



BAB I

MENGENAL JAVA SCRIPT

Oleh: Taryana Suryana M.Kom
Teknik Informatika Unikom

taryanarx@email.unikom.ac.id

taryanarx@gmail.com

[Line/Telegram: 081221480577](https://www.line.me/tv/081221480577)

Join Zoom Meeting

<https://us04web.zoom.us/j/3629329963?pwd=ZUhWcXl6RHp3dTNkZmlxWkUrV2ZCQT09>

Meeting ID: 362 932 9963

Passcode: 8HQiXn

1.1. JavaScript

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 dokumen, URL, windows, frame form, button atau item lain yang semuanya itu mempunyai properti yang saling berhubungan dengannya dan masing-masing memiliki nama, lokasi, warna nilai dan atribut lain.

1.2. Memulai JavaScript

Untuk dapat memulai dan menjalankan program JavaScript, ada dua perangkat lunak yang diperlukan yaitu browser dan teks editor. Teks editor adalah sebuah pengolah kata yang menghasilkan file dalam format ASCII murni. Bila Anda adalah pengguna Windows, Anda bisa menggunakan Notepad, Wordpad atau menggunakan Notepad++. Selain itu browser web yang akan anda gunakan harus mendukung



JavaScript, Anda dapat menggunakan Chrome, Firefox, Internet Explorer, Opera ataupun browser lainnya yang mendukung JavaScript.

JavaScript sebagai bahasa pemrograman yang berjalan di atas browser harus di tulis di dalam HTML. Ada empat cara penulisan javascript pada HTML.

1. Tag <script>

Cara umum yang dipakai adalah menuliskannya dalam tag <script>. Tag <script> bisa dibuat di dalam tag <head>, maupun di dalam tag <body>.

Menulis ke dalam output HTML menggunakan document.write ().

Contoh penulisan Java Script

NamaFile:js1.html

```
<html>
  <title>Penulisan Javascript</title>
  <script>
    document.write("ini kode Javascript");
  </script>
</html>
```

Menulis ke dalam elemen HTML, menggunakan innerHTML.

Namafie:js2.html

```
<html>
<h2>Penulisan dengan innerHTML</h2>
<p id="demo"></p>
<script>
  document.getElementById("demo").innerHTML = "Belajar Java Script";
</script>
</html>
```

2. File Eksternal

Jika tidak ingin kode Javascript bercampur dengan HTML, bisa menuliskannya pada file terpisah. Caranya, adalah dengan menyimpannya pada sebuah file yang berekstensi .js, misalnya isinya sebagai berikut.

```
// file-eksternal.js
alert("Kode Javascript dari File Eksternal");
```

Kemudian, kita perlu menghubungkan file eksternal tersebut dengan file HTML. Caranya, gunakan tag <script> dengan atribut src untuk menentukan lokasi file Javascriptnya.

```
<html>  
<title>Penulisan dengan external Javascript</title>  
  <script src="file-eksternal.js">  
  </script>  
</head>  
</html>
```

3. Atribut Event

Cara atribut event ini sering digunakan untuk memanggil fungsi pada event tertentu. Misalnya ketika suatu elemen diklik, maka jalankan fungsi Javascript.

```
<button onclick="alert('Ok Terima kasih!')">Klik donk!</button>
```

Selain *event onclick* ada juga *event-event* yang lain, seperti *onsubmit*, *onload*, *ondblclick*, *onmouseover*, *onmouseout*, dsb.

4. URL

Cara penulisan Javascript pada URL, cara ini jarang digunakan, namun kita juga perlu mengetahuinya. Penulisan Javascript pada URL menggunakan protokol Javascript. Misal, cobalah untuk menuliskan kode seperti ini pada URL browser.

```
javascript:alert("Nah! ini Javascript")
```

Kemudian bagaimana agar kita dapat menggunakan cara ini di HTML?

Cara ini bisa kita gunakan pada tag `<a>`, kemudian mengisi kode javascript pada atribut href.

Cara ini bisa menggantikan event onclick.

