



JAVASCRIPT

- Introduction -

Pertemuan VIII

Pemrograman Web Dasar
Semester 1

Pengenalan Javascript

- Asal mula nama JavaScript adalah **LiveScript**, dikembangkan pertama kali pada tahun 1995 di Netscape Communications.
- Hasil kolaborasi antara **Netscape** dan **Sun** (pengembang bahasa pemrograman “*Java*”) memberikan nama baru “*JavaScript*” pada tanggal 4 desember 1995.
- Bahasa ini dikenali pada browser Netscape Navigator mulai versi di atas 2.0., sedangkan Microsoft melengkapi Internet Explorer dengan JavaScript mulai versi 3.0 ke atas.

Pengenalan Javascript

- **JavaScript** adalah **bahasa skrip yang ditempelkan pada kode HTML dan diproses pada sisi klien (Browser)**, sehingga kemampuan dokumen HTML menjadi lebih luas.
 - JavaScript memungkinkan untuk memvalidasi masukan-masukan pada formulir sebelum dikirim ke server
 - JavaScript dapat mengimplementasi web secara dinamis dan interaktif
- Javascript bergantung kepada browser(navigator) yang memanggil halaman web yang berisi skrip skrip dari javascript yang terselip di dalam dokumen HTML.
- Javascript tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya
- Javascript merupakan bahasa pemrograman yang bisa diterapkan pada HTML, Web, aplikasi komputer, servers, laptop, tablet, smart phones, dan lainnya.

Pengenalan Javascript

- Javascript biasa digunakan untuk memanipulasi elemen-elemen HTML dan menambahkan Style secara otomatis atau lebih sederhananya membuat dokumen HTML menjadi lebih Interaktif.
- Untuk menambahkan Script javascript pada dokumen HTML, dapat digunakan tag script dan diletakkan di dalam head atau di area manapun di dalam tag body.

Contoh Penulisan Script

```
<body>
```

```
  <script type="text/ javascript">
```

```
    //...Script javascript diketik disini...
```

```
    //contoh:
```

```
      document.write( "Hello world" );
```

```
    </script>
```

```
</body>
```

Letak JavaScript dalam dokumen HTML

- Menggunakan tag `<SCRIPT>`
 - Tag `<SCRIPT>` diletakkan diantara bagian kepala dari dokumen HTML, yaitu bagian antara tag `<HEAD>` dan `</HEAD>`. Pemanggilan fungsi JavaScript (atau disebut juga *event*) diletakkan di bagian badan dokumen HTML atau bisa kita sebut diantara tag `<BODY>` dan `</BODY>`.
 - Keterangan tambahan di dalam tag `<SCRIPT>` menunjukkan jenis bahasa yang digunakan dan versinya, contohnya `"JavaScript"`, `"JavaScript 1.1"`, `"JavaScript 1.2"` untuk bahasa JavaScript

Letak JavaScript dalam dokumen HTML

- Contoh :

```
<HTML>
<HEAD>
    <TITLE>Contoh Program Javascript</TITLE>
</HEAD>
<BODY>
    <SCRIPT language="Javascript">
        <!--
            alert("Hallo !");
        // -->
    </SCRIPT>
</BODY>
</HTML>
```

Letak JavaScript dalam dokumen HTML

- Menggunakan file eksternal
 - Menuliskan kode program JavaScript dalam suatu file teks dan kemudian file teks yang berisi kode JavaScript di panggil dari dalam dokumen HTML (khusus Netscape mulai versi 3 ke atas).

```
<SCRIPT LANGUAGE="Javascript"
SRC="url/file.js"> </SCRIPT>
```

dimana url/file.js adalah lokasi dan nama file yang berisi kode JavaScript, jika perintah tambahan SRC tidak disertakan maka tag Script akan mencari kode yang terletak di dalam tag Script.

Letak JavaScript dalam dokumen HTML

- Melalui event tertentu
 - Event adalah sebutan dari satu action yang dilakukan oleh user, contohnya seperti klik tombol mouse.

```
<tag eventHandler="kode  
Javascript yang akan  
dimasukkan">
```

dimana **eventHandler** adalah nama dari event tersebut.

JavaScript Sebagai Bahasa Berorientasi Objek

- JavaScript memperlakukan elemen elemen yang tampil di jendela navigator sebagai suatu **obyek** , yang artinya adalah elemen :
 - Diklasifikasikan berdasarkan hirarki/tingkatan khusus sehingga kita bisa mengetahui dimana letak/lokasi obyek itu sebenarnya.
 - Diasosiasikan dengan kondisi atau sifat sifat khusus (properti)

- Ilustrasi :

Kebun

□ □ Pohon

- Dahan
 - Daun
 - Sarang Burung
 - □ Panjang = 20
 - □ Warna = kuning
 - □ Tinggi = 4
- Batang
- Akar

