

Modul Perkuliahan Pemrograman Web

Topik: Java Script

(Aksara Jawa)

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Universitas Kristen Maranatha
Bandung

Disusun Oleh:
Setia Budi, S.Kom

Catatan:

Modul ini disusun dengan menggunakan Open Office

yang berjalan di atas sistem operasi Ubuntu Linux

Pengujian Java Script dilakukan dengan menggunakan FireFox

yang berjalan di atas sistem operasi Ubuntu Linux

Mari budayakan penggunaan open source software :)

Pengenalan Java Script

Java Script merupakan bahasa pemrograman yang dieksekusi di sisi klien atau yang biasa dikenal dengan istilah client side programming di mana suatu kode program akan dikirimkan ke klien dan dieksekusi / dijalankan oleh web browser dan bukannya dieksekusi di web server.

Dalam implementasinya, suatu kode program Java Script akan disisipkan pada suatu dokumen HTML dengan menggunakan pasangan tag `<SCRIPT>` dan `</SCRIPT>` di atribut `LANGUAGE` dari tag `<SCRIPT>` akan diberi nilai `JavaScript`.

```
<SCRIPT LANGUAGE="JavaScript">
...
...
...
</SCRIPT>
```

Dalam menuliskan kode program Java Script ada beberapa hal yang perlu diperhatikan:

1. Java Script merupakan bahasa pemrograman yang bersifat case sensitive, artinya Java Script akan membedakan penggunaan huruf besar dan huruf kecil. "DOCUMENT" akan lain artinya dengan "document" ataupun "Document".
2. Setiap statement / pernyataan dalam kode program Java Script selalu diakhiri dengan karakter ";" (titik koma / semicolon).

Mencetak Teks

Dalam Java Script perintah yang digunakan untuk mencetak suatu teks pada dokumen HTML adalah `document.write("pesannya")`.

Untuk lebih jelasnya bisa melihat contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      document.write("<H1>Aksara Jawa</H1>");
    </SCRIPT>
    <I>Tulisan di atas dihasilkan oleh Java Script</I>
  </BODY>
</HTML>
```



Memberi Komentar

Dalam menuliskan kode program adakalanya dibutuhkan untuk menuliskan suatu komentar ataupun keterangan berkenaan dengan kode program tersebut, oleh karenanya dibutuhkan suatu mekanisme penulisan komentar. Komentar itu sendiri akan diabaikan dan tidak akan dieksekusi / dijalankan oleh web browser.

Terdapat dua cara untuk menyertakan komentar dalam Java Script:

1. Menggunakan penanda `//`
Penanda `//` digunakan untuk menuliskan komentar dalam satu baris
2. Menggunakan pasangan penanda `/*` dan `*/`
Pasangan penanda `/*` dan `*/` digunakan untuk menuliskan komentar yang lebih dari satu baris

Untuk lebih jelasnya bisa melihat contoh berikut ini:

```

<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      //ini adalah komentar dalam satu baris
      document.write("<H1>Belajar Java Script itu menyenangkan</H1>");
      /* ini adalah komentar
         dengan jumlah baris
         lebih dari satu
      */
    </SCRIPT>
    <I>Tulisan di atas dihasilkan oleh Java Script</I>
  </BODY>
</HTML>

```



Tipe Data dan Deklarasi Variabel

Dalam Java Script terdapat tiga macam tipe data utama, yaitu tipe data String (kumpulan karakter), tipe data bilangan / numerik, dan tipe data logika / boolean. Pada tipe data boolean akan mengenal dua buah nilai yaitu true dan false.

Pendeklarasian variabel dalam Java Script ditandai dengan penggunaan kata kunci **var** dan diikuti dengan nama variabel. Tipe data dari suatu variabel ditentukan oleh nilai yang diisikan ke dalam variabel tersebut dan bukannya dideklarasikan secara eksplisit. Untuk lebih jelasnya perhatikan contoh berikut ini:

