

# Belajar Java Script

[taryanarx@yahoo.com](mailto:taryanarx@yahoo.com) – <http://kuliahonline.unikom.ac.id>

## **Target Pencapaian**

*Siswa mengetahui dasar-dasar perintah dalam java script dan mengaplikasikannya dalam program*

## 1. Java Script

Java Script adalah bahasa script yang berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML. Dimana objek tersebut dapat berupa suatu window, frame, URL, dokumen, form, button atau item yang lain. Yang semuanya itu mempunyai properti yang saling berhubungan dengannya dan masing-masing memiliki nama, lokasi, warna nilai dan atribut lain.

## 2. Menjalankan Java Script

Untuk dapat mempelajari pemrograman Java Script, ada dua piranti yang diperlukan yaitu browser dan teks editor. Teks editor adalah sebuah pengolah kata (*word processor*) yang menghasilkan file dalam format ASCII murni. Bila Anda adalah pengguna Windows, Anda bisa menggunakan Notepad, Wordpad atau menggunakan Ultraedit Text Editor. Selain itu browser web yang akan anda gunakan harus mendukung Java Script, Anda dapat menggunakan Internet Explorer, Opera, FireFox dan .

Kode program JavaScript dapat dituliskan langsung pada file HTML dengan menggunakan tag kontainer <SCRIPT>. Dengan kata lain, Anda tidak perlu menuliskan program JavaScript pada file terpisah.

Ingat bahwa yang dimaksud dengan tag kontainer adalah tag yang diawali dengan <NAMA\_TAG> dan diakhiri dengan </NAMA\_TAG>.

Contoh.

```
<HTML></HTML>
<BODY></BODY>
<HEAD></HEAD>
```

Tag kontainer <SCRIPT> mempunyai dua atribut tetapi yang harus Anda isikan hanya satu atribut yaitu Language. Isilah atribut language dengan “JavaScript”. Hal ini digunakan untuk memberitahukan pada browser bahwa yang akan Anda tulis adalah JavaScript.

Contoh

```
<SCRIPT LANGUAGE=”JavaScript”>
    //program Anda masukan disini
```

</SCRIPT>

### 3. Program Pertama Java Script

Pada bagian ini, Anda akan membuat program untuk menampilkan pesan sederhana kelayar monitor.



*Gambar 1. Menampilkan Teks*

```
<html>
<head>
<title>Program Pertama</title>
</head>
<script language="JavaScript">
document.writeln("<PRE>");
document.writeln("SELAMAT DATANG DI DUNIA JAVASCRIPT ");
document.writeln("Program ini merupakan contoh sederhana menampilkan teks");
document.writeln("Dengan menggunakan JavaScript");
document.writeln("</PRE>");
</script>
</body>
</html>
```

Objek document mempunyai dua metode untuk menuliskan teks, yaitu write dan writeln. Mungkin bagi Anda yang sudah terbiasa atau pernah berkenalan dengan Pascal sudah mengerti perbedaan kedua statement ini, yaitu Metode write digunakan untuk menulis teks tanpa ganti baris (*carriage return*) sedangkan Metode writeln digunakan untuk menulis teks dengan ganti baris.



Berikut ini adalah contoh bagaimana menuliskan sebuah teks dengan menggunakan tag header di tengah jendela browser. Untuk Menampilkan teks header Anda cukup menambahkan statement `<H1> Header pertama </H1>`



*Gambar 2. Teks Header Pada JavaScript*

```
<html>
<script language="JavaScript">
document.writeln("<CENTER>");
document.writeln("<H1>UNIVERSITAS KOMPUTER INDONESIA </H1>");
document.writeln("<H2>JL.DIPATIUUR 112 BANDUNG</H2>");
document.writeln("</CENTER>");
</script>
</html>
```

## 4. Komentar

Sama seperti bahasa pemrograman yang lainnya, JavaScript juga menyediakan fasilitas untuk menuliskan komentar, komentar ini berguna bila nantinya Anda atau orang lain membaca suatu program.

