

JavaScript

Elfan Nofiari

Departemen Teknik Informatika

ITB

JavaScript

- Menyediakan fasilitas pemrograman untuk lingkungan browser web
- Membuat web menjadi lebih dinamis dan interaktif
 - aplikasi bereaksi terhadap suatu event dalam halaman web
 - membaca dan menulis ulang tag HTML
 - memvalidasi data yang akan dikirimkan ke server
- Didukung oleh banyak (tidak semua) browser web
- Tidak membutuhkan tool khusus, dapat mempergunakan editor teks biasa seperti halnya pada saat membuat dokumen HTML
- JavaScripts \neq Java
 - JavaScripts dikembangkan Netscape \rightarrow script
 - Java dikembangkan Sun Microsystems \rightarrow applet
- Variant
 - ECMAScripts
 - JScript

Program JavaScript

- Berupa script/program yang disisipkan di dalam dokumen HTML
- Peletakan JavaScript
 - Dalam internal dokumen HTML
 - Standar HTML 4.01

```
<script type="text/javascript">
...statement...
</script>
```
 - Tag Lama

```
<script language="JavaScript">
...statement...
</script>
```
 - Dalam file external
 - Script JavaScript dituliskan pada file tersendiri (ekstensi file .js)
 - Pemanggilan file JavaScript:

```
<script src="namafile.js"></script>
```

Tipe Data

- Numerik
 - Bilangan: integer, float
- String
 - Karakter, angka, tanda baca
- Boolean
 - Nilai 0 untuk false
 - Bukan 0 untuk true

Variabel

- Penamaan
 - Case sensitive
 - Harus diawali dengan huruf atau underscore
 - Bukan keyword JavaScript
 - Tidak mengandung karakter tanda baca atau karakter untuk operasi pada script
- Scope
 - Global (default)
 - Dapat diakses oleh bagian script mana pun di dalam dokumen
 - Lokal
 - Dideklarasikan di dalam fungsi
 - Hanya dapat diakses oleh fungsi yang mendefinisikannya
- Deklarasi
 - `var namavariabel=nilai (lokal)`
 - `namavariabel=nilai (global)`

Operator

- Operator Aritmetika
- Operator Assignment
- Operator Perbandingan
- Operator Logika
- Operator String
- Operator Kondisi
- Operator new

Operator Aritmetika

Operator	Penggunaan	Contoh	Hasil
+	Penjumlahan	$x=2$ $x+2$	4
-	Pengurangan	$x=2$ $5-x$	3
*	Perkalian	$x=4$ $x*5$	20
/	Pembagian	$15/5$ $5/2$	3 2.5
%	Modulus (pembagian sisa)	$5\%2$ $10\%8$ $10\%2$	1 2 0
++	Increment	$x=5$ $x++$	$x=6$
--	Decrement	$x=5$ $x--$	$x=4$

Operator Assignment

Operator	Contoh	Sama seperti
=	$x=y$	$x=y$
+=	$x+=y$	$x=x+y$
-=	$x-=y$	$x=x-y$
=	$x=y$	$x=x*y$
/=	$x/=y$	$x=x/y$
%=	$x%=y$	$x=x\%y$

Operator Perbandingan

Operator	Penjelasan	Contoh
==	Sama dengan	5==8 menghasilkan false
!=	Tidak sama dengan	5!=8 menghasilkan true
>	Lebih besar dari	5>8 menghasilkan false
<	Lebih kecil dari	5<8 menghasilkan true
>=	Lebih besar atau sama dengan	5>=8 menghasilkan false
<=	Lebih kecil atau sama dengan	5<=8 menghasilkan true

Operator Logika

Operator	Penggunaan	Contoh
&&	and	x=6 y=3 (x < 10 && y > 1) menghasilkan true
	or	x=6 y=3 (x==5 y==5) menghasilkan false
!	not	x=6 y=3 x != y menghasilkan true

Operator String

- Concatenation (+)
 - Menggabungkan 2 string

- Contoh:

```
teks1="Hello "
```

```
teks2="JavaScript "
```

```
teks3=teks1+teks2
```

- Hasil:

```
teks3="Hello JavaScript "
```

Operator Kondisi

- Proses pengisian nilai ke dalam variabel berdasarkan nilai dari suatu ekspresi

- Sintaks

ekspresi ? nilaiBilaTrue : nilaiBilaFalse

- Contoh:

```
Bi1=5
```

```
Jenisbi1=(Bi1%2==0)?"Genap":"Ganjil"
```

Operator new

- Sintaks

```
var namaVar=new objectJavaScript();
```

- Mendefinisikan variabel bertipe objek.