```

<SCRIPT LANGUAGE="JavaScript">
  var kota = "bandung"; // deklarasi variabel bernama kota bertipe data string
  var gaji = 0; // deklarasi variabel bernama gaji bertipe data integer
  var ipk = 2.75; // deklarasi variabel bernama ipk bertipe data float
  var status_cekal = false; // deklarasi variabel bernama status_cekal bertipe data boolean
</SCRIPT>

```

Operator Aritmatika

Operator aritmatika adalah operator yang digunakan untuk menangani /mengoperasikan suatu nilai bertipe data numerik.

Berikut ini adalah daftar tipe data numerik:

operator	keterangan	contoh penggunaan
+	penjumlahan	3 + 7 menghasilkan nilai 10 2 + 5 menghasilkan nilai 7
-	pengurangan	9 – 3 menghasilkan nilai 6 6 – 2 menghasilkan nilai 4
*	perkalian	2 * 3 menghasilkan nilai 6 4 * 5 menghasilkan nilai 20
/	pembagian	8 / 2 menghasilkan nilai 4 10 / 5 menghasilkan nilai 2
%	modulus / sisa pembagian	8 % 3 menghasilkan nilai 2 7 % 2 menghasilkan nilai 1
++	increment / penaikan nilai sebesar 1	x = 5 x ++ maka nilai variabel x saat ini adalah 6
--	decrement / penurunan nilai sebesar 1	x = 8 x -- maka nilai variabel x saat ini adalah 7

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE>Ini Judul</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    <H1>Operasi Aritmatika</H1>
```

```
    <SCRIPT LANGUAGE="JavaScript">
```

```
      var a, b, c;
```

```
      a = 7; b = 4; c = 0;
```

```
      document.write("nilai variabel a = " + a + "<BR>");
```

```
      document.write("nilai variabel b = " + b + "<BR>");
```

```
      c = a + b;
```

```
      document.write("a + b = " + c + "<BR>");
```

```
      c = a - b;
```

```
      document.write("a - b = " + c + "<BR>");
```

```
      c = a * b;
```

```
      document.write("a * b = " + c + "<BR>");
```

```
      c = a / b;
```

```
      document.write("a / b = " + c + "<BR>");
```

```
      c = a % b;
```

```
      document.write("a % b = " + c + "<BR>");
```

```
      a++;
```

```
      document.write("a++ maka nilai a sekarang adalah " + a + "<BR>");
```

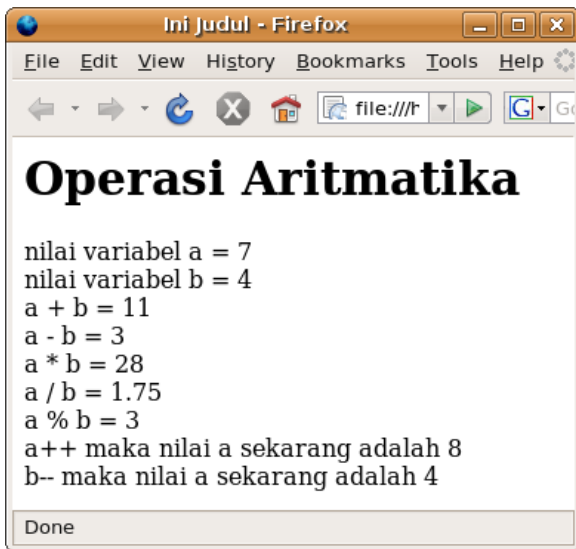
```
      a--;
```

```
      document.write("b-- maka nilai a sekarang adalah " + b);
```

```
    </SCRIPT>
```

```
  </BODY>
```

```
</HTML>
```



Operator Pemberian Nilai / Assignment

Operator penugasan / assignment adalah operator yang digunakan untuk memberikan / memasukkan nilai ke dalam suatu variabel.

Berikut ini adalah daftar operator penugasan:

operator	keterangan	contoh
=	nilai yang berada di sisi kanan akan dimasukkan ke variabel di sisi kiri	x = 3 maka nilai variabel x saat ini adalah 3
+=	nilai dari variabel yang berada di sisi kiri akan ditambahkan dengan nilai yang berada di sisi kanan	x = 5 x += 2 maka nilai variabel x saat ini adalah 7
-=	nilai dari variabel yang berada di sisi kiri akan dikurangkan dengan nilai yang berada di sisi kanan	x = 6 x -= 4 mana nilai variabel x saat ini adalah 2
*=	nilai dari variabel yang berada di sisi kiri akan dikalikan dengan nilai yang berada di sisi kanan	x = 2 x *= 7 maka nilai variabel x saat ini adalah 14
/=	nilai dari variabel yang berada di sisi kiri akan dibagi dengan nilai yang berada di sisi kanan	x = 24 x /= 8 maka nilai variabel x saat ini adalah 3
%=	nilai dari variabel yang berada di sisi kiri akan dimoduluskan dengan nilai yang berada di sisi kanan	x = 7 x %= 3 maka nilai variabel x saat ini adalah 1