Pemberian komentar dalam JavaScript dapat dilakukan dengan dua cara yaitu dengan menuliskan komentar setelah tanda garis miring dua kali, contoh:

```
// ini komentar
atau
/* ini komentar juga */
```

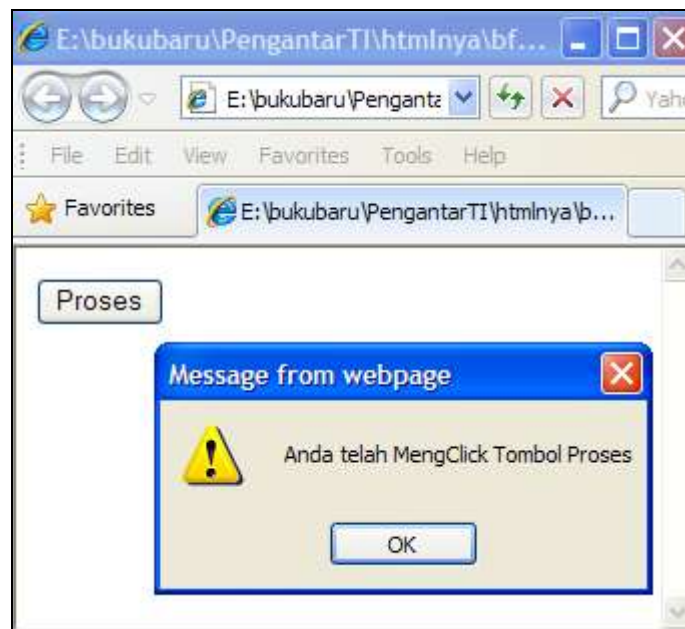
## 5. Event

Event dan event handler merupakan hal yang sangat penting dalam pemrograman JavaScript. Event adalah sesuatu yang terjadi karena aksi user. Contohnya jika user men-click tombol mouse maka akan terjadi event `Click`. Jika `MousePointer` bergerak melewati sebuah link terjadilah event

MouseOver. Ada banyak event yang terdapat dalam JavaScript yang selengkapnya bisa anda lihat *JavaScript Reference*.

Tentunya kita ingin program JavaScript kita bereaksi jika terjadi suatu event tertentu. Ini bisa dilakukan dengan bantuan event-handlers. Sebagai contoh kita memiliki sebuah tombol yang akan mengeluarkan sebuah popup window jika di-tekan. Ini berarti bahwa sebuah window popup harus muncul sebagai reaksi dari event `Click`. Event-handler yang harus kita gunakan adalah `onClick`, yang memberitahukan apa yang harus dikerjakan oleh komputer jika event ini terjadi. Contoh di bawah ini menunjukkan bagaimana mudahnya menggunakan event-handler `onClick`:

```
<form>
<input type="button" value="Proses" onClick="alert('Anda telah MengClick Tombol Proses')">
</form>
```



Gambar 3. Penggunaan Event

Ada beberapa hal baru dalam kode di atas, mari kita lihat satu per satu. Kita telah membuat form dan sebuah tombol (ini merupakan bahasa HTML standard). Bagian yang baru adalah `onClick="alert('Anda telah MengClick Tombol Proses')"` yang berada di dalam tag `<input>`. Inilah yang mendefinisikan apa yang akan terjadi jika tombol ditekan. Jadi jika terjadi event `Click`, komputer akan mengeksekusi `alert('Anda telah MengClick Tombol Proses')`, yang merupakan kode JavaScript.

`alert()` berfungsi untuk menampilkan window popup. Di dalam tanda kurung anda dapat menentukan string yang akan muncul pada window yang dimaksud.

