



JavaScript

Teknologi Web

Agung Nugroho
Universitas Pelita Bangsa Bekasi



Pengenalan Javascript

Apa itu JavaScript?

- JavaScript adalah bahasa “Scripting”, bukan bahasa pemrograman.
- JavaScript didisain untuk membuat halaman HTML menjadi lebih interaktif.
- JavaScript biasanya disisipkan langsung pada halaman HTML.
- Client Side scripting.
- Apakah JavaScript sama dengan Java? **TIDAK**. JavaScript dan Java sangat berbeda baik itu dari bahasa maupun dari konsep dan disainnya.
- JavaScript bersifat **Case Sensitive**.



Pengenalan Javascript

- Nama resmi JavaScript : **ECMAScript**.
- ECMAScript dikembangkan oleh ECMA Organization.
- **ECMA-262** adalah standar resmi JavaScript.
- Diciptakan oleh Brendan Eich. Muncul pertama kali di semua browser Netscape dan Microsoft pada tahun 1996.
- Disetujui sebagai standar internasional pada tahun 1998.



Keunggulan Javascript

- JavaScript dapat menempatkan text secara dinamis pada halaman HTML.
- JavaScript dapat dieksekusi berdasarkan event tertentu.
- JavaScript dapat membaca dan menuliskan elemen-elemen HTML.
- JavaScript dapat digunakan untuk memvalidasi data dari form, sebelum dikirimkan ke server.
- JavaScript dapat digunakan untuk mendeteksi browser pengunjung website.
- JavaScript dapat digunakan untuk menyimpan dan mengambil informasi dari komputer pengunjung website.



Penggunaan Javascript

- Menggunakan tag HTML <script>
- Contoh:

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



Penempatan Javascript

- Ditempatkan pada tag <head>
- Ditempatkan pada tag <body>
- Ditempatkan pada tag <head> dan <body>
- Sebagai file external.

Javascript pada Tag <head>

- Dengan menempatkan syntax JavaScript pada tag head, tidak akan mengganggu isi dari halaman web karena semua script disatukan pada satu tempat.
- Contoh:

```
<head>  
    <script type="text/javascript">  
        alert("JavaScript pada tag Head.");  
    </script>  
</head>
```

Javascript pada Tag <body>

- Contoh:

```
<body>
  <h1>Belajar JavaScript</h1>
  <p>

  <script type="text/javascript">
      document.write("JavaScript pada tag Body.");
  </script>

  </p>
</body>
```




Javascript Sebagai File External

- Jika ingin menjalankan script JavaScript yang sama di banyak halaman, gunakan JavaScript sebagai file independen.
- Simpan file tersebut dengan ekstensi .js
- Contoh:

```
<head>  
    <script type="text/javascript" src="external.js">  
    </script>  
</head>
```

Penggunaan Javascript External

File contoh1.js

```
document.write("<h2>Text ini ditulis dengan JavaScript</h2>");
```

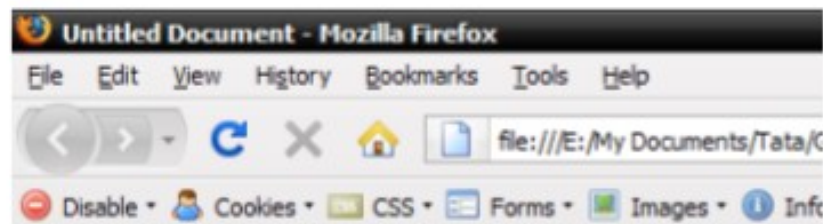
File contoh1.html

```
<head>
```

```
<script type="text/javascript" src="contoh1.js"></script>
```

```
</head>
```

Hasil



Text ini ditulis dengan JavaScript



Variabel

- Aturan penulisan Variabel:
 - Nama variabel bersifat Case Sensitive
 - Harus dimulai dengan huruf atau karakter underscore.
- Deklarasi Variabel dengan statement **var**
 - `var nama;`
 - `var nama="Rahmat"`
- Variabel yang tidak di deklarasikan
 - `nama="Rahmat"`
 - `x=y+5`

Variabel

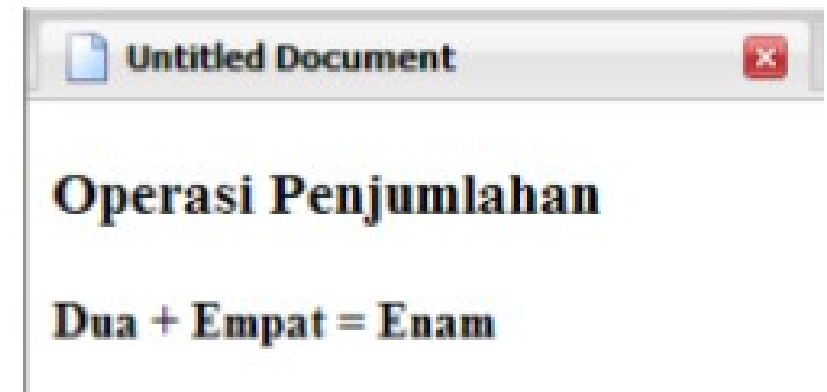
Contoh :

```
<body>
  <h3>Operasi Penjumlahan</h3>
  <b>Dua + Empat =

  <script type="text/javascript">
    var jumlah;
    jumlah = "Enam";
    document.write(jumlah);
  </script>

  </b>
</body>
```

Hasil :





Operator Aritmatika

- Macam-macam Operator Aritmatika pada JavaScript

Operator	Deskripsi	Contoh	Hasil
+	Penjumlahan	<code>x = 5 + 2;</code>	<code>x = 7</code>
-	Pengurangan	<code>x = 10 - 2;</code>	<code>x = 8</code>
*	Perkalian	<code>x = 5 * 4;</code>	<code>x = 20</code>
/	Pembagian	<code>x = 10 / 2;</code>	<code>x = 2.5</code>
%	Modulus	<code>x = 5 % 2;</code>	<code>x = 1</code>
++	Increment	<code>x++;</code>	<code>x = 1</code>
--	Decrement	<code>x--;</code>	<code>x = -1</code>

Contoh

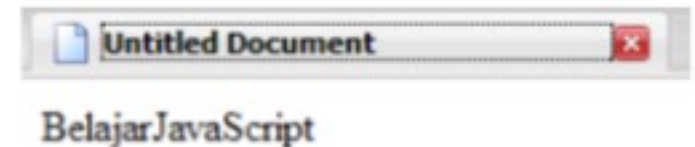
- Penggunaan operator + pada string.