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

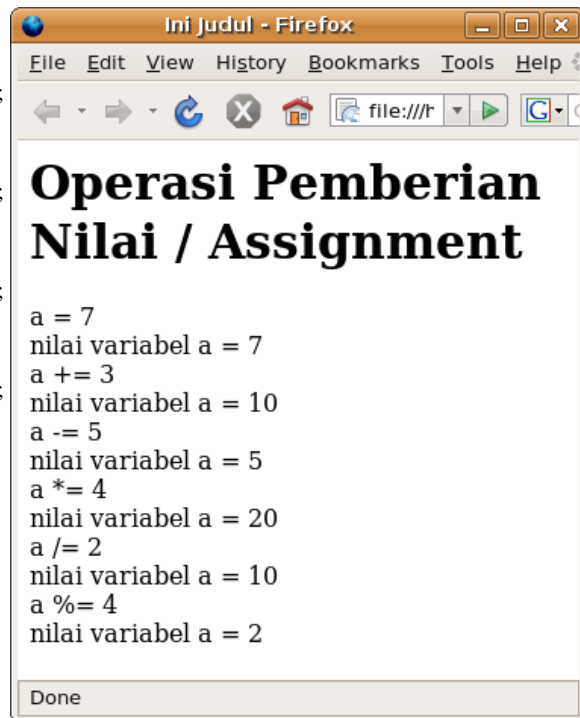
```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Operasi Pemberian Nilai / Assignment</H1>
    <SCRIPT LANGUAGE="JavaScript">
```

```
var a;
a = 7;
document.write("a = 7 <BR>");
document.write("nilai variabel a = " + a + "<BR>");
a += 3;
document.write("a += 3<BR>");
document.write("nilai variabel a = " + a + "<BR>");
```

```

a -= 5;
document.write("a -= 5<BR>");
document.write("nilai variabel a = " + a + "<BR>");
a *= 4;
document.write("a *= 4<BR>");
document.write("nilai variabel a = " + a + "<BR>");
a /= 2;
document.write("a /= 2<BR>");
document.write("nilai variabel a = " + a + "<BR>");
a %= 4;
document.write("a %= 4<BR>");
document.write("nilai variabel a = " + a + "<BR>");
</SCRIPT>
</BODY>
</HTML>

```



Operator Pembanding / Comparison

Operator pembanding adalah operator yang digunakan untuk membandingkan dua buah nilai dan akan menghasilkan nilai bertipe data boolean yaitu benar(true) atau salah (false).

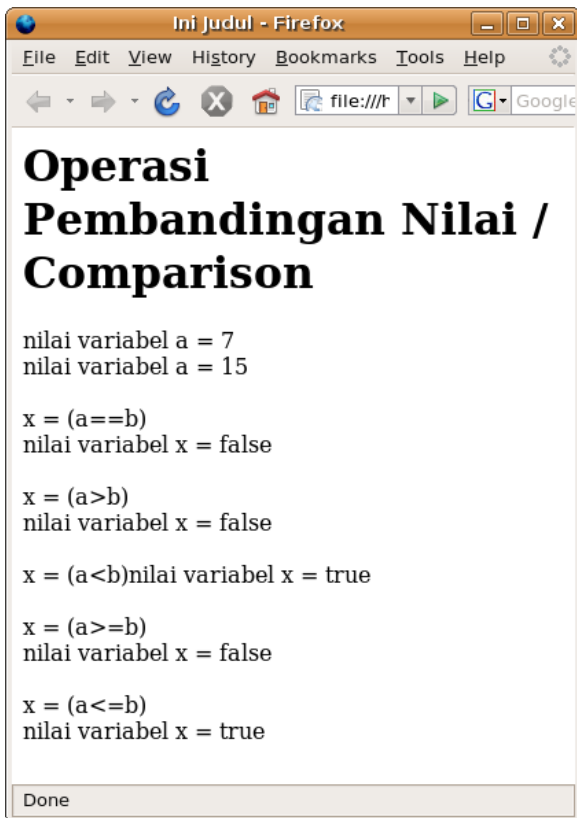
Berikut ini adalah daftar operator pembanding / comparison