- Contoh:

```
var tanggal=new Date( )
```

Konstruksi dasar program JavaScript

- **Sequence** : berurutan, dikerjakan dari baris pertama ke baris berikutnya, dari atas ke bawah
- **Pencabangan/pemilihan** : dikerjakan atau tidak tergantung suatu kondisi
 - `if`
 - `if .. else`
 - `switch .. case`Dapat divariasikan dengan `break` dan `continue`
- **Pengulangan** : dikerjakan berulang-ulang selama kondisi tertentu dipenuhi
 - `while`
 - `do .. while`
 - `for`

Contoh Sequence (#1)

```
<html>
<body>
<h1>Tampilan Javascript & HTML</h1>
<script type="text/javascript">
document.write("Hello..saya dari JavaScript!")
</script>
<p>
Saya asli dari HTML
</body>
</html>
```

Contoh Sequence (#2)

```
<html>
<body>
<h1>Menjumlahkan 2 bilangan</h1>
<script type="text/javascript">
  var a=4
  B=28
  Jml=a+B
  document.write("a = " + a)
  document.write("<br>")
  document.write("B=" + B)
  document.write("<br>")
  document.write("Jml=" + Jml)
</script>
</body>
</html>
```

if

- Sintaks:

```
if (ekspresi) {  
    ...statement...  
}
```

- Digunakan untuk melakukan menentukan apakah blok statement setelah if dikerjakan atau tidak. Dikerjakan apabila nilai ekspresi true.

Contoh if

```
<html>
<body>
<h1>Demo if</h1>
Jam sekarang adalah:
<script type="text/javascript">
    var tg=new Date()
    jam=tg.getHours()
    document.write(jam)
    if (jam>17) document.write("Sedang lembur?")
</script>
</body>
</html>
```

if .. else

- Sintaks

```
if (ekspresi) {  
    ...Blok_statement_if...  
}  
else {  
    ...Blok_statement_else...  
}
```

- Digunakan untuk melakukan pengambilan keputusan blok statement mana yang harus dikerjakan, berdasarkan nilai ekspresi. Jika nilai ekspresi true maka blok statement di belakang if yang dikerjakan, apabila nilai ekspresi false maka blok statement di belakang else yang dikerjakan.

Contoh if .. else

```
<html>
<body>
<h1>Ganjil Genap</h1>
<p>Bilangan yang diinput dari keyboard termasuk bilangan
  <i><b>
<script type="text/javascript">
  var bil=prompt("Masukkan bilangan",2)
  hasil=bil % 2
  if (hasil==0) {
    document.write("Genap!")
  }
  else {
    document.write("Ganjil!")
  }
</script>
</b></i></p>
</body>
</html>
```

switch ... case

- Sintaks

```
switch (ekspresi) {  
    case nilai1 :  
        Statementberdasarnilai1;  
    [case nilai2 :  
        Statementberdasarnilai2;  
    ...]  
    [default :  
        StatementDefault;]  
}
```

- Pengambilan keputusan/pemilihan blok statement yang harus dieksekusi berdasarkan banyak kemungkinan nilai ekspresi

Contoh switch ... case

```
<html>
<body>
<h1>Menyebutkan bilangan</h1>
<p>Bilangan yang diinput dari keyboard adalah
<i><b>
<script type="text/javascript">
  var bil=prompt("Masukkan bilangan",2)
  document.write(bil+" : ")
  switch (bil) {
    case "0" : document.write("nol");break;
    case "1" : document.write("satu");break;
    case "2" : document.write("dua");break;
    default : document.write("tidak dikenal");
  }
</script>
</b></i>
</p>
</body>
</html>
```

while

- Sintaks

```
while (ekspresi) {  
    statement;  
    statement;  
    ...  
}
```

- Pengulangan blok statement selama kondisi benar (bernilai true), minimal 0 kali (blok statement mungkin tidak pernah dieksekusi)

Contoh while

```
<html>
<body>
<h1>While</h1>
<p>Menampilkan Angka 0..5</p>
<script type="text/javascript">
  i=0;
  while (i<6) {
    document.write(i+"<br>")
    i++
  }
</script>
</body>
</html>
```

do-while

- Sintaks

```
do {  
    statement;  
    statement;  
    ...  
} while (ekspresi)
```

- Pengulangan blok statement selama kondisi benar (bernilai true), minimal 1 kali

Contoh do-while

```
<html>
<body>
<h1>Do While</h1>
<p>Menampilkan Angka 0..5</p>
<script type="text/javascript">
  i=0;
  do {
    document.write(i+"<br>")
    i++
  } while (i<6)
</script>
</body>
</html>
```

for

- Sintaks

```
for (inisialisasi;terminasi;increment) {  
    statement;  
    statement;  
    ...  
}
```

- Pengulangan blok statement sejumlah n kali yang ditentukan, n adalah bilangan yang menunjukkan jumlah pengulangan.

