak ada lagi pesan error. Tampilan di web browser seharusnya seperti berikut ini.

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

Bagaimana output dalam browser dan dalam console? Apakah ada error?

10. Dalam elemen skrip dengan id s4 yang baru saja kita tambahkan, ubah kode berikut

```
var x = 519;
```

menjadi

```
var y = 519;
```

Simpan kembali file. Bersihkan layar console. Kemudian muat kembali ke dalam web browser. Bagaimana output dalam browser dan dalam console? Apakah ada error? Apa yang ingin ditunjukkan dalam latihan butir i dan j di atas berkaitan dengan scope/cakupan deklarasi variabel dalam JavaScript? Apa yang ingin ditunjukkan dalam latihan butir i dan j di atas berkaitan dengan eksekusi kode JavaScript? Jika ada error pada eksekusi latihan butir i dan j di atas, coba perbaiki sekarang dan ulang kembali.

5.4.2. 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. Gunakan converter online untuk bilangan desimal menjadi bilangan pecahan seperti

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, dan hasil dari operasi $0.3 - 0.2$ dan $0.2 - 0.1$.

4. 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>")

```

5. Simpan file jsDemo07A.js lalu buka/muat kembali file jsDemo07A.html 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?

5.4.3. Operasi Matematika

1. Mari kita lihat lebih lanjut bagaimana operasi matematika dalam JavaScript. Sekaligus, kita akan mencoba untuk lebih baik memisahkan kode JavaScript dari markup HTML.
 - a. 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>")
}
```

- b. Pastikan kedua file di atas telah tersimpan dan ada di direktori yang sama.
 - c. 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.
 - d. 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.

- e. 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+" ) = "+Math.sqrt(jj)+"<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.

2. Pada latihan berikutnya, kita akan melihat bagaimana cara kerja tipe, persamaan dan perbandingan dalam JavaScript.
 - a. 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 += ", "
    }
    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>

```

- b. Buka file jsDemo07C.html menggunakan web browser dan amati hasil eksekusi kode JavaScript. Pastikan anda memahami mengapa tiap elemen array ditampilkan seperti itu.
- c. 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]

```

Anda bisa melihat, di mana kita mengharapkan muncul 'undefined' dan 'null', di sana muncul string kosong. Walaupun, ini dikarenakan join merupakan fungsi penggabungan elemen array yang direpresentasikan dalam string, dan representasi dari undefined dan null dalam string adalah masing-masing 'undefined' dan 'null'.

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

- d. 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
```

- e. Ubah kode pada latihan sebelumnya pada butir d agar hasilnya ditampilkan dalam bentuk tabel menggunakan elemen tabel HTML. Misalnya sembilan output 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. Responsi

Untuk responsi, salin potongan program kegiatan praktikum yang telah Anda kerjakan beserta screenshot outputnya masing-masing pada file word, lalu kumpulkan dalam format pdf.