Contoh :

```
<body>
  <script type="text/javascript">
    var x="Belajar";
    var y="JavaScript";
    z = x + y;

    document.write(z);
  </script>
</body>
```

Hasil :



Contoh

<code>x=5+5;</code> <code>document.write(x);</code>	→	10
--	---	----

<code>x="5"+"5";</code> <code>document.write(x);</code>	→	55
--	---	----

<code>x=5+"5";</code> <code>document.write(x);</code>	→	55
--	---	----

<code>x="5"+5;</code> <code>document.write(x);</code>	→	55
--	---	----

Jika menggabungkan
angka dengan string,
maka hasilnya akan
string.

Operator Assignment

- Macam-macam operator Assignment pada JavaScript.

Operator	Contoh	Sama Dengan	Hasil
=	x = "Belajar JS";		x = Belajar JS
+=	x += 6;	x = x + 6;	x = 8
-=	x -= 2;	x = x - 2;	x = 6
*=	x *= y;	x = x * y;	x = 20
/=	x /= 2;	x = x / 2;	x = 5
%=	x %= y;	x = x % y;	x = 1

Contoh

- Penggunaan operator Assignment


Contoh :

```
<script type="text/javascript">
    var x = 9;
    var y = 5;
    x += 20;
    x++;
    x -= y;
    document.write("Nilai x = "+x);
</script>
```

Hasil :



Nilai x = 25

- 
- Digunakan untuk menerangkan script JavaScript yang dibuat.
 - Mencegah script tersebut dieksekusi oleh browser.

Macam-macam komentar.

- Single Line
 - Dimulai dengan karakter : //
- Multi Line
 - Dimulai dengan karakter : /*
 - Diakhiri dengan karakter : */
 - **Contoh**

```
/*
```

```
    document.write("Script ini tidak akan dieksekusi");
```

```
*/
```

Operator Pembandingan

- Macam - macam operator pembandingan
 - Misal nilai $x = 5$.

Operator	Deskripsi	Contoh
<code>==</code>	Sama dengan	$x == 8 \rightarrow$ salah
<code>===</code>	Sama persis dengan (tipe data dan nilai)	$x === 5 \rightarrow$ benar $x === "5" \rightarrow$ salah
<code>!=</code>	Tidak sama dengan	$x != 8 \rightarrow$ benar
<code>></code>	Lebih besar dari	$x > 8 \rightarrow$ salah
<code><</code>	Lebih kecil dari	$x < 8 \rightarrow$ benar
<code>>=</code>	Lebih besar sama dengan dari	$x >= 8 \rightarrow$ salah
<code><=</code>	Lebih kecil sama dengan dari	$x <= 8 \rightarrow$ benar



Operator Logika

- Macam – macam operator logika
 - Misal nilai $x = 6$, dan $y = 3$.

Operator	Deskripsi	Contoh
&&	Dan (and)	$(x < 10 \ \&\& \ y > 1) \rightarrow \text{benar}$
	Atau (or)	$(x == 5 \ \ y == 3) \rightarrow \text{benar}$
!	Tidak (not)	$!(x == y) \rightarrow \text{benar}$



Pengkondisian

Pada JavaScript terdapat beberapa macam pengkondisian

- If (kondisi)
 - Gunakan statement ini untuk mengeksekusi source code hanya jika satu kondisi bernilai true.
- if (kondisi1) else (kondisi2)
 - Gunakan statement ini untuk mengeksekusi source code jika satu kondisi bernilai true, dan kondisi lain bernilai false.
- if (kondisi1) else if (kondisi2) else (kondisi3)
 - Gunakan statement ini untuk memilih satu dari banyak kondisi.
- Switch
 - Gunakan statement ini untuk memilih satu dari banyak kondisi.

Pengkondisian

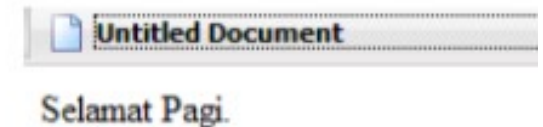
- if statement
- Sintaks

```
if (kondisi) {  
    aksi  
}
```

Contoh

```
<script type="text/javascript">  
    var jam = 5;  
    if (jam <= 10)  
    {  
        document.write("Selamat Pagi.");  
    }  
</script>
```

Hasil





Pengkondisian

- if (kondisi1) else (kondisi2)
- Sintaks

```
if (kondisi) {  
    aksi  
} else {  
    aksi2  
}
```

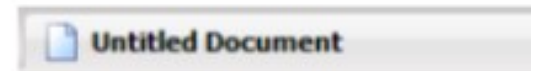
Pengkondisian

- If (kondisi) else

Contoh

```
<script type="text/javascript">
  var jam = 13;
  if (jam <= 10)
  {
    document.write("Selamat Pagi.");
  }
  else
  {
    document.write("Selamat Siang.");
  }
</script>
```

Hasil



Selamat Siang.