Contoh for

```
<html>
<body>
<h1>For</h1>
<p>Menampilkan Header</p>
<script type="text/javascript">
  for (i=1; i<=6; i++){
    document.write("<h"+i+">Header"+i+"</h"+i+">")
  }
</script>
</body>
</html>
```

Fungsi

- Suatu blok statement dapat dipisahkan penulisannya dari blok program utama, dan dieksekusi apabila dipanggil
- Alasan penggunaan fungsi
 - Modular
 - Reusable
- Jenis
 - Built-in (berupa method dari object built-in)
 - User defined
- Bentuk
 - Dapat mengandung argumen
 - Dapat menghasilkan nilai

Fungsi User-defined

- Fungsi yang dikembangkan oleh pemrogram, karena tidak/belum disediakan oleh JavaScript (User Defined Function – UDF)

- Deklarasi

```
function namaFungsi ([parameter]) {  
    ... statement2 fungsi...  
    [return nilai]  
}
```

Contoh fungsi

```
<html>
<body>
<h1>Menjumlahkan 2 bilangan menggunakan Fungsi</h1>
<script type="text/javascript">
  var a=4
  b=28
  document.write("a = "+a+"<br>"+"b="+b+"<br>")
  document.write("Jmlah a+b =" + jumlah(a,b))
  function jumlah(a,b) {
    return a+b
  }
</script>
<p>
</body>
</html>
```

Event

- Kejadian yang terjadi pada halaman web yang dapat diidentifikasi dan ditangani menggunakan JavaScript
 - MouseOver
 - Pemakai menggerakkan mouse di atas suatu link
 - Click
 - Pemakai menekan tombol atau mengklik suatu link
 - Focus
 - Kursor pada field isian pada suatu form
 - Blur → lost focus
 - Kursor tepat berpindah dari suatu field isian form
 - Change
 - Perubahanterjadi pada suatu field isian
 - Select
 - Proses highlight karakter pada suatu field isian
 - Submit
 - Tombol submit ditekan
 - Load
 - Proses menampilkan dokumen
 - UnLoad
 - Selesai/keluar dari suatu halaman dokumen
- Untuk mendefinisikan event handler, buat atribut pada tag HTML dengan nama "on" diikuti nama event, misalnya

```
<p onMouseOver = "alert('halo')">tes</p>
```

Object

- Definisi Objek
 - Umum : suatu cara untuk mengorganisir informasi, termasuk aksi-aksi yang bisa dilakukan untuk memanipulasi informasi tersebut
 - Teknis : entitas program pada saat runtime yang memiliki **property** (atribut) dan **method** (aksi).
- JavaScript memiliki sejumlah objek internal (built-in).
- JavaScript dapat mengakses objek eksternal (built-in), misalnya milik browser (→ Document Object Model).
- JavaScript memungkinkan pembuatan objek baru dengan mendefinisikan property dan method.

Object Internal JavaScript

- String
- Array
- Math
- Date

String Object

Properties

- **length** An integer value containing the length of the string expressed as the number of characters in the string.

Methods

- **charAt(*index*)** Returns the character at the location specified by *index*.
- **indexOf(*findString, startingIndex*)** Returns the index of the first occurrence of *findString*, starting the search at *startingIndex* where *startingIndex* is optional-if it is not provided, the search starts at the start of the string.
- **lastIndexOf(*findString, startingIndex*)** Returns the index of the last occurrence of *findString*. This is done by searching backwards from *startingIndex*. *startingIndex* is optional and assumed to be the last character in the string if no value is provided.
- **split(*separator*)** Returns an array of strings created by splitting the string at every occurrence of *separator*.
- **substring(*firstIndex, lastIndex*)** Returns a string equivalent to the substring starting at *firstIndex* and ending at the character before *lastIndex*. If *firstIndex* is greater than *lastIndex*, the string starts at *lastIndex* and ends at the character before *firstIndex*.
- **toLowerCase()** Returns a string containing the value of the String object with all character converted to lowercase.
- **toUpperCase()** Returns a string containing the value of the String object with all character converted to uppercase.

