

# JavaScript

Sumber : Buku Pemrograman Web karangan Abdul  
Kadir & Pengantar Java Script di  
[http://www.ilmukomputer.com/2006/08/19/pengantar-  
java-script](http://www.ilmukomputer.com/2006/08/19/pengantar-java-script)

# Pengenalan JavaScript

Javascript diperkenalkan pertama kali oleh **Netscape** pada tahun 1995. Pada awalnya bahasa ini dinamakan “*LiveScript*” yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2. Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru-buru dan tidak ada pesan kesalahan yang di tampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program.

Kemudian sejalan dengan sedang giatnya kerjasama antara **Netscape** dan **Sun** (pengembang bahasa pemrograman “*Java*”) pada masa itu, maka Netscape memberikan nama “*JavaScript*” kepada bahasa tersebut pada tanggal 4 desember 1995.

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML,

Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengeksekusian perintah-perintah di sisi user, yang artinya di sisi browser bukan di sisi server web.

# Apa Perbedaan JavaScript & Java ?

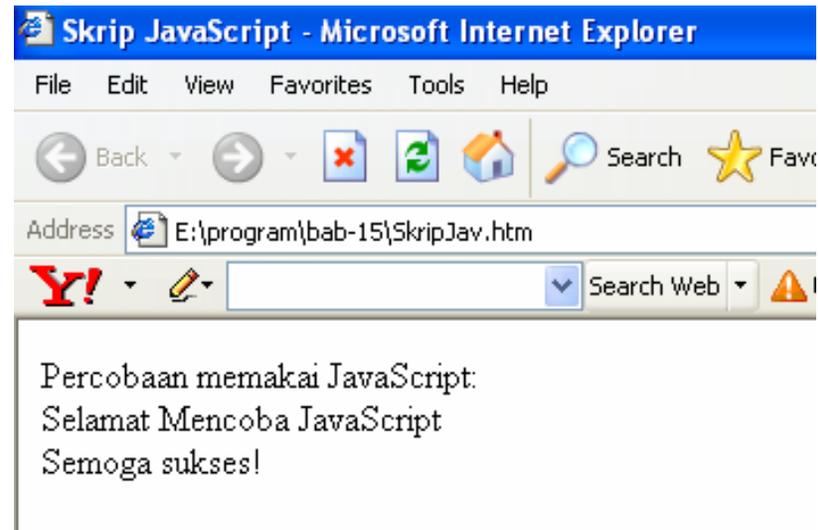
<b>JavaScript</b>	<b>Java</b>
Bahasa yang diinterpretasikan langsung oleh browser	Bahasa yang setengah terkompilasi dan memerlukan Java Virtual Machine untuk menterjemahkannya
Kode terintegrasi dengan HTML	Kode(applet) terpisah dari dokumen HTML, dipanggil pada saat membuka dokumen HTML
Bahasa dengan karakteristik yang terbatas	Bahasa dengan karakteristik yang luas (pendeklarasian jenis variabel)
Hubungan dinamis, referensi dari obyek diverifikasi pada saat loading.	Hubungan statis, obyek harus ada pada saat program di loading (di kompilasi)
Kode program bisa di akses	Kode program tersembunyi

# Bentuk skrip dari JavaScript

Skrip dari JavaScript terletak di dalam dokumen HTML. Kode tersebut tidak akan terlihat dari dalam jendela navigator anda, karena diantara tag tertentu yang memerintahkan navigator untuk memperlakukan bahwa skrip tersebut adalah skrip dari JavaScript. Contoh dari skrip yang menunjukkan bahwa skrip tersebut adalah skrip dari JavaScript adalah sebagai berikut :

```
<SCRIPT language="Javascript">  
    letakkan script anda disini  
</SCRIPT>
```

```
<HTML>  
<HEAD>  
<TITLE>Skrip JavaScript</TITLE>  
</HEAD>  
<BODY>  
Percobaan memakai JavaScript:<BR>  
<SCRIPT LANGUAGE = "JavaScript">  
<!--  
    document.write("Selamat Mencoba JavaScript<BR>");  
    document.write("Semoga sukses!");  
//-->  
</SCRIPT>  
</BODY>  
</HTML>
```



# Memberikan komentar

Agar skrip tidak dibaca sebagai skrip, akan tetapi di baca sebagai komentar dan tidak akan dieksekusi sebagai program. Contohnya adalah sebagai berikut :

```
<SCRIPT language="Javascript">  
<!--  
letakkan script anda disini  
// -->  
</SCRIPT>
```

kegunaannya antara lain adalah :

- Mengingatkan kita akan bagian-bagian khusus di dalam skrip tersebut, jika kita ingin merubah sesuatu di dalamnya.
- Membuat orang yang tidak mengerti skrip anda, menjadi mengerti dengan petunjuk-petunjuk yang anda bikin melalui komentar komentar, kecuali anda tidak mau mensharing program yang anda miliki.