operator	keterangan	contoh
==	akan menghasilkan nilai true bila kedua nilai yang dibandingkan memiliki bobot yang sama	x = (4 == 2) maka nilai variabel x saat ini adalah false
!=	akan menghasilkan nilai true bika kedua nilai yang dibandingkan memiliki bobot berbeda / tidak sama	x = (4 != 2) maka nilai variabel x saat ini adalah true
>	akan menghasilkan nilai true apabila nilai pertama (nilai yang berada di sisi kiri) memiliki bobot lebih besar dibandingkan nilai kedua (nilai yang berada di sisi kanan)	x = (4 > 2) maka nilai variabel x saat ini adalah true
<	akan menghasilkan nilai true apabila nilai pertama (nilai yang berada di sisi kiri) memiliki bobot lebih kecil dibandingkan nilai kedua (nilai yang berada di sisi kanan)	x = (4 < 2) maka nilai variabel x saat ini adalah false
>=	akan menghasilkan nilai true apabila nilai pertama (nilai yang berada di sisi kiri) memiliki bobot lebih besar atau sama dengan nilai kedua (nilai yang	x = (4 >= 4) maka nilai variabel x saat ini adalah true

	berada di sisi kanan)	
<=	akan menghasilkan nilai true apabila nilai pertama (nilai yang berada di sisi kiri) memiliki bobot lebih kecil atau sama dengan nilai kedua (nilai yang berada di sisi kanan)	x = (4 <= 8) maka nilai variabel x saat ini adalah true

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Operasi Pembandingan Nilai / Comparison</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var a, b, x;
      a = 7; b = 15;
      document.write("nilai variabel a = " + a + "<BR>");
      document.write("nilai variabel a = " + b + "<BR><BR>");
      x = (a==b);
      document.write("x = (a==b)<BR>");
      document.write("nilai variabel x = " + x + "<BR><BR>");
      x = (a>b);
      document.write("x = (a > b)<BR>");
      document.write("nilai variabel x = " + x + "<BR><BR>");
      x = (a<b);
      document.write("x = (a < b)<BR>");
      document.write("nilai variabel x = " + x + "<BR><BR>");
      x = (a>=b);
      document.write("x = (a >= b)<BR>");
      document.write("nilai variabel x = " + x + "<BR><BR>");
      x = (a<=b);
      document.write("x = (a <= b)<BR>");
      document.write("nilai variabel x = " + x + "<BR><BR>");
    </SCRIPT>
  </BODY>
</HTML>
```



Operator Logika

Operator logika adalah operator yang digunakan untuk mengoperasikan nilai-nilai bertipe data boolean yang bernilai true atau false.

Berikut ini adalah daftar operator logika

operator	keterangan	contoh
&&	operator and, operator ini akan menghasilkan nilai false bila di antara nilai-nilai yang dioperasikan terdapat nilai false	x = (3<7) && (4>7) maka nilai variabel x saat ini adalah false
	operator or, operator ini akan menghasilkan nilai true bila di antara nilai-nilai yang dioperasikan terdapat nilai true	x = (3<7) (4>7) maka nilai variabel x saat ini adalah true
!	operator not, operator ini akan menghasilkan nilai true bila nilai yang dioperasikan bernilai false dan akan menghasilkan nilai false bila nilai yang dioperasikan bernilai true	x = !(3<7) maka nilai variabel x saat ini adalah false

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:


```

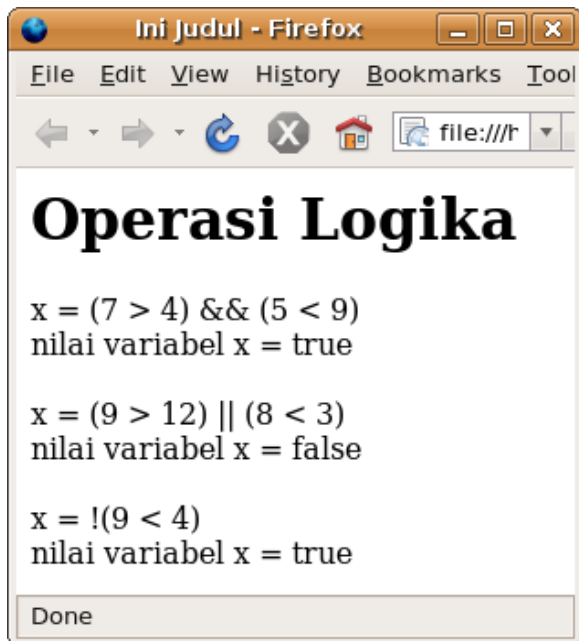
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Operasi Logika</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var x;