Satu hal yang mungkin membingungkan: pada perintah `document.write()` kita menggunakan double quotes (") dan pada `alert()` kita menggunakan juga single quotes ('), mengapa? Pada contoh ini kita menuliskan `onClick="alert('Ya')"`, kita gunakan keduanya (double dan single quote). Jika kita tulis `onClick="alert("Ya")"` maka komputer akan bingung mana yang menjadi bagian event-handler `onClick` dan mana yang bukan. Urutan penggunaannya tidak menjadi masalah. Jadi bisa juga kita tuliskan `onClick='alert("Ya")'`.

Ada banyak event-handler lain yang bisa kita gunakan. Selengkapnya lihat pada referensi JavaScript.

## 6. Function

Pada dasarnya function merupakan suatu cara yang dapat anda gunakan untuk menyatukan beberapa perintah. Mari kita tulis sebuah script yang mengeluarkan teks tertentu sebanyak tiga kali. Perhatikan contoh berikut ini:

```
<html>
<script language="JavaScript">
  document.write("Belajar Java Script<br>");
  document.write("Belajar Java Script<br>");
  document.write("Belajar Java Script<br>");
  document.write("Belajar Java Script<br>");
</script>
</html>
```

Hasil keluarannya adalah:

**Belajar Java Script**

Sebanyak empat kali. Perhatikan baris kode – menuliskan kode sebanyak empat kali akan memberikan hasil yang diinginkan.

Selain dengan cara manual seperti diatas, ada cara yang lebih efisien untuk menampilkan sesuatu yang berulang, yaitu dengan menggunakan Function seperti pada kode di bawah ini:

```
<html>
<script language="JavaScript">
function myFunction() {
  document.write("Belajar Java Script<br>");
}
myFunction();
myFunction();
myFunction();
```

```
myFunction();  
</script>  
</html>
```

Pada script di atas kita definisikan sebuah function yang dilakukan melalui baris-baris:

```
function myFunction() {  
    document.write("Belajar Java Script<br>");  
}
```

Perintah-perintah di dalam tanda {} merupakan milik function `myFunction()`. Ini berarti ada sebuah perintah `document.write()` yang dapat dieksekusi melalui pemanggilan function yang dimaksud. Pada contoh kita memanggil function ini sebanyak empat kali dan berarti bahwa function akan dieksekusi sebanyak empat kali.

Functions dapat pula dikombinasikan dengan event-handler seperti pada contoh berikut ini:

```
<html>  
<head>  
<script language="JavaScript">  
function calculation() {  
    var x = 12;  
    var y = 5;  
    var result = x + y;  
    alert(result);  
}  
</script>  
</head>  
<body>  
  
    <form>  
    <input type="button" value="Calculate" onClick="calculation()">  
    </form>  
</body>  
</html>
```

Tombol akan memanggil function `calculation()` jika di-click. Kita lihat bahwa function melakukan perhitungan tertentu sehingga kita perlu menggunakan variabel `x`, `y` dan `result`. Kita mendefinisikan variabel dengan keyword `var`. Variables dapat digunakan untuk menyimpan harga-harga yang berbeda- seperti angka, text, strings dan lainnya. Baris `var result= x + y;` memberitahu browser untuk membuat variabel `result` dan menyimpan harga `x + y` (`5 + 12`) dalam variabel `result`. Setelah operasi ini isi variabel `result` adalah 17. Perintah `alert(result)` artinya sama dengan `alert(17)`, yaitu popup window akan muncul dengan isi angka 17.

## 7. Variabel dalam Java Script

---

Dalam hampir setiap bahasa pemrograman ada yang namanya variable. Variabel berguna untuk menyimpan data. Tanpa bantuan variabel sebuah program tidak bisa melakukan apa yang Anda inginkan.

Dalam Java Script pendeklarasian sebuah variabel sifatnya opsional, artinya Anda boleh mendeklarasikan atau tidak, Jika Anda memberi nilai pada suatu variabel, maka dalam JavaScript dianggap bahwa Anda telah mendeklarasikan variabel tersebut.

Contoh:

Nama = "Fahra Ragita Musyafa"

X = 2010

Y = 121299

Untuk mendeklarasikan secara eksplisit, tuliskan variabel tersebut dengan didahului kata kunci var.