# JavaScript Sebagai Bahasa Berorientasi Objek

Sebuah objek tersusun atas properti, metode, dan penanganan kejadian.

## *Properti*

- Adalah atribut dari sebuah objek.
- Contoh Objek dalam JavaScript yaitu window. Objek ini memiliki banyak properti. Salah satu diantaranya adalah defaultStatus. Properti ini menyatakan baris status yang terdapat pada jendela browser.
- Untuk mengakses sebuah properti, perlu penulisan dengan bentuk sebagai berikut: *nama\_objek.nama\_properti*,

- Contoh :  

<b>Nama objek</b>	<b>Nama properti</b>
↓	↙
windows.defaultStatus	

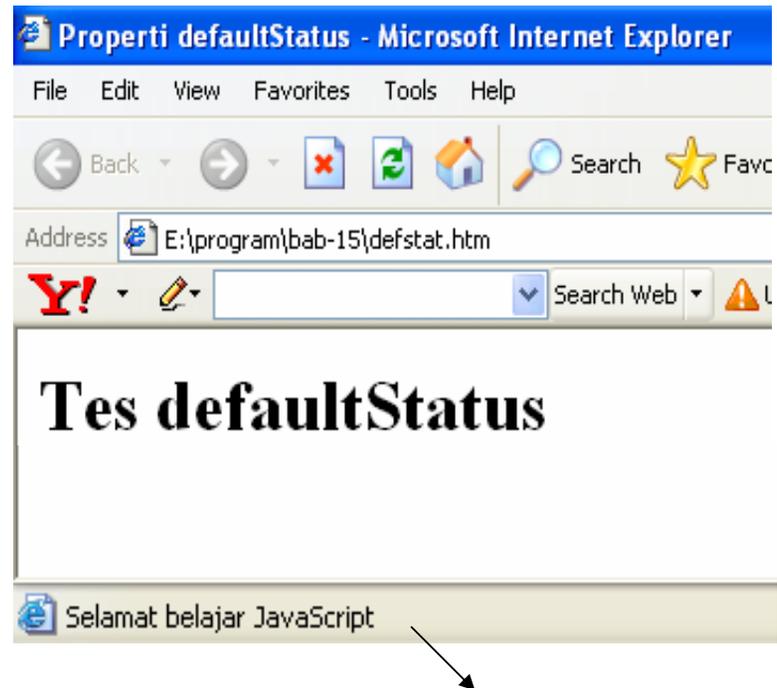
# Properti (Cont.)

Properti dapat diberi nilai melalui bentuk penugasan berikut:

Objek.properti = nilai

Contoh : `window.defaultStatus = "Selamat belajar JavaScript";`

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



Window.defaultStatus

# Metode

Adalah suatu kumpulan kode yang digunakan untk melakukan sesuatu tindakan terhadap objek.

Sebagai contoh, Write() pada objek document adalah metode yang digunakan untuk menuliskan tulisan ke jendela browser.

Contoh :

document.write("Halo")

Jika sebuah metode tidak melibatkan parameter, tanda kurung buka dan tutup tetap harus disertakan. Contoh :  
window.focus()

# Penanganan Kejadian (Event Handler)

Adalah sekumpulan kode yang akan dijalankan manakala pemakai melakukan suatu tindakan. Contoh kejadian adalah ketika pemakai meng-klik tombol mouse atau ketika menutup jendela browser.

Bentuk penanganan kejadian adalah seperti berikut:

Nama\_kejadian = “kumpulan kode”

Kumpulan kode berisi sejumlah pernyataan. Antar pernyataan dipisah dengan ;.

Contoh kejadian adalah `onMouseOver` (ketika penunjuk mouse menunjuk ke link) dan `onMouseOut` (ketika penunjuk mouse tidak lagi menyorot link).