Pengkondisian

- if (kondisi1) else if (kondisi2) else (kondisi3)
- Sintaks

```
if (kondisi1) {  
    dieksekusi jika kondisi1 bernilai true  
}  
else if (kondisi2) {  
    dieksekusi jika kondisi1 false dan kondisi2 true  
}  
else {  
    dieksekusi jika kondisi1 dan kondisi2 false  
}
```

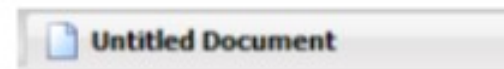
Pengkondisian

- if (kondisi1) else if (kondisi2) else (kondisi3)

Contoh

```
<script type="text/javascript">
  var jam = 17;
  if (jam <= 10)
  {
    document.write("Selamat Pagi.");
  }
  else if (jam > 10 && jam <= 15)
  {
    document.write("Selamat Siang.");
  }
  else
  {
    document.write("Selamat Sore.");
  }
</script>
```

Hasil



Selamat Sore.

Ternary

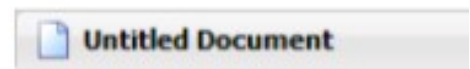
- Menggunakan tanda “?”
- Merupakan model penyederhanaan dari **if...else**
- Cocok untuk melakukan proses pengisian variabel suatu hasil pengujian.
- Sintaks

```
namaVariabel = (kondisi)? Nilai1 : Nilai2;
```

Contoh

```
<script type="text/javascript">  
  var jam = 20;  
  var pesan = "";  
  
  pesan = (jam <= 10)? "Selamat pagi." : "Selamat datang."  
  
  document.write(pesan);  
</script>
```

Hasil



Selamat datang.

Pengkondisian

- Switch
- Sintaks

```
switch (n)
{
    case 1 : { aksi 1 } break;
    case 2 : { aksi 2 } break;
    default : { aksi default }
}
```

-
- Gunakan break untuk mencegah case berikutnya tereksekusi otomatis.

Pengkondisian

- Switch

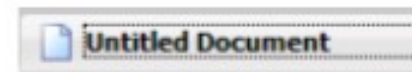
Contoh

```
<script type="text/javascript">
    var hari = 1;
    var pesan = "";

    switch (hari)
    {
        case 0 : { pesan = "Hari Minggu"; } break;
        case 1 : { pesan = "Hari Senin"; } break;
        case 2 : { pesan = "Hari Selasa"; } break;
        case 3 : { pesan = "Hari Rabu"; } break;
        case 4 : { pesan = "Hari Kamis"; } break;
        case 5 : { pesan = "Hari Jumat"; } break;
        case 6 : { pesan = "Hari Sabtu"; } break;
        default : pesan = "Hari apa?";
    }

    document.write(pesan);
</script>
```

Hasil



Hari Senin



Pengulangan

- Pada JavaScript, ada dua macam pengulangan :
 - **FOR**, pengulangan sebanyak jumlah yang ditentukan,
 - **WHILE**, pengulangan sampai bertemu kondisi “false”.
 - **DO...WHILE**, pengulangan sampai bertemu kondisi “false”.

Pengulangan - for

- Sintaks

```
for (var=startValue; var<=endValue; var=var+increment)
{
    source code
}
```

Contoh

```
<script type="text/javascript">
    var i=0;
    for (i=0;i<=5;i++)
    {
        document.write("Nomor : " + i);
        document.write("<br />");
    }
</script>
```

Hasil



Nomor : 0
Nomor : 1
Nomor : 2
Nomor : 3
Nomor : 4
Nomor : 5

Pengulangan - while

- Sintaks
- `var=startValue;`
`while (var<=endValue)`
`{`
`|source code`
`}`

Contoh

```
<script type="text/javascript">
    var i=0;
    while (i<=5)
    {
        document.write("Nomor : " + i);
        document.write("<br />");
        i++;
    }
</script>
```

Hasil



Nomor : 0
Nomor : 1
Nomor : 2
Nomor : 3
Nomor : 4
Nomor : 5

Pengulangan For...In

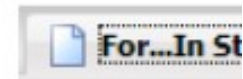
- Sintaks For...In melakukan pengulangan sebanyak elemen array atau sebanyak properti dari suatu objek.

Contoh

```
<script type="text/javascript">
    var x;
    var mycars = new Array();
    mycars[0] = "Saab";
    mycars[1] = "Volvo";
    mycars[2] = "BMW";

    for (x in mycars)
    {
        document.write(mycars[x] + "<br />");
    }
</script>
```

Hasil



Saab
Volvo
BMW

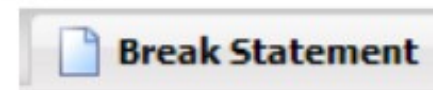
Break Statement

- Perintah break akan menghentikan pengulangan dan akan mengeksekusi sintaks berikutnya setelah pengulangan.

Contoh

```
<script type="text/javascript">
    var i=0;
    for (i=0;i<=10;i++)
    {
        if (i==3)
        {
            break;
        }
        document.write("Nomor : " + i);
        document.write("<br />");
    }
    document.write("Akhir pengulangan.");
</script>
```

Hasil



Nomor : 0
Nomor : 1
Nomor : 2
Akhir pengulangan.

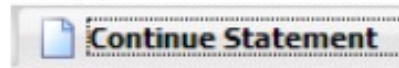
Continue Statement

- Perintah continue akan melewati value pengulangan yang ditentukan, kemudian melanjutkan ke value berikutnya.