      x = (7>4) && (5<9);
      document.write("x = (7 > 4) && (5 < 9)<BR>");
      document.write("nilai variabel x = " + x + "<BR><BR>");

      x = (9>12) || (8<3);
      document.write("x = (9 > 12) || (8 < 3)<BR>");
      document.write("nilai variabel x = " + x + "<BR><BR>");

      x = !(9<4);
      document.write("x = !(9 < 4)<BR>");
      document.write("nilai variabel x = " + x);
    </SCRIPT>
  </BODY>
</HTML>

```



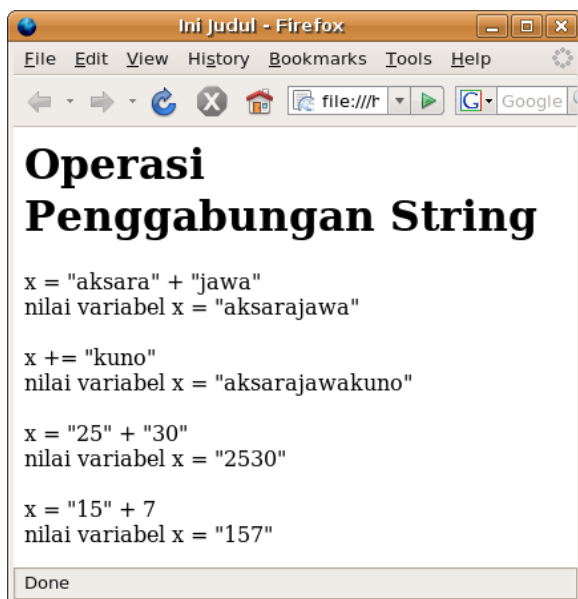
Operasi Penggabungan String

Seperti pernah dijelaskan sebelumnya bahwa nilai bertipe data string merupakan nilai yang terdiri dari sekumpulan karakter sehingga membentuk suatu nilai berupa kata ataupun kalimat tertentu. Untuk nilai-nilai bertipe data string terdapat suatu operator yang berguna untuk menggabungkan nilai string. Operator ini dikenal dengan nama operator penggabungan string yang dituliskan dengan penanda +.

Apabila suatu nilai bertipe data string dijumlahkan dengan nilai bertipe data numerik maka secara otomatis nilai bertipe data numerik tersebut akan dikonversi menjadi tipe data string sehingga pada akhirnya akan dihasilkan suatu nilai bertipe data string.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Operasi Penggabungan String</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var x;
      x = "aksara" + "jawa";
      document.write('x = "aksara" + "jawa"<BR>');
      document.write('nilai variabel x = "' + x + '"<BR><BR>');
      x += "kuno";
      document.write('x += "kuno"<BR>');
      document.write('nilai variabel x = "' + x + '"<BR><BR>');
      x = "25" + "30";
      document.write('x = "25" + "30"<BR>');
      document.write('nilai variabel x = "' + x + '"<BR><BR>');
      x = "15" + 7;
      document.write('x = "15" + 7<BR>');
      document.write('nilai variabel x = "' + x + '"');
    </SCRIPT>
  </BODY>
</HTML>
```



Konversi Tipe Data

Dalam pemrograman java script dimungkinkan untuk melakukan konversi / pengubahan tipe data.

Tipe data yang memungkinkan untuk dikonversi adalah tipe data string ke tipe data numerik.

Ada dua fungsi yang bisa digunakan untuk mengkonversi suatu nilai bertipe data string ke numerik, Fungsi

yang pertama adalah fungsi parseInt() dan fungsi yang kedua adalah fungsi parseFloat().