# Contoh Penanganan Kejadian

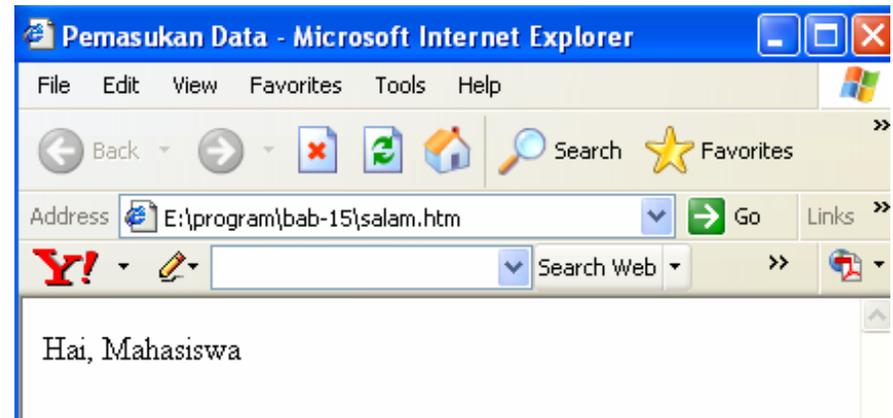
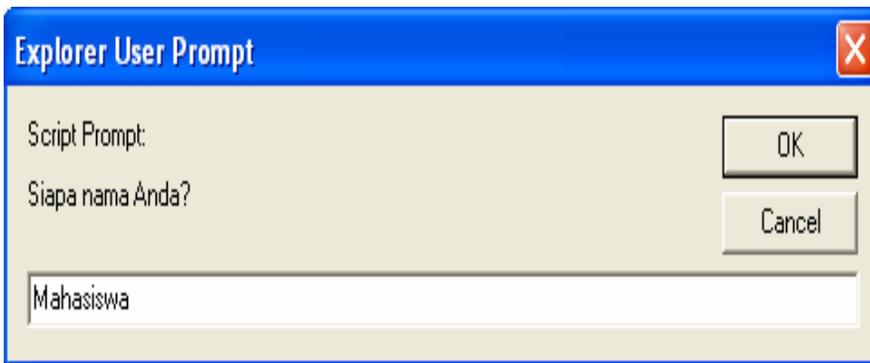
```
<HTML>
<HEAD>
<TITLE>Kejadian</TITLE>
</HEAD>
<BODY>
<H1>Tes Kejadian</H1>
<P>Cobalah meletakkan penunjuk mouse
ke link berikut
dan perhatikan isi baris status.
Kemudian pindahkan penunjuk mouse dari
link berikut dan perhatikan isi baris status
</P>

<A HREF = "www.fujitsu.com"
  onMouseOver = "window.status =
                  'Anda menyorot link, lho';
                  return true"
  onMouseOut = "window.status = ";
               return true">Fujitsu</A>
</BODY>
</HTML>
```



# Menangani Pemasukan Data

```
<HTML>  
<HEAD>  
<TITLE>Pemasukan Data</TITLE>  
</HEAD>  
<BODY>  
<SCRIPT LANGUAGE = "JavaScript">  
<!-- var nama = prompt("Siapa nama Anda?");  
    document.write("Hai, " + nama);  
//--> </SCRIPT>  
</BODY>  
</HTML>
```



# Mengenal Jendela Peringatan

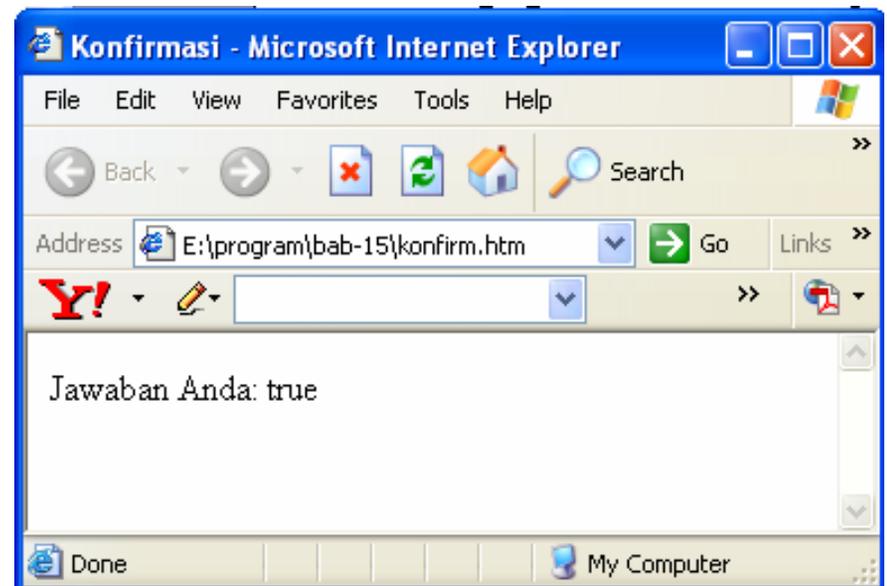
Jika karena sesuatu hal anda ingin memberikan pesan kepada pemakai, anda bisa menyajikannya dalam bentuk kotak dialog yang disebut Alert Box.

```
<HTML>
<HEAD>
<TITLE>Alert Box</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    window.alert("Ini merupakan pesan
    untuk Anda");
-->
</SCRIPT>
</BODY>
</HTML>
```