□ □ Sangkar Ternak

- Ayam
- Bebek

- Sarang burung yang berada di atas pohon dapat di tuliskan sebagai berikut :
Kebun.Pohon.Dahan.Sarang Burung
- Bila ingin mengecat atau mengganti warna sarang burung yang terletak di atas pohon, maka perintahnya :
Kebun.Pohon.Dahan.Sarang Burung.warna=hijau

Properti

- Properti adalah atribut dari sebuah objek.
- Penulisannya (dipisahkan dengan tanda “.”) :
nama_objek . nama_properti
- Properti dapat diberi nilai, penulisannya :
objek . properti = nilai

- Contoh :

```
<HTML>
<HEAD>
<TITLE>Properti defaultStatus</TITLE>
</HEAD>
<BODY>
<H1>Tes defaultStatus</H1>
<SCRIPT LANGUAGE = "JavaScript">
<!--
  window.defaultStatus = "Selamat belajar JavaScript" ;
  //-->
</SCRIPT>
</BODY>
</HTML>
```

Nama Properti

Nilai

Nama Objek

Metode

- Properti adalah suatu kumpulan kode yang digunakan untuk melakukan sesuatu tindakan terhadap objek.
- Penulisannya (dipisahkan dengan tanda “.”) :
nama_objek . nama_metode(“parameter”)

- Contoh :

```
<HTML>
<HEAD>
<TITLE>Skrip Javascript</TITLE>
</HEAD>
<BODY>
<SCRIPT language="Javascript">
<!--
    document.write("Selamat Mencoba JavaScript <BR>");
    document.write("Semoga Sukses" !");
// -->
</SCRIPT>
</BODY>
</HTML>
```

Nama Metode

Parameter

Nama Objek

JavaScript dapat mengubah Elemen HTML

- HTML DOM (**D**ocument **O**bject **M**odel) merupakan standar W3C untuk mengakses elemen-elemen HTML.
- JavaScript dapat memanipulasi DOM (mengubah isi dari HTML).
- Baris berikut merupakan contoh untuk mengubah isi (innerHTML) dari sebuah elemen HTML yang diidentifikasi dengan id="demo":

```
document.getElementById( "demo" ).innerHTML =  
"Hello JavaScript";
```

Contoh Lengkap

```
<h1>JavaScript</h1>
```

```
<p>JavaScript dapat mengubah isi elemen  
HTML:</p>
```

```
<button type="button"  
  onclick="myFunction()">Click Me!</button>
```

```
<p id="demo">Ini adalah sebuah demonstrasi.</p>
```

```
<script>
```

```
function myFunction() {
```

```
  document.getElementById("demo").innerHTML =  
  "Selamat datang Javascript!";
```

```
}
```

```
</script>
```

JavaScript dapat mengubah Elemen HTML

- Method **document.getElementById()** adalah salah satu dari banyak method dalam HTML DOM.
- JavaScript juga dapat digunakan untuk:
 - Mengubah elemen HTML
 - Menghapus elemen HTML
 - Membuat elemen HTML baru
 - Mengkopi dan menggandakan elemen HTML
 - dan banyak lagi ...

JavaScript Untuk Mengubah Atribut HTML

- Contoh berikut digunakan untuk mengubah nilai dari atribut *source* (src) dari elemen `` HTML:

Contoh Web

JavaScript Untuk Mengubah Style HTML (CSS)

- Mengubah style elemen HTML merupakan varian dari pengubahan atribut HTML.
- Dengan JavaScript, hampir semua nilai CSS (value) apapun dapat diubah.

Contoh Web

JavaScript Untuk Validasi Data

- JavaScript sering digunakan untuk melakukan validasi data input.

Contoh Web

Syntax JavaScript

- JavaScript adalah bahasa **scripting**.
- Bahasa scripting merupakan bahasa pemrograman tingkat rendah (lightweight programming language).
- Kalimat-kalimat dalam bahasa pemrograman disebut dengan **statement**.
- Secara prinsip, aturan untuk menyusun kalimat-kalimat tersebut dalam sebuah bahasa, maka disebut sebagai **syntax**.

JavaScript Statements

- Statement dalam JavaScript terdiri atas kumpulan-kumpulan dari :
 - Values,
 - Expressions,
 - Operators,
 - Variables,
 - Keywords, dan
 - Comments.

JavaScript Values (Literals)

- Dalam bidang ilmu komputer, nilai pasti (**fixed values**) disebut sebagai **literal**.
- Dengan JavaScript, aturan yang paling penting untuk menuliskan **fixed values** (konstanta) adalah :
 - **Number Literal** yang dapat ditulis secara desimal ataupun tidak, dan juga bisa ditulis dengan notasi scientific (e):
 - Contoh:
3.14, 1001, 123e5

JavaScript Values (Literals)

- Dalam bidang ilmu komputer, nilai pasti (**fixed values**) disebut sebagai **literal**.
- Dengan JavaScript, aturan yang paling penting untuk menuliskan **fixed values** (konstanta) adalah :
 - **String Literal** yang dapat ditulis menggunakan tanda petik tunggal atau petik dua:
 - Contoh:
“Informatika”, ‘Informatika’