Contoh String Object

```
<html>
<script type="text/javascript">
str = "Sangkuriang is from Bandung";
alert(str.length);           //27
alert(str.charAt(4));        //k
alert(str.indexOf("from"));  //15
alert(str.lastIndexOf("an")); //21
alert(str.split(" "));       //Sangkuriang,is,from,Bandung
alert(str.substring(6, 11)); //riang
alert(str.toLowerCase());    //sangkuriang is from bandung
alert(str.toUpperCase());     //SANGKURIANG IS FROM BANDUNG

alert("the end".toUpperCase()); //THE END
</script>
</html>
```

Array

- Array adalah himpunan nilai yang elemennya dapat diacu berdasarkan indeks (angka) atau nama (string)
- Contoh: Array mhs yang berisi nama-nama mahasiswa yang diindeks berdasar nomor mahasiswa, sehingga mhs[1] adalah mahasiswa nomor 1, mhs[2] mahasiswa nomor 2, dst
- Array dalam JavaScript dapat berisi type informasi apapun: number, boolean, string, array, object
- Indeks dalam array JavaScript dimulai dari 0
- Pembuatan object Array

```
array_name = new Array();
```

```
array_name = new Array(array_length);
```

```
array_name = new Array(element0, element1, ..., elementN);
```

```
array_name = [element0, element1, ..., elementN];
```

```
array_name = {keyA:valueA, keyB:valueB, ..., keyN:valueN};
```

Array Object

Properties

- **length** An integer value reflecting the number of elements in an array.

Methods

- **join(*string*)** Returns a string containing each element of the array separated by *string*.
- **reverse()** Reverses the order of an array.
- **sort(*function*)** Sorts an array based on function which indicates a *function* defining the sort order. *function* can be omitted in which case the sort defaults to dictionary order.

Contoh Array (menggunakan indeks)

```
<html>
<script type="text/javascript">
//cara 1 pendefinisian array
mhs = new Array();
mhs[0] = "Bevin";
mhs[1] = "Andini";
mhs[2] = "Citra";
//cara 2 pendefinisian array
mhs = new Array("Bevin", "Andini", "Citra");
//cara 3 pendefinisian array
mhs = ["Bevin", "Andini", "Citra"];

//cara pengaksesan array
document.write("Mahasiswa pertama adalah "+mhs[0]+".<br>"); //Bevin
document.write("Mahasiswa terakhir adalah "+mhs[2]+".<br>"); //Citra
//cara pengaksesan array menggunakan loop
for (i=0; i<mhs.length; i++) {
    document.write(mhs[i] + "<br>");
}
//mengurutkan array
mhs.sort();
//menggabungkan array
document.write(mhs.join("-")); //Andini-Bevin-Citra
</script>
</html>
```

Contoh Array (menggunakan nama)

```
<html>
<script type="text/javascript">
//cara 1 pendefinisian array
nilai = new Array();
nilai["Bevin"] = 20;
nilai["Andini"] = 42;
nilai["Citra"] = 37;

//cara 2 pendefinisian array
nilai = {"Bevin":20, "Andini":42, "Citra":37};

//cara pengaksesan array
document.write("Nilai Bevin adalah "+nilai["Bevin"]+".<br>"); //20

//cara pengaksesan array menggunakan loop
for (nama in nilai) {
    document.write("Nilai "+nama+" adalah "+nilai[nama]+"<br>");
}

</script>
</body>
</html>
```

Math Object

Properties

- **E** The value of Euler's constant (roughly 2.718) used as the base for natural logarithms.
- **LN10** The value of the natural logarithm of 10 (roughly 2.302).
- **LN2** The value of the natural logarithm of 2 (roughly 0.693).
- **PI** The value of pi-used in calculating the circumference and area of circles (roughly 3.1415).
- **SQRT1_2** The value of the square root of one-half (roughly 0.707).
- **SQRT2** The value of the square root of two (roughly 1.414).