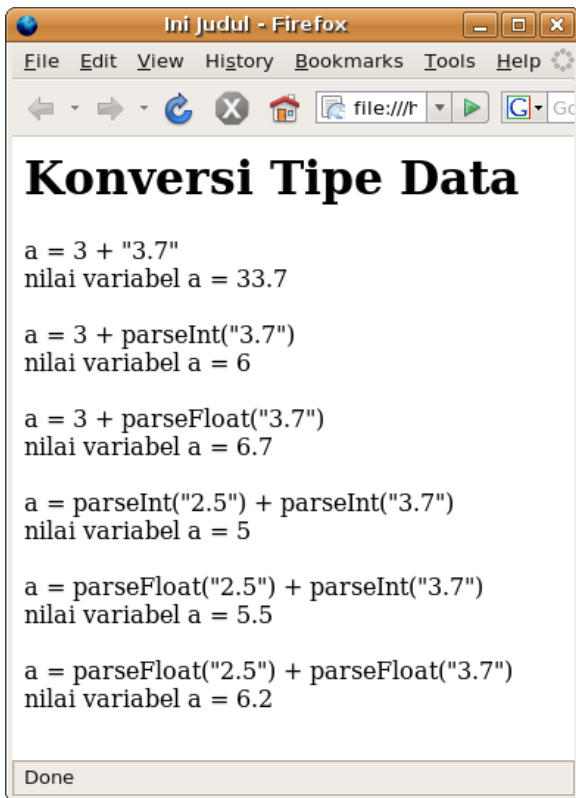
Fungsi parseInt() digunakan untuk melakukan konversi dari tipe data string ke tipe data numerik berbentuk bilangan bulat / integer.

Dan fungsi parseFloat() digunakan untuk melakukan konversi dari tipe data string ke tipe data numerik berbentuk bilangan pecahan / floating point.

fungsi	keterangan	contoh
parseInt()	digunakan untuk konversi dari tipe data string ke tipe data numerik bilangan bulat / integer	a = "2.5" x = 3 + parseInt(a); maka nilai variabel x saat ini adalah 5
parseFloat()	digunakan untuk konversi dari tipe data string ke tipe data numerik bilangan pecahan / floating point	a = "2.5" x = 3 + parseFloat(a); maka nilai variabel x saat ini adalah 5.5

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Konversi Tipe Data</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var a;
      a = 3 + "3.7";
      document.write('a = 3 + "3.7"<BR>');
      document.write("nilai variabel a = " + a + "<BR><BR>");
      a = 3 + parseInt("3.7");
      document.write('a = 3 + parseInt("3.7")<BR>');
      document.write("nilai variabel a = " + a + "<BR><BR>");
      a = 3 + parseFloat("3.7");
      document.write('a = 3 + parseFloat("3.7")<BR>');
      document.write("nilai variabel a = " + a + "<BR><BR>");
      a = parseInt("2.5") + parseInt("3.7");
      document.write('a = parseInt("2.5") + parseInt("3.7")<BR>');
      document.write("nilai variabel a = " + a + "<BR><BR>");
      a = parseFloat("2.5") + parseInt("3.7");
      document.write('a = parseFloat("2.5") + parseInt("3.7")<BR>');
      document.write("nilai variabel a = " + a + "<BR><BR>");
      a = parseFloat("2.5") + parseFloat("3.7");
      document.write('a = parseFloat("2.5") + parseFloat("3.7")<BR>');
      document.write("nilai variabel a = " + a + "<BR><BR>");
    </SCRIPT>
  </BODY>
</HTML>
```



Mengenal Statemen if

Statemen if digunakan untuk melakukan percabangan dalam melakukan eksekusi sejumlah statement.

Bentuk umum dari statement if yang paling sederhana adalah:

```
if (kondisi) {
    //statement-statement yang akan dieksekusi bila kondisi benar
}
```

Statement if tersebut adalah bentuk statement if yang paling sederhana di mana statement-statement yang berada di antara penanda kurung kurawal akan dieksekusi apabila kondisi yang dicek menghasilkan nilai true. Apabila kondisi yang dicek ternyata menghasilkan nilai false, maka statement-statement yang berada di antara penanda kurung kurawal tersebut tidak akan dieksekusi.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Statement if</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var x = 8;
      document.write("nilai variabel x = " + x + "<BR>");
      if (x > 0) {
        document.write("x adalah bilangan positif");
      }
    </SCRIPT>
  </BODY>
</HTML>
```