Contoh

```
<script type="text/javascript">
  for (i=0;i<=5;i++)
  {
    if (i==3)
    {
      continue;
    }
    document.write("Nomor : " + i);
    document.write("<br />");
  }
</script>
```

Hasil



Nomor : 0
Nomor : 1
Nomor : 2
Nomor : 4
Nomor : 5



Javascript

Teknologi Web

Agung Nugroho
Universitas Pelita Bangsa Bekasi



Popup Box

JavaScript memiliki tiga macam Popup Box, yaitu:

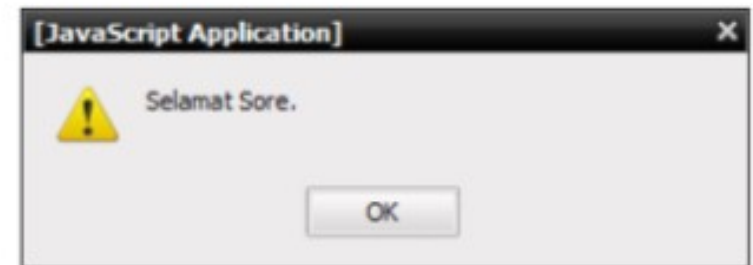
- Alert Box
 - Biasanya digunakan untuk memberikan informasi ke pengguna.
- Confirm Box
 - Biasanya digunakan apabila aplikasi membutuhkan verifikasi atau persetujuan dari pengguna.
- Prompt Box
 - Biasanya digunakan apabila aplikasi ingin pengguna memasukkan dahulu suatu nilai sebelum memasuki suatu halaman.

Popup Box - Alert

Contoh

```
<script type="text/javascript">
  var jam = 17;
  if (jam <= 10)
  {
    alert("Selamat Pagi.");
  }
  else if (jam > 10 && jam <= 15)
  {
    alert("Selamat Siang.");
  }
  else
  {
    alert("Selamat Sore.");
  }
</script>
```

Hasil

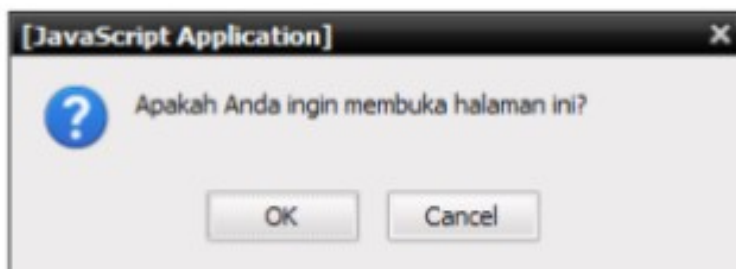


Popup Box - Confirm

Contoh

```
<script type="text/javascript">
    var konfirmasi = confirm("Apakah Anda ingin membuka halaman ini?");
    if (konfirmasi == true)
        document.location.href = "halaman1.html";
    else
        document.location.href = "home.html";
</script>
```

Hasil



`document.location.href`

Sintaks JavaScript untuk mengalihkan browser ke halaman yang dituju (*redirect*).

`var konfirmasi = confirm();`

Jika tombol "OK" yang ditekan, maka confirm akan mengisi nilai "**true**" ke variabel konfirmasi, sebaliknya "**false**" jika tombol "Cancel" yang ditekan.

Popup Box - Prompt

- Sintaks

`prompt("text", "nilai default");`

- Contoh

```
<script type="text/javascript">
    var bil1, bil2, jml;

    bil1 = prompt("Bilangan 1", "0");
    bil2 = prompt("Bilangan 2", "0");
    jml = bil1 + bil2;

    document.write("<h2>" + jml + "</h2>");
</script>
```


Popup Box - Prompt

- **Penjelasan**

```
bil1 = prompt("Bilangan 1", "0");
```

Fungsi **prompt** akan mengisi nilai yang diinput oleh pengguna ke variabel **bil1**. Tipe data yang dikeluarkan oleh **prompt** bertipe **string**.

```
jml = bil1 + bil2;
```

jml berisi nilai **"00"** karena **bil1** dan **bil2** bertipe string. Tanda plus **"+"** jika digunakan pada tipe string, maka akan menempelkan isi dari dua variabel tersebut (*concat*).

- **Konversi tipe data**

```
jml = parseInt(bil1) + parseInt(bil2);
```

jml berisi nilai **"0"** karena tipe data **bil1** dan **bil2** dikonversi menjadi integer.



Fungsi

- Sebuah fungsi berisi source code yang akan dieksekusi ketika dipanggil.
- Sebuah fungsi akan dieksekusi jika dipanggil nama fungsinya atau dieksekusi oleh suatu event.
- Sisipkan source code ke dalam fungsi untuk mencegah suatu source code langsung tereksekusi saat halaman terbuka.
- Fungsi dapat dipanggil dari mana saja di suatu halaman bahkan dari file JavaScript yang berbeda (external javascript), jika pada halaman tersebut file-file javascript sudah disisipkan.
- Fungsi bisa mengembalikan nilai ke pemanggilnya.

Fungsi

- Sintaks

```
function namaFungsi (parameter1, parameter2, ... parameterN)
{
    ...
}
```

- Contoh

```
<html>
<head>
    <script type="text/javascript">
        function displaymessage()
        {
            alert("Hello World!");
        }
    </script>
</head>

<body>
<form>
    <input type="button" value="Click me!" onclick="displaymessage()" />
</form>
</body>
</html>
```

Fungsi – Return value

- Contoh fungsi dengan return value

```
<html>
<head>
  <script type="text/javascript">
    function kali(a,b)
    {
      return a*b;
    }
  </script>
</head>

<body>
  <script type="text/javascript">
    var hasil = kali(4,3);
    document.write(hasil);
  </script>
</body>
</html>
```



Event Handler

- Event adalah aksi yang dilakukan user terhadap elemen HTML yang dapat dideteksi oleh JavaScript.
- Setiap elemen di halaman web memiliki event tertentu yang dapat memicu JavaScript.
- Event didefinisikan dalam atribut tag HTML.
- Event digunakan dalam kombinasi dengan fungsi, dan fungsi tersebut tidak akan tereksekusi sebelum event tersebut terjadi.