# Mengenal Jendela Konfirmasi

```
<HTML>
<HEAD>
<TITLE>Konfirmasi</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    var jawaban = window.confirm(
        "Anda ingin meneruskan?");
    document.write("Jawaban Anda: " +
jawaban);
//-->
</SCRIPT>
</BODY>
</HTML>
```

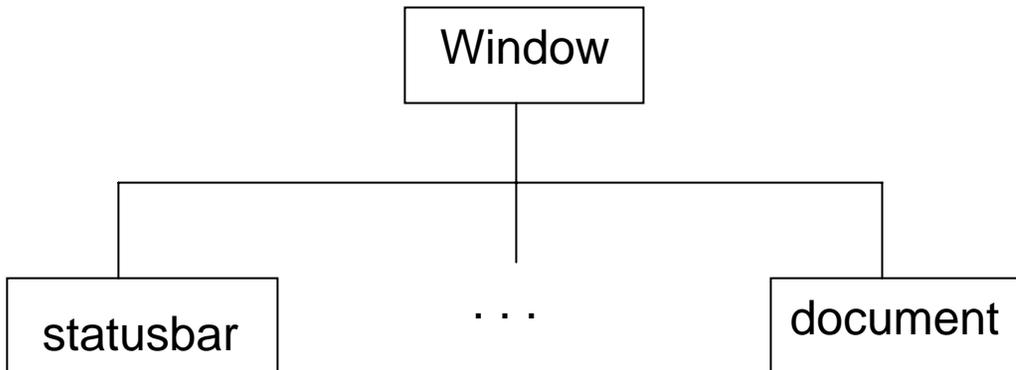


# Mengenal Hirarki Objek

Salah satu objek yang mempunyai kedudukan tertinggi dalam JavaScript adalah window. Objek ini sendiri memiliki objek lain; antara lain yaitu statusbar dan document. Dengan demikian untuk mengakses document diperlukan penulisan sbb:

Nama objek  
↓  
Window.document.write("Halo")  
↓  
metode milik objek document

Objek ini merupakan properti bagi window



# Konsep Variabel

Variable adalah suatu obyek yang berisi data data, yang mana dapat di modifikasi selama pengekseskusion program. Di JavaScript kita bisa memberikan nama variabel dengan kriteria berikut ini .

- Nama variabel harus dimulai oleh satu huruf (huruf besar maupun huruf kecil) atau satu karakter "\_".
- Nama variabel bisa terdiri dari huruf huruf, angka angka atau karakter \_ dan & (spasi kosong tidak diperbolehkan).
- Nama variabel tidak boleh memakai reserved word

Pada JavaScript kita menggunakan sistem *case sensitive* yang artinya membedakan nama variabel dengan huruf besar dan huruf kecil.

Nama variabel yang benar	Nama variabel yang tidak benar	Alasan
Variabel	Nama Dari Variabel	Ada spasi kosong
Nama_Dari_Variabel	123Nama_Dari_Variabel	Dimulai dgn angka
nama_dari_variabel	andry@yahoo.com	Karakter @
nama_dari_variabel_123	Nama-Dari-Variabel	Karakter -
_707	transient	Reserved word

# Mendeklarasikan Variabel

Deklarasi variabel di JavaScript dapat kita lakukan dengan dua cara :

- **eksplisit** : dengan menuliskan kata kunci *var* kemudian diikuti dengan nama variabel dan nilai dari variabel : `var test = "halo"`
- **implisit** : dengan menuliskan secara langsung nama dari variabel dan diikuti nilai dari variabel : `test = "halo"`

navigator secara otomatis akan memperlakukan pernyataan itu sebagai deklarasi dari sebuah variabel.