Contoh

```
var nama;
```

```
var nama="Mirawati Randani"
```

```
var X = 2010;
```

```
var Y;
```

Untuk mendeklarasikan beberapa variabel dalam satu baris, dapat dilakukan dengan menuliskan seperti berikut:

```
var A,B,C;
```

## 8. Tipe Data

---

Tidak seperti bahasa pemrograman lainnya, JavaScript tidak mempunyai tipe data secara eksplisit. Hal ini dapat dilihat dari beberapa contoh yang telah disajikan sebelumnya. Anda mendeklarasikan variabel tapi tidak menentukan tipenya.

Meskipun JavaScript tidak mempunyai tipe data eksplisit, JavaScript mempunyai tipe data Implisit. Ada empat macam tipe data implisit yang dimiliki JavaScript yaitu:

- Numerik, seperti 2351978, 2003948, 3.14, 100 dsb
- String, seperti "Halo". "Mei", "Juli", "Ah Kamu", "JavaScript" dsb
- Boolean, hanya bernilai true atau false
- Null, yaitu variabel yang tidak diinisialisasi

### 8.1. Tipe Numerik



Pada dasarnya JavaScript hanya mengenal dua macam tipe numerik, yaitu bilangan bulat dan bilangan real.

Untuk bilangan bulat, Anda bisa merepresentasikan dengan basis desimal, oktal, atau heksadesimal.

Contoh:

```
var A = 100;
```

```
var A=0x2F;
```

Untuk pendeklarasian tipe bilangan real, Anda bisa menggunakan tanda titik atau notasi ilmiah (notasi E).

Contoh:

```
var a = 123.567
```

```
var b = 1.234567E+3
```

## 8.2. Tipe String

Untuk mendeklarasikan tipe string dapat dilakukan dengan cara menuliskan string diantara tanda tunggal (') atau tanda petik ganda (");

Contoh:

```
var A = 'Ini pendeklarasian String';
```

```
var C = "Ini juga string";
```

## 8.3. Tipe Boolean

Tipe Boolean hanya mempunyai nilai True atau false. Tipe ini biasanya digunakan untuk mengecek suatu kondisi atau keadaan.

Contoh

```
Var X = (Y>90);
```

Pada contoh ini menunjukan bahwa jika Y lebih besar dari 90 maka X akan bernilai True.

## 8.4. Tipe Null

Tipe Null digunakan untuk merepresentasikan variabel yang tidak diberi nilai awal (inisialisasi)

## 9. Operator

Operator pada JavaScript dibagi menjadi enam, yaitu

- Aritmetik
- Pemberian nilai (Assign)
- Pemanipulasian bit (bitwise)
- Pembandingan
- Logika
- String

### 9.1. Operator Aritmetik



Operator ini digunakan untuk operan bertipe numerik, Ada dua macam operator aritmetik, yaitu operator numerik tunggal dan operator aritmetik biner. Perbedaan kedua operator ini terletak pada jumlah operan yang harus dioperasikan.

Operator	Tunggal/Biner	Keterangan
+	Biner	Penjumlahan
-	Biner	Pengurangan
*	Biner	Perkalian
/	Biner	Pembagian
%	Biner	Modulus
-	Tunggal	Negasi
++	Tunggal	Penambahan dengan Satu
--	Tunggal	Pengurangan dengan Satu

## 9.2. Operator Pemberian Nilai

Digunakan untuk memberi nilai ke suatu operan atau mengubah nilai suatu operan.

Operator	Keterangan	Contoh	Exuivalen
=	Sama dengan	X=Y	
+=	Ditambah dengan	X+=Y	X=X+Y
-=	Dikurangi dengan	X-=Y	X=X-Y
*=	Dikalikan dengan	X*=Y	X=X*Y
/=	Dibagi dengan	X/=Y	X=X/Y
%=	Modulus dengan	X%=Y	X=X%Y
&=	Bit AND dengan	X&=Y	X=X&Y
=	Bit OR dengan	X =Y	X=X Y

## 9.3. Operasi Pemanipulasi Bit

Opeasi ini berhubungan dengan pemanipulasian bit pada operan bertipe bilangan bulat.