Event – onLoad onUnload

- OnLoad

- Event yang terjadi saat memasuki suatu halaman.
- Biasanya dipakai untuk mendeteksi browser yang dipakai.

```
<body onLoad="alert('Selamat datang di website kami.')">  
  Ini halaman web.  
</body>
```

- OnUnload

- Event yang terjadi saat keluar dari suatu halaman.
- Biasanya dipakai untuk menghapus cookie atau session.

```
<body onUnload="alert('Terima Kasih telah berkunjung.')">  
</body>
```

Event - onChange

- Event yang terjadi bila nilai suatu elemen berubah.
- Contoh

```
<html>
<head>
  <script type="text/javascript">
    function tampilKota(propinsi)
    {
      //Source code untuk tampil kota.
      alert("Tampil kota pada propinsi "+propinsi);
    }
  </script>
</head>

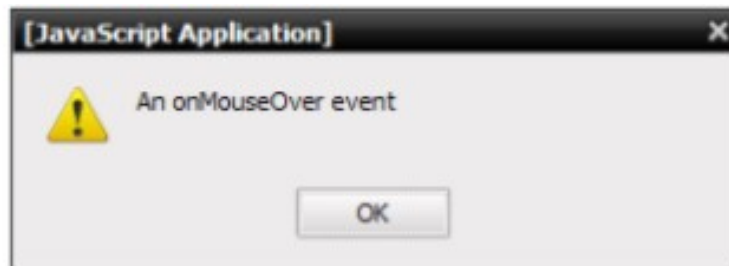
<body>
  <form>
    <select name="propinsi" onChange="tampilKota(this.value);">
      <option value="Jabar">Jawa Barat</option>
      <option value="Jateng">Jawa Tengah</option>
    </select>
  </form>
</body>
</html>
```

Event - onMouseOver

- Event yang terjadi ketika kursor berada di atas suatu elemen.
- Contoh

```
<a href="http://www.w3schools.com" onmouseover="alert('An onMouseOver event');">  
  www.w3schools.com  
</a>
```

- Hasil



Timing

- Contoh berikut ini akan menampilkan jam digital yang perpindahan detik, menit dan jam-nya realtime.



20:22:42



Timing

- Contoh (jam.html)

```
<head>
  <script type="text/javascript" src="jam.js"></script>
  <style type="text/css">
    .jam{
      font-size:24px;
      color:#0000FF;
    }
  </style>
</head>

<body onload="startTime()">
  <div id="jam" class="jam"></div>
</body>
```

Timing

- **Contoh (jam.js)**

```
function startTime()
{
    var today=new Date();
    var h=today.getHours();
    var m=today.getMinutes();
    var s=today.getSeconds();
    // Menambahkan angka 0 jika nilai <10
    h=checkTime(h);
    m=checkTime(m);
    s=checkTime(s);
    document.getElementById('jam').innerHTML=h+":"+m+": "+s;
    t=setTimeout('startTime()',500);
}

function checkTime(i)
{
    if (i<10)
    {
        i="0" + i;
    }
    return i;
}
```

```
setTimeout("namaFungsi()", delay);
```

Fungsi setTimeout akan memanggil namaFungsi() bila waktu delay sudah terpenuhi.

Satuan waktu delay : milisecond.

1000ms = 1s



Form Validation

- Validasi Form merupakan hal yang sangat penting yang harus ada dalam sebuah form input data.
- Bertujuan untuk mengatur user agar memasukkan data yang sesuai.
- Validasi dilakukan di sisi client/browser sebelum data dikirim ke server.
- Jenis validasi:
 - Field required
 - Valid email address
 - Valid date
 - Text or numeric field

Form Validasi

- Contoh validasi required field

```
function validateForm()  
{  
  var x=document.forms["myForm"]["fname"].value;  
  if (x==null || x=="")  
  {  
    alert("First name must be filled out");  
    return false;  
  }  
}
```

```
<form name="myForm" action="demo_form.asp" onsubmit="return validateForm()" method="post">  
  First name: <input type="text" name="fname">  
  <input type="submit" value="Submit">  
</form>
```



Array

- Merupakan variabel spesial yang dapat menggabungkan beberapa variable dalam satu bentuk variabel yang memiliki index
- Variable biasa:

```
var car1="Saab";  
var car2="Volvo";  
var car3="BMW";
```

- Variable Array

```
var myCars=new Array();  
myCars[0]="Saab";  
myCars[1]="Volvo";  
myCars[2]="BMW";
```



Membuat Variable Array

Ada tiga cara dalam mendefinisikan variable Array

- Regular

```
var myCars=new Array();  
myCars[0]="Saab";  
myCars[1]="Volvo";  
myCars[2]="BMW";
```

- Condensed

```
var myCars=new Array("Saab","Volvo","BMW");
```

- Literal

```
var myCars=["Saab","Volvo","BMW"];
```



Mengakses Array

- Untuk mengakses array dengan menggunakan index pada elemen array tersebut.
- Mengambil nilai dari elemen array

```
var name=myCars[0] ;
```


- Mengubah/meberikan nilai pada elemen array

```
myCars[0]="Opel" ;
```


JavaScript Objects

- Dalam javascript, semuanya disebut sebagai objek
- Memungkinkan untuk membuat objek sesuai dengan kebutuhan.
- Objek memiliki property dan method
 - Property adalah nilai yang berasosiasi dengan objek
 - Method adalah fungsi yang terdapat pada objek