Pengaksesan variabel bergantung lokasi dimana dia dideklarasikan :

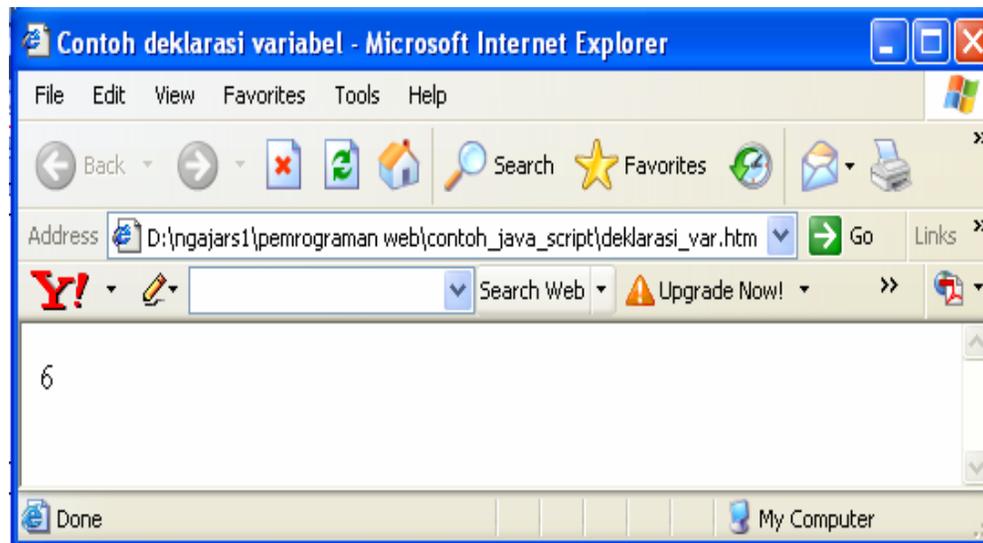
- Jika dia dideklarasikan dibagian awal dari skrip program, yang artinya sebelum pendeklarasian semua fungsi, maka semua fungsi di dalam program bisa mengakses variabel ini, dan variabel ini menjadi **variabel global**.
- Jika dia dideklarasikan dengan menggunakan kata kunci *var* di dalam suatu fungsi tertentu, maka variabel itu hanya bisa di akses dari dalam fungsi tersebut, dan artinya variabel ini tidak berguna bagi fungsi fungsi yang lain, dan kita sebut variabel ini menjadi **variabel lokal**

*Untuk penjelasan tipe data silahkan anda baca sendiri dalam tutorial JavaScript atau buku mengenai JavaScript.*

# Contoh Mendeklarasikan Variabel

```
<HTML>
<HEAD>
<TITLE> Contoh deklarasi variabel
</TITLE>
</HEAD>
<BODY>
<SCRIPT language="Javascript">
<!--
var VariabelKu;
var VariabelKu2 = 3;
VariabelKu = 2;

document.write(VariabelKu*VariabelKu2);
// -->
</SCRIPT>
</BODY>
</HTML>
```



# Contoh Variabel lokal & Global

```
<HTML>
<HEAD>
<TITLE> Contoh variabel Lokal dan global</TITLE>
</HEAD>
<BODY>
<SCRIPT language="Javascript">
<!--
var a = 12;
var b = 4;

function PerkalianDengan2(b)
{ var a = b * 2; return a; }

document.write("Dua kali dari ",b," adalah
",PerkalianDengan2(b));
document.write(" Nilai dari a adalah ",a);
// -->
</SCRIPT>
</BODY>
</HTML>
```



# Konversi jenis variabel

Meskipun JavaScript memungkinkan pengaturan perubahan jenis variabel secara transparan, kadang kadang kita perlu juga untuk melakukan konversi jenis variabel secara paksa. Ada 2 fungsi dasar yang memungkinkan merubah jenis variabel yang dilewatkan dengan parameter tertentu :

`parseInt()`

Fungsi ini mungkinkan merubah satu variabel yang dilewatkan dengan parameter tertentu menjadi bilangan bulat. Sintaksnya adalah sebagai berikut :

`parseInt(string[, basis]);`

Berikut ini salah satu contoh penggunaan fungsi `parseInt()` :

```
var a = "123";   var b = "456";  
document.write(a+b,"<BR>"); // hasil 123456  
document.write(parseInt(a)+parseInt(b),"<BR>"); // hasil 579
```

# Konversi jenis variabel

## parseFloat()

Adalah satu fungsi inti dari JavaScript yang memungkinkan merubah variabel yang dilewatkan dengan parameter tertentu menjadi bilangan desimal, Sintaks dari fungsi parseFloat adalah sebagai berikut :

`parseFloat(string);`

tabel contoh tentang penggunaan fungsi parseFloat()