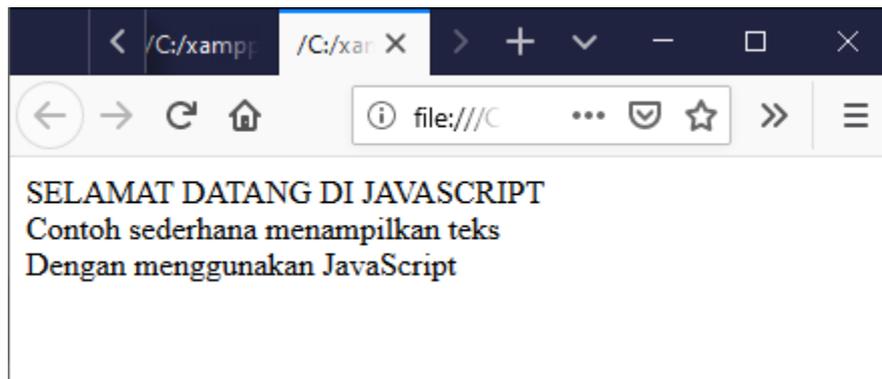
```
<a href="javascript:alert('Wih! hebat bukan?')">Klik Aku</a>
```

Dari ke empat cara penulisan JavaScript tadi diatas, gunakanlah yang menurut anda lebih mudah dalam penggunaannya.

1.3. Program JavaScript

Pada bagian ini, akan diberikan contoh program untuk menampilkan teks sederhana kelayar monitor.

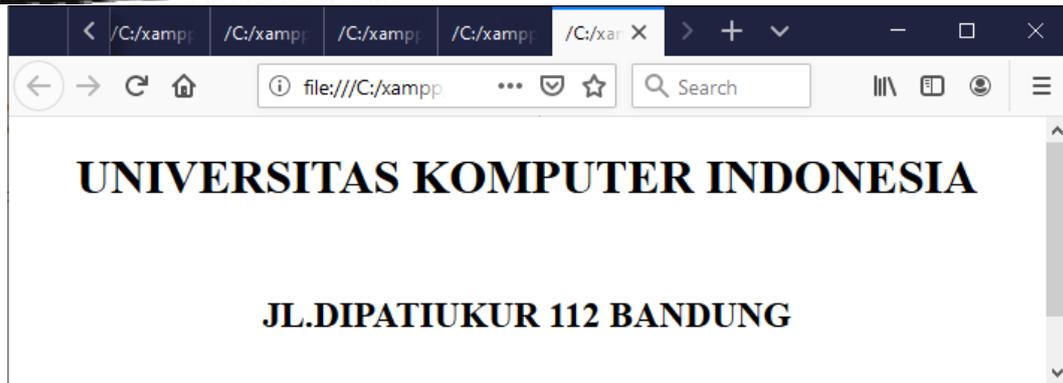
```
<script>
  document.write("SELAMAT DATANG DI JAVASCRIPT <br>");
  document.write("Contoh sederhana menampilkan teks<br>");
  document.write("Dengan menggunakan JavaScript<br>");
</script>
```



Gambar 1.1. Hasil dari Kode Program diatas.

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>

```
<script>
document.write("<center><h1>");
document.write("<h1>UNIVERSITAS KOMPUTER INDONESIA</h1><br>");
document.write("<h2>JL.DIPATIUKUR 112 BANDUNG<h2><br>");
document.write("</center>");
</script>
```



Gambar 1.2. Teks Header Pada JavaScript

1.4. 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:

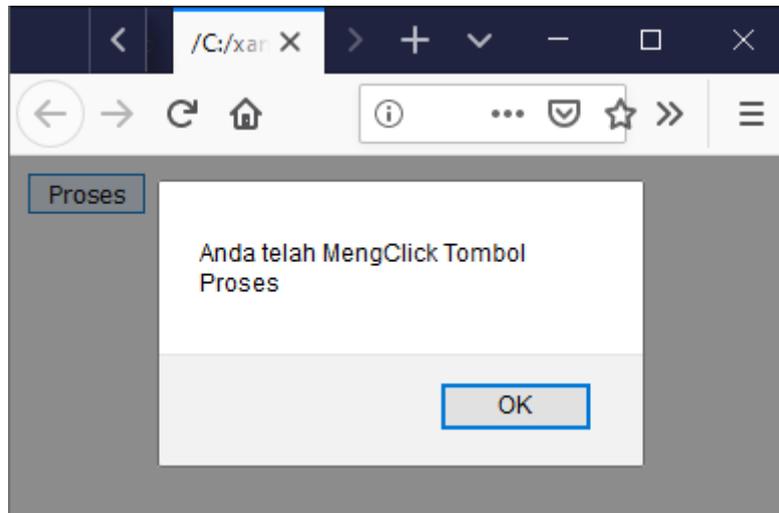
```
// ini cara menuliskan komentar  
atau  
/*  
  ini juga cara menuliskan komentar apabila lebih  
  dari satu baris  
*/
```

1.5. 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**. 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 1.3. 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.