A Real Life Object. A Car:

Object	Properties	Methods
	<code>car.name = Fiat</code> <code>car.model = 500</code> <code>car.weight = 850kg</code> <code>car.color = white</code>	<code>car.start()</code> <code>car.drive()</code> <code>car.brake()</code>



Membuat Objek

- Contoh Objek Person

```
person=new Object();  
person.firstname="John";  
person.lastname="Doe";  
person.age=50;  
person.eyecolor="blue";|
```



Mengakses Objek

- Mengakses Properti Objek

objectName.propertyName

contoh:

```
var message="hello world";
```

```
var x=message.length;
```

- Mengakses Method Objek

objectName.methodName()

contoh:

```
var message="hello world";
```

```
Var x=message.toUpperCase();
```



HTML DOM

Teknologi Web

Agung Nugroho
Universitas Pelita Bangsa Bekasi



HTML DOM

- DOM, singkatan dari Document Object Model, adalah sebuah cross-platform dan sebuah bahasa independen untuk merepresentasikan dan berinteraksi dengan objek dalam HTML, XHTML, dan dokumen XML.
- HTML DOM adalah cara baku untuk mengakses dan memanipulasi dokumen HTML.
- Setiap dokumen HTML yang ditampilkan pada browser akan menjadi sebuah ***Document Object***.
- Sebuah Document Object menyediakan akses ke semua elemen HTML di dalam halaman.
- Setiap elemen HTML dapat diakses dengan JavaScript atau bahasa pemrograman lain.
- Panduan lengkap mengenai HTML DOM dapat dilihat di
 - <http://www.w3schools.com/jsref/default.asp>

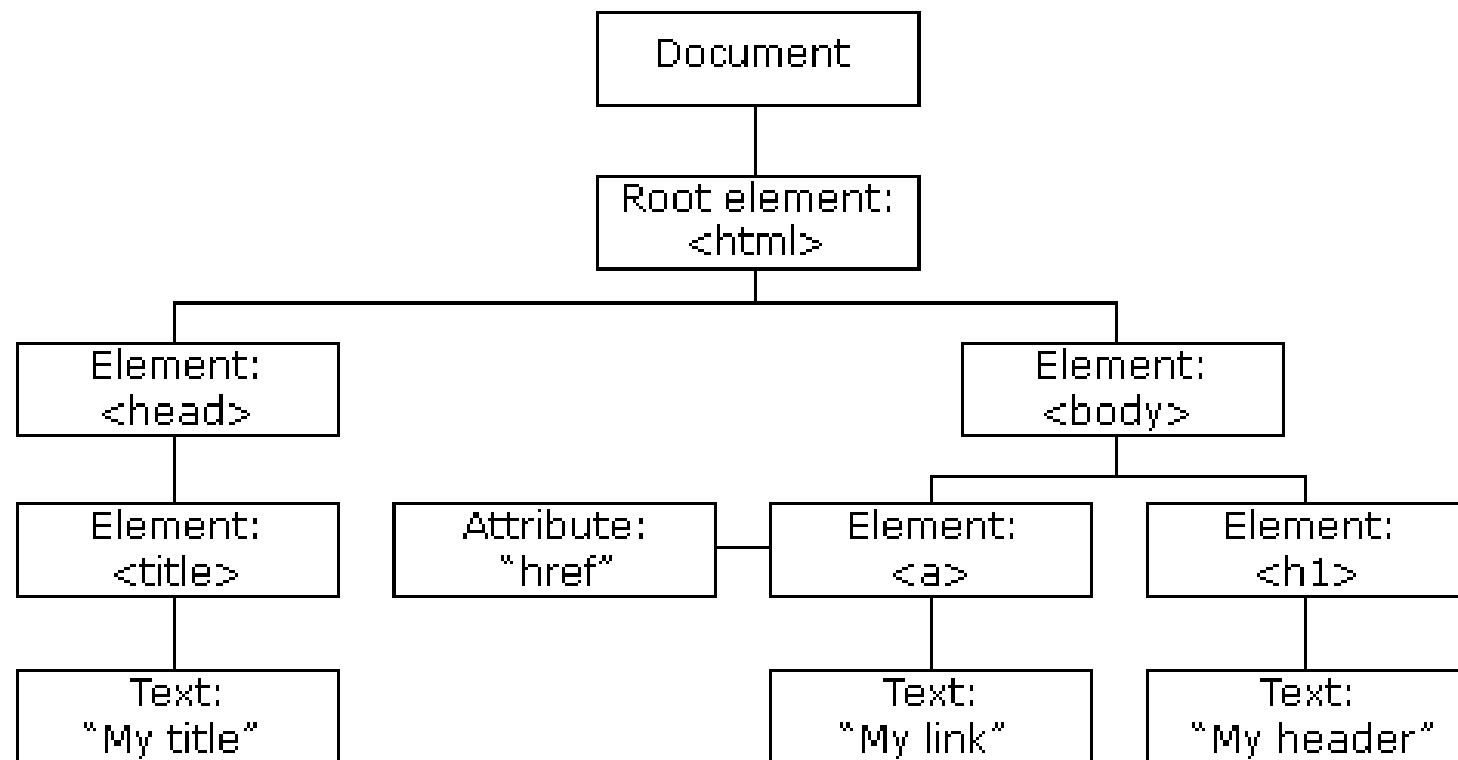


Struktur HTML DOM

- Dalam HTML DOM, semua yang berada di dokumen HTML adalah sebuah node (titik)
 - Seluruh dokumen HTML disebut document node.
 - Setiap elemen HTML disebut element node.
 - Text di dalam elemen HTML disebut text node.
 - Setiap atribut TAG HTML disebut attribute node.
 - Komentar disebut comment node.