Contoh	Hasil
<code>parseFloat("128.34");</code>	128.34
<code>parseFloat("128,34");</code>	128
<code>parseFloat("12.3E-6");</code>	0.0000123
<code>parseFloat("Halo");</code>	NaN
<code>parseFloat("24.568Halo38");</code>	24.568
<code>parseFloat("Halo38.24");</code>	NaN
<code>parseFloat("AF8BEF");</code>	NaN
<code>parseFloat("0284");</code>	284
<code>parseFloat("0xAB882F");</code>	11241519

# Struktur Kondisional

Struktur kondisional adalah instruksi instruksi yang memungkinkan kita untuk melakukan test apakah satu kondisi adalah benar atau tidak, dan memungkinkan juga terjadinya proses iterasi di dalam skrip yang kita buat

## Pernyataan if

Instruksi ini memungkinkan kita untuk mengeksekusi satu blok instruksi jika kondisi syaratnya terpenuhi. Sintaks dari instruksi ini adalah sebagai berikut :

```
if (kondisi syarat terpenuhi)  
    { daftar instruksi atau blok instruksi }
```

Beberapa catatan penting tentang instruksi ini

- Kondisi harus terletak diantara dua tanda kurung
- Kita bisa meletakkan beberapa kondisi dengan menggunakan operator AND atau OR (&& atau ||)

```
contoh : if ((kondisi1)&&(kondisi2))
```

```
if ((kondisi1)||(kondisi2))
```

# Struktur Kondisional

## Pernyataan if..else

Instruksi If, adalah instruksi dasar yang hanya memungkinkan kita untuk melakukan pemeriksaan apakah satu kondisi terpenuhi atau tidak. Pada kenyataannya kita menginginkan lebih dari satu kondisi syarat untuk menjalankan program, untuk kebutuhan itu, kita bisa menggunakan instruksi If ...Else....

Sintaks lengkap dari instruksi ini adalah :

```
if (kondisi syarat1 terpenuhi) {  
    daftar instruksi atau blok instruksi  
}  
else {  
    daftar instruksi/blok instruksi yang lain  
}
```

# Struktur Kondisional

## Pernyataan While

Sintaks dari instruksi ini adalah sebagai berikut :

```
while (kondisi syarat terpenuhi) {  
    daftar instruksi instruksi atau blok instruksi  
}
```

## Pernyataan do . . While

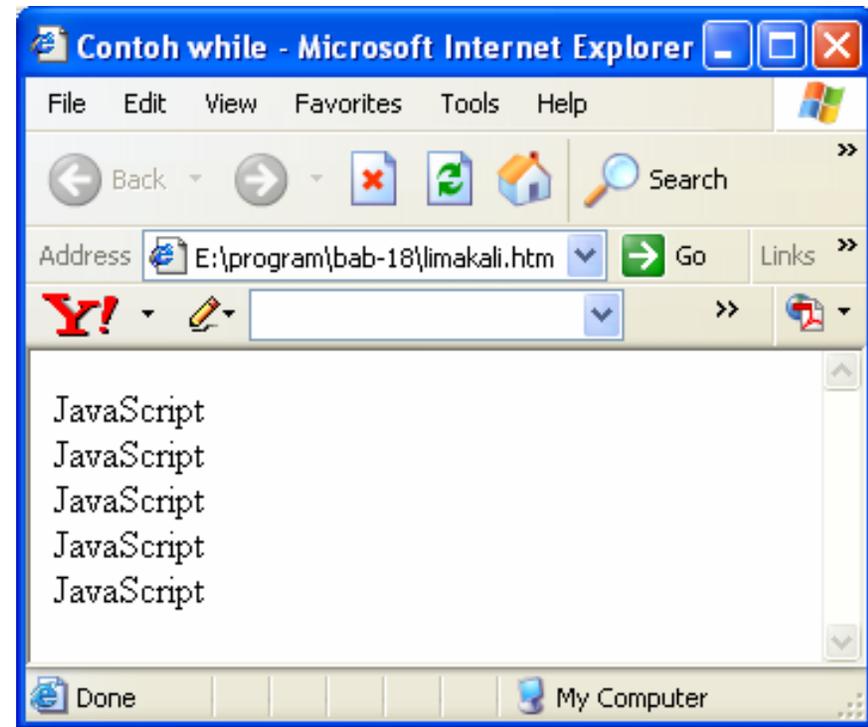
Bentuk pernyataan do . .while adalah sbb:

```
do {  
    blok pernyataan  
} while (kondisi);
```