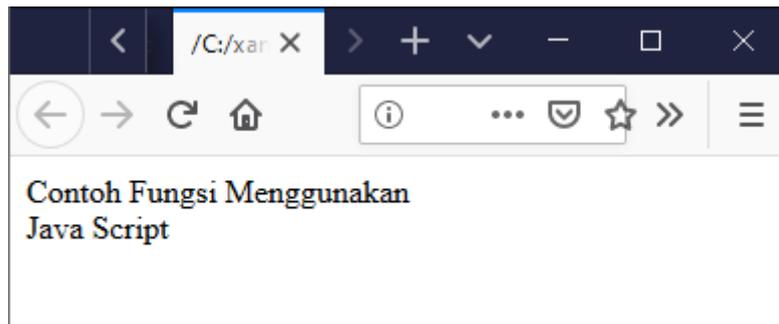
1.6. Function

Pada dasarnya function merupakan suatu cara yang dapat anda gunakan untuk menyatukan beberapa perintah

Fungsi Tanpa Parameter

Berikut merupakan contoh sebuah fungsi sederhana untuk menampilkan teks.

```
<script>
function menampilkan_teks()
{
    document.write("Contoh Fungsi Menggunakan<br>");
    document.write("Java Script<br>");
}
menampilkan_teks();
</script>
```



Gambar 1.4. Penggunaan Fungsi

Fungsi dengan Parameter

Parameter adalah variabel yang menyimpan nilai untuk diproses di dalam fungsi.

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

```
<script>
function penjumlahan(x,y) {
    var result = x + y;
    alert(result);
}
penjumlahan(10,5);
</script>
```

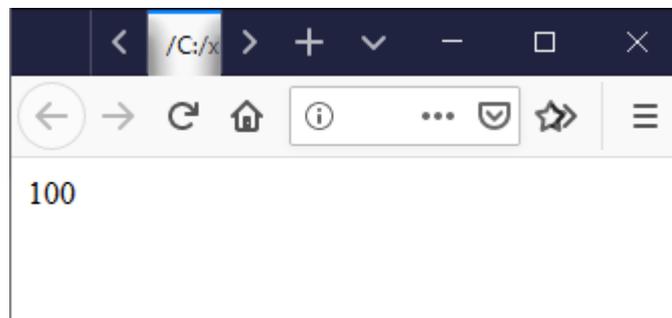
Fungsi yang Mengembalikan Nilai

Agar hasil pengolahan nilai di dalam fungsi dapat digunakan untuk proses berikutnya, maka fungsi dapat juga diprogram untuk mengembalikan nilai.

Pengembalian nilai pada fungsi menggunakan kata kunci return kemudian diikuti dengan nilai atau variabel yang akan dikembalikan. Contoh:

```
<script>
function kali(a,b){
    hasilkali = a * b;
    return hasilkali;
}

// memanggil fungsi
var nilai1 = 20;
var nilai2 = 5;
var hasilperkalian = kali(nilai1, nilai2);
document.write(hasilperkalian);
</script>
```



Gambar 1.5. Fungsi yang mengembalikan nilai

1.7. Variabel dalam JavaScript

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"

X = 2021



Y = 12122019

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

Contoh

```
var nama;
```

```
var nama="Fahra Ragita";
```

```
var X = 2021;
```

```
var Y;
```

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

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

1.8. 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 "Asep", "Mei", "Juli", "Kenapa Mundur", "JavaScript" dsb
- Boolean, hanya bernilai true atau false
- Null, yaitu variabel yang tidak diinisialisasi

1.8.1. 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
```

1.8.2. 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 ";
```

1.8.3. 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 menunjukkan bahwa jika Y lebih besar dari 90 maka X akan bernilai True.