Struktur HTML DOM

- Direpresentasikan menjadi struktur pohon sebagai berikut:





Properti

- Berikut ini beberapa properti DOM:

Properti	Fungsi
x.innerHTML	Menuliskan atau mengambil text dari elemen x.
x.nodeName	Memberi atau mengambil nama elemen x.
x.nodeValue	Memberi atau mengambil nilai dari elemen x.
x.parentNode	Mengambil informasi <i>parent</i> dari elemen x.
x.childNodes	Mengambil informasi <i>child</i> dari elemen x.
x.attributes	Memberi atau mengambil informasi atribut dari elemen x.

-



Fungsi

- Beberapa fungsi HTML DOM

Nama Fungsi	Kegunaan
<code>x.getElementById(<i>id</i>)</code>	Mengakses elemen dengan <i>ID</i> tertentu.
<code>x.getElementsByTagName(<i>name</i>)</code>	Mengakses seluruh elemen dengan <i>name</i> tertentu.
<code>x.appendChild(<i>node</i>)</code>	Menambahkan elemen (<i>child</i>) ke suatu elemen <i>x</i> .
<code>x.removeChild(<i>node</i>)</code>	Menghapus elemen (<i>child</i>) dari suatu elemen <i>x</i> .

-



Cara Akses Elemen

Mengakses suatu elemen (node) dapat dilakukan dengan 3 cara:

- Dengan menggunakan `getElementById()`
 - Contoh: `x = document.getElementById("intro");`
- Dengan menggunakan `getElementsByTagName()`
 - `x = document.getElementsByTagName("input");`
Mengakses seluruh elemen input dari dokumen HTML.
 - `x = document.getElementById("form").getElementsByTagName("input");`
Mengakses seluruh elemen input dari elemen dengan ID form.
 - `y = x[0]`
Mengakses elemen input pertama.

Cara Akses Elemen

- Dengan menelusuri struktur elemen

```
<html>
<body>
  <p id='contoh'>
    Anak
  </p>
  <script type="text/javascript">
    x = document.getElementById("contoh");
    anak = x.firstChild.nodeValue;
    document.write(anak);
  </script>
</body>
</html> |
```



HTML DOM OBJECTS

- Setiap elemen HTML yang ditampilkan di browser akan menjadi sebuah objek yang dapat dimanipulasi.
- Ada banyak macam DOM Object, namun ada 3 objek utama yaitu:
 - Document Object
 - Setiap halaman yang ditampilkan di browser akan menjadi Document Object.
 - Document object menyediakan akses ke semua elemen HTML
 - Event Object
 - Memberi informasi tentang event (aksi user) yang terjadi.
 - Merepresentasikan kondisi elemen saat diberi event, misalkan saat suatu button di klik, atau keyboard ditekan.
 - Event biasanya dikombinasikan dengan pemanggilan suatu fungsi.
 - Element Object
 - Dipergunakan untuk memanipulasi elemen HTML.



Document Object

- Document Object Collection

Koleksi	Deskripsi
anchor[]	Memberikan informasi semua <i>anchor</i> (tag a) dalam dokumen.
forms[]	Memberikan informasi semua <i>form</i> dalam dokumen.
images[]	Memberikan informasi semua <i>image</i> dalam dokumen.
links[]	Memberikan informasi semua <i>link</i> dalam dokumen.

Document Object

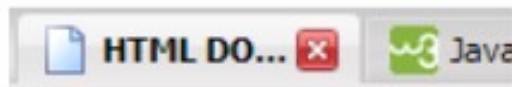
- Document Object Collections

Contoh

```
<form name="Form1"></form>
<form name="Form2"></form>
<form></form>
```

```
<script type="text/javascript">
    document.write("Jumlah form : "+document.forms.length);
    document.write("<br />");
    document.write("Nama form ke-1 : "+document.forms[0].name);
</script>
```

Hasil



Jumlah form : 3

Nama form ke-1 : Form1



Document Object

- Document Object Properties

Properti	Deskripsi
cookie	Memberikan informasi nama / nilai <i>cookie</i> pada dokumen.
documentMode	Memberikan informasi mode yang digunakan browser untuk me-render dokumen.
domain	Memberikan informasi nama domain dari server yang mengolah dokumen.
lastModified	Memberikan informasi tanggal dan waktu kapan dokumen terakhir dimodifikasi.
readyState	Memberikan informasi status <i>loading</i> dari dokumen.
referrer	Memberikan informasi URL yang memanggil halaman yang sedang ditampilkan.
title	Mengatur atau memberikan informasi <i>title</i> suatu halaman.
URL	Memberikan informasi lengkap URL dari suatu dokumen/halaman.

Document Object

- Document Object Properties

Contoh

The title of the document is:

```
<script type="text/javascript">
    document.title = "HTML DOM";
    document.write(document.title);
</script>
```

Hasil



The title of the document is: HTML DOM



Document Object

- Document Object Methods

Method	Deskripsi
open()	Membuka <i>output stream</i> untuk mengumpulkan informasi yang dikirimkan oleh document.write() atau document.writeln().
close()	Menutup <i>output stream</i> .
write()	Menuliskan text (HTML atau sintaks JavaScript) ke dokumen/halaman.
writeln()	Sama dengan write(), tapi dilanjutkan dengan pindah baris.
getElementById()	Mengakses elemen berdasarkan <i>id</i> yang ditentukan.
getElementsByName()	Mengakses elemen berdasarkan <i>name</i> yang ditentukan.
getElementsByTagName()	Mengakses elemen berdasarkan <i>tag</i> yang ditentukan.