Operator	Keterangan
&	Bit AND
	Bit OR
^	Bit XOR
~	Bit NOT
<<	Geser ke kiri
>>	Geser ke kanan
>>>	Geser kekanan dengan diisi nol

Contoh:

```
var A = 12; // A = 1100b
var B = 10; // B = 1010b
var C = A & B
    maka akan dihasilkan bilangan seperti berikut:
    1100b
    1010b AND
    1000b
```

```
var A = 12;
var C = A << 2;
var D = A >> 1
maka variabel C akan bernilai 48 (0011 0000b)
variabel D akan bernilai 6 (0110b).
```

## 9.4. Operator Pembandingan

Digunakan untuk membandingkan dua buah operan. Operan yang dikenai operator ini dapat bertipe string, numerik, maupun ekspresi lain.

Operator	Keterangan
==	Sama dengan
!=	Tidak sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

## 9.5. Operator Logika

Sesuai namanya operator ini digunakan untuk mengoperasikan operan yang bertipe boolean,

Operator	Keterangan
&&	Operator logika AND
	Operator logika OR
!	Operator logika NOT

Contoh:

```
var A = true;
var B = false;
var C = A && B; //false
var D = A || B; //true
var E = !A; // false
```

## 9.6. Operator String

Selain operator perbandingan, operan bertipe string pada JavaScript juga mengenal satu operator lagi yang dinamakan penggabungan. Operator ini digunakan untuk menggabungkan beberapa string menjadi sebuah string yang lebih panjang.

Contoh:

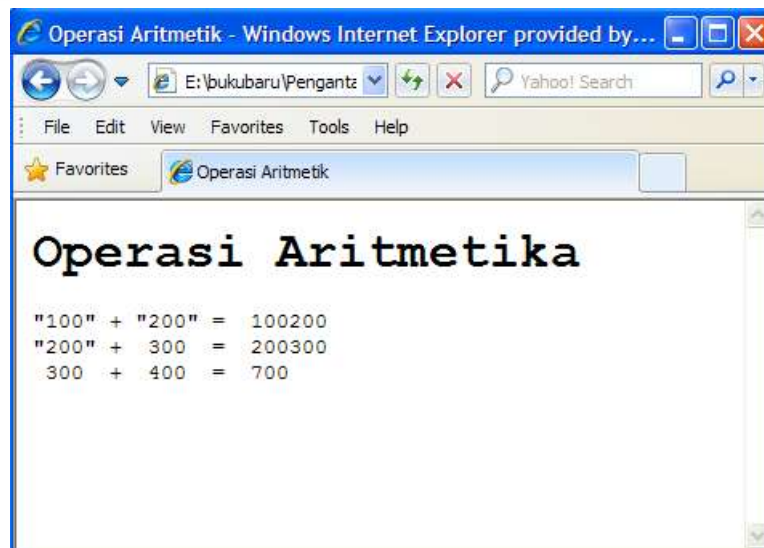
Nama = "Java" + Script";

Akan menghasilkan "JavaScript" pada variabel Nama

### Listing Program Java Script

```
<html>
<head>
<title>Operasi Aritmetik</title>
</head>
<p>
<script language="JavaScript">
document.writeln("<PRE>");
document.writeln("<h1>Operasi Aritmetika </h1>");
var A = "100";
var B = "200";
var C = 300;
var D = 400;
var E = A + B;
document.writeln('"100" + "200" = ' + E);
E = B + C;
document.writeln('"200" + 300 = ' + E);

E = C + D;
document.writeln(' 300 + 400 = ' + E);
document.writeln("</PRE>");
</script>
</body>
</html>
```



Gambar 4. Output Program