1.8.4. Tipe Null

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

```
var x;
```

1.9. Operator

Operator pada JavaScript dibagi menjadi enam, yaitu

- Aritmetik
- Pemberian nilai (Assign)
- Pemanipulasian bit (bitwise)
- Perbandingan

- Logika
- String

1.9.1. 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

1.9.2. Operator Pemberian Nilai

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

Operator	Keterangan	Contoh	Equivalen
=	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

1.9.3. Operasi Pemanipulasi Bit

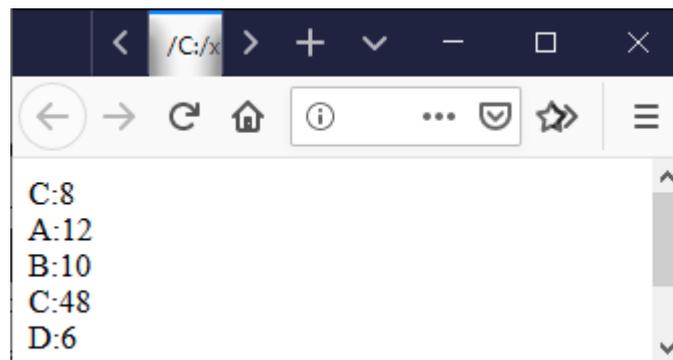
Operasi ini berhubungan dengan pemanipulasian bit pada operan bertipe bilangan bulat.

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

```
<script>
var A = 12;      // A = 1100b
var B = 10;     // B = 1010b
var C = A & B
/*maka akan dihasilkan bilangan seperti berikut:
1100b
```

```
1010b AND 1000b  
*/  
document.write("C:",C,"<br>");  
var A = 12;  
var C = A << 2;  
var D = A>>1  
document.write("A:",A,"<br>");  
document.write("B:",B,"<br>");  
document.write("C:",C,"<br>");  
document.write("D:",D,"<br>");  
</script>
```

maka variabel C akan bernilai 48 (0011 0000b)
variabel D akan bernilai 6 (0110b).



Gambar 1.7. Contoh Hasil Operasi Bit

1.9.4. Operator Perbandingan

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

1.9.5. 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
```

1.9.6. Operator String

Selain operator pembandingan, 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.

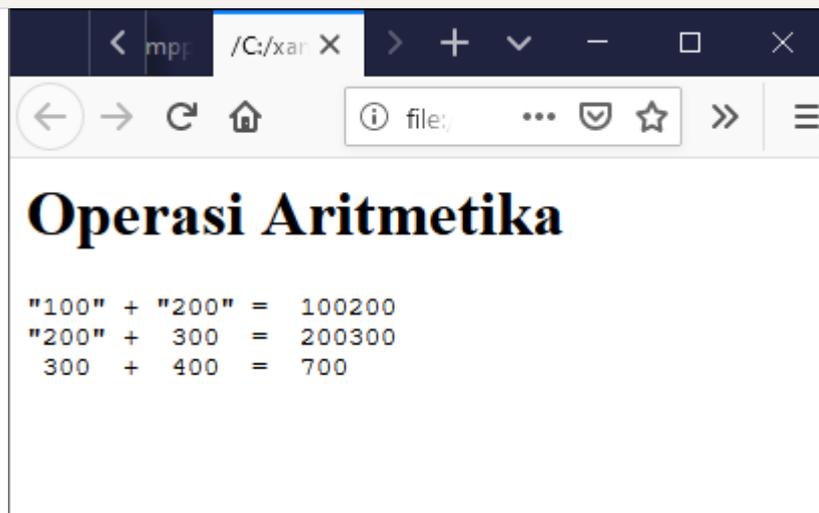
Contoh:

Nama = "Java" + " Script";

Akan menghasilkan "JavaScript" pada variabel Nama

Penggunaan Operator String

```
<script>
document.writeln("<h1>Operasi Aritmetika </h1><pre>");
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);
</script>
```



Gambar 1.8. Output Program Operasi String

Referensi:

**Aplikasi Internet Menggunakan HTML, CSS & Java Script, Taryana Suryana, Elexmedia Komputindo*