JavaScript Values (Literals)

- Dalam bidang ilmu komputer, nilai pasti (**fixed values**) disebut sebagai **literal**.
- Dengan JavaScript, aturan yang paling penting untuk menuliskan **fixed values** (konstanta) adalah :
 - **Expression Literal** yang digunakan untuk mengevaluasi value:
 - Contoh:
 $5 + 6, 5 * 10$

JavaScript Variable

- Dalam bahasa pemrograman (juga aljabar), **variabel** digunakan untuk menyimpan value.
- JavaScript menggunakan kata kunci **var** untuk **mendefinisikan** variabel.
- **Tanda sama dengan** digunakan untuk **memasukkan value** ke dalam variabel (seperti pada aljabar).

JavaScript Variable

- Pada contoh berikut, panjang didefinisikan sebagai variabel.
- Kemudian, panjang diberi nilai (value) 6:

```
var panjang;  
panjang = 6;
```

- Sebuah literal adalah **fixed** value. Variabel adalah **name**. Sebuah variabel dapat memiliki nilai **variabel**.

JavaScript Variable (Identifier)

- Semua **variabel** (dan nama fungsi JavaScript) harus diidentifikasi dengan **unique names**.
- Nama unik ini disebut sebagai **identifier**.
- Identifier dapat berupa nama singkat (seperti x dan y), atau nama yang lebih deskriptif (umur, jumlah, totalVolume).
- Aturan umum membuat nama variabel (unique identifier) :
 - Nama harus diawali dengan sebuah huruf
 - Nama juga dapat diawali dengan \$ dan _
 - Nama dapat berisi huruf-huruf, digit, underscore, dan simbol dollar.
 - Nama bersifat case sensitive (y dan Y adalah variabel berbeda)
 - Kata kunci (cth: JavaScript) tidak bisa dipakai sbg identifier

JavaScript Operators

- JavaScript menggunakan 4 operator aritmatika utama (**arithmetic**) dan sebuah operator modulus untuk menghitung angka (persis seperti aljabar):

No	Operator	Simbol	Contoh
1	Perkalian	*	$5 * 4 = 20$
2	Pembagian	/	$20 / 5 = 4$
3	Penjumlahan	+	$5 + 4 = 9$
4	Pengurangan	-	$5 - 4 = 1$
+	Modulus	%	$10 \% 4 = 2$

Aturan Syntax JavaScript

- Semua perintah dalam JavaScript berupa **case sensitive**.
- Variabel **lastName** dan **lastname** merupakan 2 buah variabel yang berbeda.
- JavaScript tidak mengartikan **VAR** atau **Var** sebagai kata kunci perintah **var**.

JavaScript Keywords

Keyword	Deskripsi
break	Menghentikan perintah switch atau perulangan
catch	Menandai blok statement untuk dieksekusi jika terjadi error pada blok try
continue	Keluar dari loop dan mulai kembali dari atas
debugger	Menghentikan eksekusi JavaScript, dan memanggil (jika ada) fungsi debugging
do ... while	Mengeksekusi blok statement, dan mengulang blok tersebut selama kondisi terpenuhi
for	Menandai blok statement yang akan dieksekusi selama kondisi terpenuhi
for ... in	Menandai blok statement yang akan dieksekusi untuk tiap elemen pada suatu objek (atau array)

JavaScript Keywords

Keyword	Deskripsi
function	Mendeklarasikan fungsi
if ... else	Menandai blok statement yang akan dieksekusi tergantung pada kondisi
return	Keluar dari fungsi
switch	Menandai blok statement yang akan dieksekusi tergantung pada kasus tertentu
throw	Menghilangkan (membangkitkan) sebuah error
try	Menerapkan error handling pada blok statement
var	Mendeklarasikan variable
while	Menandai blok statement yang akan dieksekusi selama kondisi terpenuhi

Operator Penugasan

- Dalam JavaScript, simbol sama dengan (=) merupakan operator penugasan (assignment), bukan operator persamaan (equal to)
- Hal ini berbeda dengan aljabar.
- Contoh berikut tidak ada perubahan dalam aljabar:

$$x = x + 5$$

Operator Penugasan

- Dalam JavaScript, hal tersebut sangat berpengaruh dan mengubah nilai, yaitu menugaskan hasil operasi $x + 5$ dan menyimpan ke dalam variabel x .
- Operator persamaan (equal to) dalam JavaScript ditulis dengan simbol `==` or `===`.

Tipe Data dalam JavaScript

- Variabel JavaScript dapat menangani banyak **tipe data**: bilangan, string, array, object dan lainnya:

```
var length = 16; // number literal
```

```
var points = x * 10; // expression literal
```

```
var lastName = "Johnson"; // string literal
```

```
var cars = ["Saab", "Volvo", "BMW"]; // array  
literal
```

```
var x = {firstName:"John", lastName:"Doe"}; //  
object literal
```