# Contoh : pernyataan while

```
<HTML>
<HEAD>
<TITLE>Contoh while</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    var bilangan = 0;
    while (bilangan < 5) {

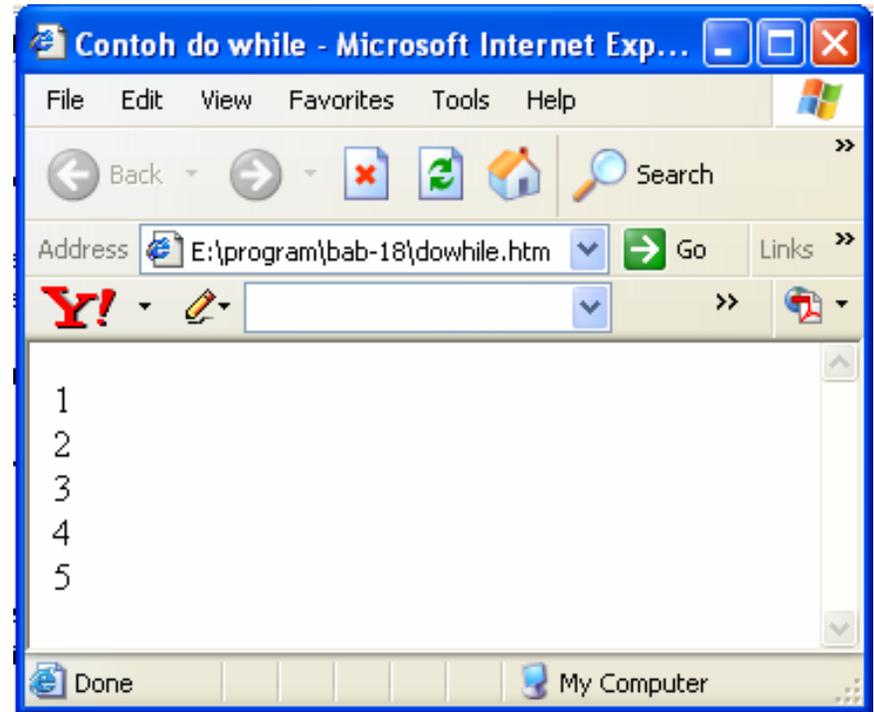
document.write(" JavaScript<BR>");
        bilangan++;
    }
-->
</SCRIPT>
</BODY>
</HTML>
```



# Contoh : pernyataan do . . while

```
<HTML>
<HEAD>
<TITLE>Contoh do while</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    var bilangan = 1;
    do {
        document.write(bilangan
            "<BR>");
        bilangan++;
    } while (bilangan < 6);
//-->
</SCRIPT>
</BODY>
</HTML>
```

+



# Pernyataan For

For adalah salah satu Instruksi yang menggunakan fasilitas *Loop*. Dalam sintaksnya kita hanya perlu memasukkan nama variabel sebagai penghitung (dan juga nilai awalnya, serta kondisi dimana loop akan berhenti (pada dasarnya, kondisi dimana nilai penghitung melewati angka tertentu)), dan yang terakhir instruksi modifikasi penghitung, increment (naik per unit) atau decrement (turun per unit)

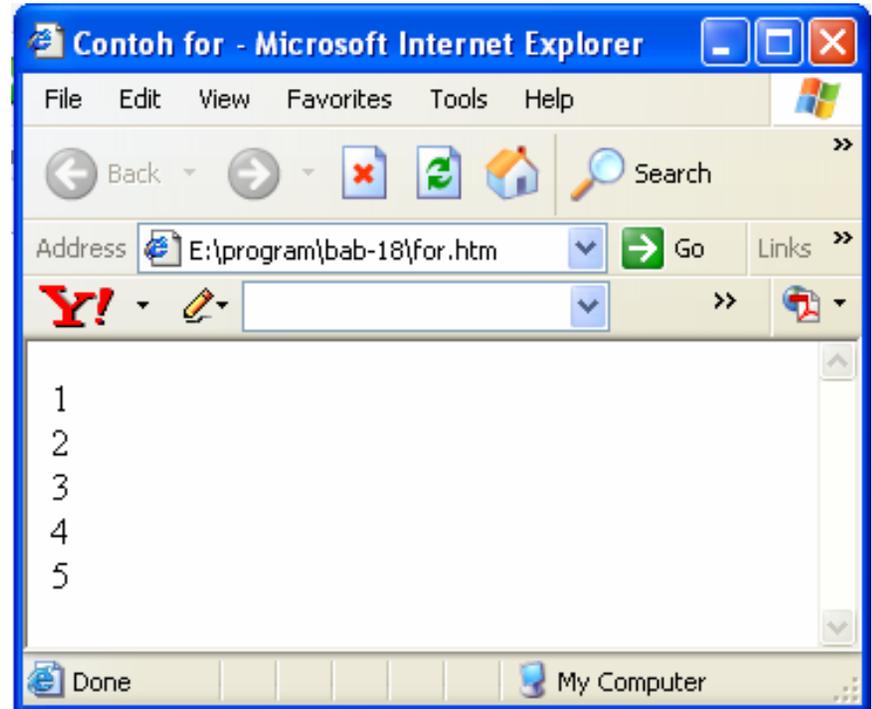
Sintaks lengkap dari instruksi ini adalah :

```
for (penghitung; kondisi loop berhenti; modifikasi
penghitung) {
daftar instruksi-instruksi atau blok instruksi
}
```