Methods

- **abs(*number*)** Returns the absolute value of *number*.
- **sin(*number*), cos(*number*), tan(*number*), asin(*number*), acos(*number*), atan(*number*)** Standard trigonometric functions with *number* in radians.
- **atan2(*number1*,*number2*)** Returns the angle of the polar coordinate corresponding to the Cartesian coordinate (*number1*, *number2*). [Not I]
- **ceil(*number*)** Rounds up to the next integer.
- **exp(*number*)** Returns the value of e to the power of *number*.
- **floor(*number*)** Rounds down to the nearest integer.
- **log(*number*)** Returns the natural logarithm of *number*.
- **max(*number1*,*number2*)** Returns the greater of *number1* and *number2*.
- **min(*number1*,*number2*)** Returns the smaller of *number1* and *number2*.
- **pow(*number1*,*number2*)** Returns the value of *number1* to the power of *number2*.
- **random()** Returns a random number between zero and one.
- **round(*number*)** Rounds to the closest integer.
- **sqrt(*number*)** Returns the square root of *number*.

Contoh Math Object

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

document.write(Math.PI + "<br>");           //3.141592653589793
document.write(Math.abs(-8.23) + "<br>");    //8.23
document.write(Math.random() + "<br>");      //0.34306549528684166
document.write(Math.min(478, 237) + "<br>"); //237

</script>
</body>
</html>
```

Date

- Object date dapat digunakan untuk mengeset, mengambil, dan memanipulasi tanggal dan waktu
- JavaScript menyimpan Date sebagai angka integer berupa banyaknya milidetik sejak 1 Januari 1970, pukul 00:00:00. Tanggal sebelum itu tidak dapat diakses
- Pembuatan object Date

obj_name = **new Date()**;

obj_name = **new Date(string_date)**;

format of *string_date* = "month date, year hour:min:sec"

obj_name = **new Date(list_of_integers)**;

list_of_integers = year, month, date[, hour, min, sec]

Date Object

Methods

- **getDate()** The day of the month as an integer from 1 to 31.
- **getDay()** The day of the week as an integer from 0 to 6 (where 0 is Sunday, 1 is Monday, etc.).
- **getHours()** The hours the time as an integer from 0 to 23.
- **getMinutes()** The minutes as an integer from 0 to 59.
- **getMonth()** The month as an integer from 0 to 11 (where 0 is January, 1 is February, etc.).
- **getSeconds()** The seconds as an integer from 0 to 59.
- **getTime()** The time of the current Date object as an integer representing the number of milliseconds since January 1, 1970 at 00:00:00.
- **getFullYear()** The year as an integer representing the year minus 1900.
- **getFullYear()** The year as an integer the year.
- **parse(*dateString*)** The number of milliseconds between January 1, 1970 at 00:00:00 and the date specified in *dateString*. *dateString* should take the format
Day, DD Mon YYYY HH:MM:SS TZN or *Mon DD, YYYY*
- **setDate(*dateValue*)** Sets the day of the month. *dateValue* is an integer from 1 to 31.
- **setHours(*hoursValue*)** Sets the hour. *hoursValue* is an integer from 0 to 23.
- **setMinutes(*minutesValue*)** Sets the minutes. *minutesValue* is an integer from 0 to 59.
- **setMonth(*monthValue*)** Sets the month. *monthValue* is an integer from 0 to 11.
- **setSeconds(*secondsValue*)** Sets the seconds. *secondsValue* is an integer from 0 to 59.
- **setTime(*timeValue*)** Sets the value for the current Date object. *timeValue* is an integer representing the number of milliseconds since January 1, 1970 at 00:00:00.
- **setYear(*yearValue*)** Sets the yea. *yearValue* is an integer greater than 1900.
- **toGMTString()** Returns the value of the current Date object in GMT as a string using Internet conventions in the form *Day, DD Mon YYYY HH:MM:SS GMT*

Contoh Date Object

```
<html>
<script type="text/javascript">
//mengambil informasi waktu saat ini
hari_ini = new Date();
document.write("Tanggal = "+hari_ini.getDate() +"<br>");

//sembarang waktu
waktu = new Date("March 15, 2004 17:04:22");
document.write("Tanggal = "+waktu.getDate() +"<br>"); //15
document.write("Bulan = "+waktu.getMonth() +"<br>"); //2
document.write("Tahun = "+waktu.getFullYear() +"<br>"); //2004
document.write("Hari = "+waktu.getDay() +"<br>"); //1 (Monday)

//mengubah informasi waktu
waktu.setDate(13);
document.write("Tanggal = "+waktu.getDate() +"<br>"); //13

//mengubah informasi waktu di luar jangkauan
waktu.setDate(32);
document.write("Tanggal = "+waktu.getDate() +"<br>"); //1
document.write("Bulan = "+waktu.getMonth() +"<br>"); //3
</script>
</html>
```