Document Object

- Document Object Methods

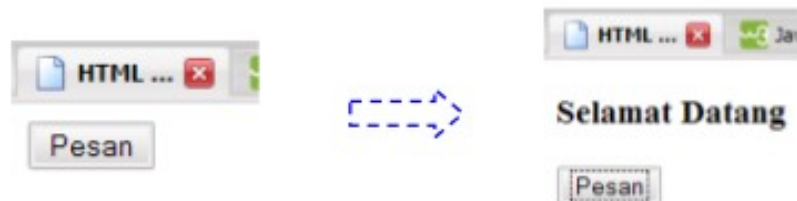
Contoh

```
<div id="pesan"></div>

<button onclick="tampilPesan()">Pesan</button>

<script type="text/javascript">
    function tampilPesan()
    {
        var pesan = document.getElementById('pesan');
        pesan.innerHTML = "<h3>Selamat Datang</h3>";
    }
</script>
```

Hasil





Event Object

- Event adalah aksi yang dapat terdeteksi oleh JavaScript.
- Dengan menggunakan Event, kita dapat membuat halaman web yang dinamis.
- Setiap elemen HTML dapat diberi event yang akan memicu suatu JavaScript. Misal event `onClick`, yang akan melakukan aksi jika suatu element di klik.
- Event dikombinasikan dengan fungsi, dan fungsi tidak akan tereksekusi jika Event yang ditentukan belum terjadi (pemanggilan fungsi).
- Macam-macam event
 - `onClick`
 - `onLoad`, `onUnload`
 - `onFocus`, `onBlur`, `onChange`
 - `onSubmit`
 - `onMouseOver`, `onMouseOut`

Event Object

Contoh (kota.js)

```
function tampilKota(propinsi)
{
    //Source code untuk tampil kota.
    var kota = "";
    switch (propinsi)
    {
        case "Jabar" : {
            kota = "<ul> \
                <li>Bandung</li> \
                <li>Bogor</li> \
                <li>Garut</li> \
            </ul>";
            } break;
        case "Jateng" : {
            kota = "<ul> \
                <li>Cilacap</li> \
                <li>Jepara</li> \
                <li>Magelang</li> \
            </ul>";
            } break;
        default : kota = "";
    }
    document.getElementById('kota').innerHTML = kota;
}
```

Event Object

Contoh (kota.html)

```
<body>
  <form>
    <select name="propinsi" onChange="tampilKota(this.value);">
      <option value=""></option>
      <option value="Jabar">Jawa Barat</option>
      <option value="Jateng">Jawa Tengah</option>
    </select>
  </form>
  <div id='kota'></div>
</body>
```



HTML Element Object

- Berikut ini beberapa proterti dari HTML Element Object yang sering dipakai

Properti	Deskripsi
accessKey	Menentukan atau memberikan informasi accesskey (shortcut) dari suatu elemen.
className	Menentukan atau memberikan informasi atribut class dari suatu elemen.
disabled	Menentukan atau memberikan informasi atribut disabled dari suatu elemen.
id	Menentukan atau memberikan informasi atribut id dari suatu elemen.
innerHTML	Menentukan atau memberikan konten HTML dari suatu elemen.
style	Menentukan atau memberikan informasi atribut style dari suatu elemen.

HTML Element Object

Contoh

```
<div id="pesan"><h1>Selamat Datang</h1></div>

<button id="buttonWarna" onclick="gantiWarna()">Hijau</button>

<script type="text/javascript">
    function gantiWarna()
    {
        var pesan = document.getElementById('pesan');
        var button = document.getElementById('buttonWarna');
        if (button.innerHTML=="Hijau")
        {
            pesan.style.cssText = "background-color: green;";
            button.innerHTML = "Merah";
        }
        else if (button.innerHTML=="Merah")
        {
            pesan.style.cssText = "background-color: red;";
            button.innerHTML = "Hijau";
        }
    }
</script>
```

HTML Element Object

Hasil



Selamat Datang

Hijau



Selamat Datang

Merah



Selamat Datang

Hijau



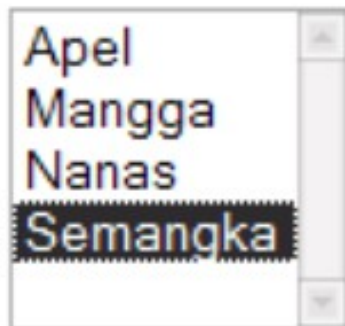
HTML DOM

- Macam-macam objek HTML DOM, diantaranya:
 - Anchor
 - Body
 - Button
 - Form
 - Frameset
 - Image
 - Input (text, password, button, checkbox, radio, hidden, submit)
 - Select
 - Option
 - Table
 - Style

Select Object

- Contoh: Menyalin text dari combo box 1 ke combo box 2.

Pilih



Apel
Mangga
Nanas
Semangka

Pilihan Anda



Mangga
Semangka

Select Object

Contoh (select.html)

```
<head>
  <script type="text/javascript" src="select.js"></script>
</head>

<body>
  <div style="float: left;">
    <p>Pilih</p>
    <select id="buah" size="5" onchange="pilih()">
      <option value="A">Apel</option>
      <option value="M">Mangga</option>
      <option value="N">Nanas</option>
      <option value="S">Semangka</option>
    </select>
  </div>

  <div style="float: left; margin-left: 20px">
    <p>Pilihan Anda</p>
    <select id="pilihan" size="5">
    </select>
  </div>
</body>
```



Select Object

Contoh (select.js)

```
function pilih()
{
    var buah = document.getElementById("buah");
    var pilihan = document.getElementById("pilihan");
    var option = document.createElement("option");

    option.text = buah.options[buah.selectedIndex].text;
    try
    {
        pilihan.add(option, null);
    }
    catch(ex)
    {
        pilihan.add(option);    // Khusus IE
    }
}
```