# Contoh : Pernyataan For

```
<HTML>
<HEAD>
<TITLE>Contoh for</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    var bilangan = 0;

    for (bilangan = 1; bilangan <= 5;
bilangan++)
        document.write(bilangan + "<BR>");
//-->
</SCRIPT>
</BODY>
</HTML>
```



## Pernyataan Break

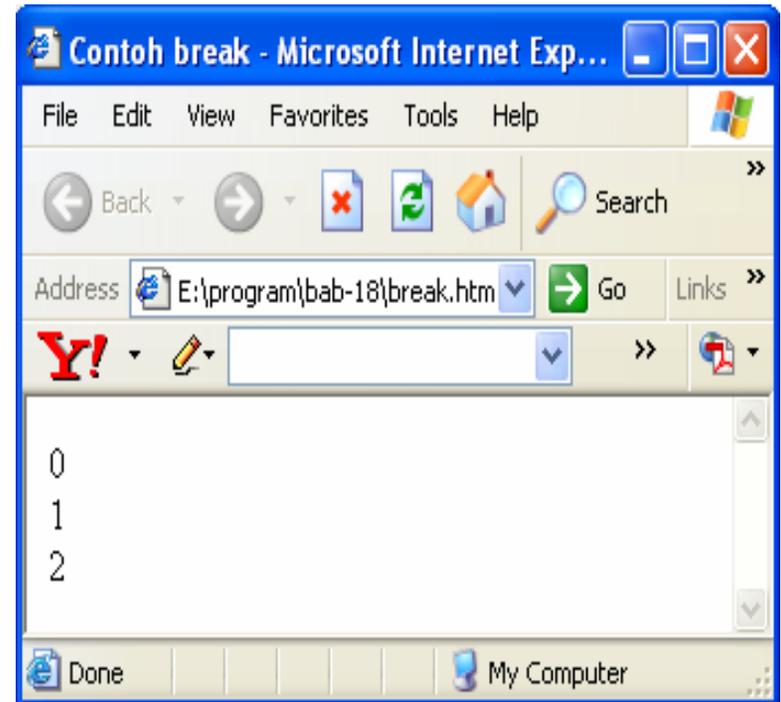
Sebaliknya kita juga bisa memaksa loop berhenti sebelum waktunya dengan alasan yang dikemukakan di bagian awal dari loop. Instruksi Break memungkinkan menghentikan suatu loop (baik untuk for ataupun while). Pemakaiannya sendiri seperti instruksi continue, yaitu penambahan struktur kondisional agar supaya loop berhenti dan tidak berulang ulang looping.

## Pernyataan Continue

Ada kalanya kita perlu melakukan lompatan (jump) terhadap satu atau beberapa nilai tertentu di dalam loop tanpa menghentikan loop itu sendiri. Sintaks yang digunakan disini adalah *continue*, dan di letakkan di dalam loop itu sendiri, pada umumnya kita tambahkan juga struktur kondisional sebagai syarat supaya sintaks tersebut berjalan lancar.

# Contoh Break

```
<HTML>
<HEAD>
<TITLE>Contoh break</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    var bilangan = 0;
    while (bilangan < 5) {
        if (bilangan == 3)
            break;
        document.write(bilangan + "<BR>");
        bilangan++;
    }
//-->
</SCRIPT>
</BODY>
</HTML>
```



# Contoh Continue

```
<SCRIPT LANGUAGE = "JavaScript">
<!--
  var i = 0; // Contoh dengan while
  while (i < 5) {
    if (i == 3)
    { i++;
      continue; }
    document.write(i + "<BR>");
    i++; }
  document.write("<HR>");
  // contoh dengan for
  for (i = 0; i < 5; i++) {
    if (i == 3)
      continue;
    document.write(i + "<BR>"); }
  //-->
</SCRIPT>
```