Mengenal Statement if ... else

Statement if ... else ... merupakan pengembangan dari bentuk statement if. Bentuk umum dari statement ini

adalah:

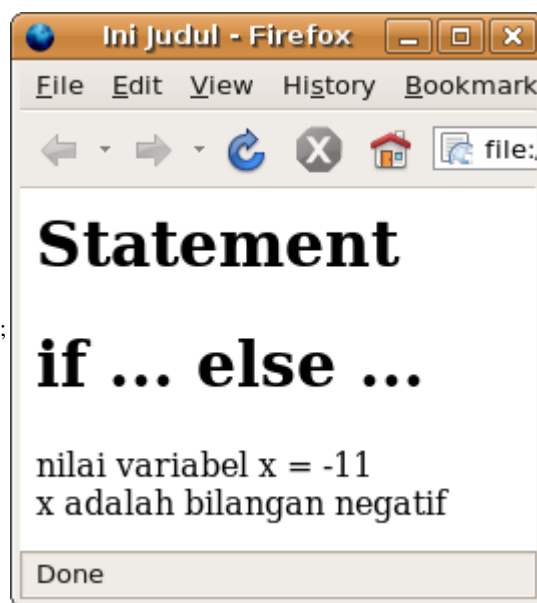
```
if (kondisi) {  
    //statement-statement yang dieksekusi bila kondisi benar  
} else {  
    statement-statement yang dieksekusi bila kondisi salah  
}
```

Pada statement ini terdapat dua blok kurung kurawal, di mana blok kurung kurawal pertama akan berisi statement-statement yang hanya akan dieksekusi apabila kondisi yang dicek menghasilkan nilai true, sedangkan statement-statement pada blok kurung kurawal kedua akan diabaikan.

Apabila ternyata kondisi yang dicek menghasilkan nilai false maka statement-statement pada blok kurung kurawal pertama tidak akan dieksekusi melainkan statement-statement pada kurung kurawal kedua lah yang akan dieksekusi.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>  
<HEAD>  
  <TITLE>Ini Judul</TITLE>  
</HEAD>  
<BODY>  
  <H1>Statement</H1>  
  <H1>if ... else ...</H1>  
  <SCRIPT LANGUAGE="JavaScript">  
    var x = -11;  
    document.write("nilai variabel x = " + x + "<BR>");  
    if (x > 0) {  
      document.write("x adalah bilangan positif");  
    } else {  
      document.write("x adalah bilangan negatif");  
    }  
  </SCRIPT>  
</BODY>  
</HTML>
```



Mengenal Statement if ... else if ...

Statement if ... else if ... memungkinkan terjadinya pengecekan kondisi secara berlapis. Berikut ini adalah bentuk umum dari statement if ... else if ...

```
if (kondisi1) {  
    //statement-statement bila kondisi 1 benar  
} else if (kondisi2) {  
    //statement-statement bila kondisi 2 benar  
} else {  
    //statement-statement bila semua kondisi salah  
}
```

Bentuk ini juga dikenal sebagai if bertingkat di mana terdapat beberapa kondisi yang akan dicek. Kondisi kedua hanya akan dicek apabila kondisi pertama bernilai salah, kondisi ketiga juga hanya akan dicek apabila kondisi pertama dan kedua bernilai salah, demikian seterusnya proses pengecekan kondisi berlangsung sampai ditemui suatu kondisi yang bernilai benar. Apabila tidak ditemui satu kondisi pun yang bernilai benar maka yang akan dieksekusi adalah statement-statement yang berada pada blok else.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Statement</H1>
    <H1>if ... else if ...</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var x = -11;
      document.write("nilai variabel x = " + x + "<BR>");
      if (x > 0) {
        document.write("x adalah bilangan positif");
      } else if (x < 0) {
        document.write("x adalah bilangan negatif");
      } else {
        document.write("x adalah bilangan nol");
      }
    </SCRIPT>
  </BODY>
```



Mengenal Statement switch ... case ...

Statement switch ... case ... merupakan suatu bentuk statement alternatif untuk melakukan percabangan. Statement ini sebenarnya merupakan bentuk ringkas dari statement if ... else if ...

Bentuk umum dari statement switch ... case ... adalah:

```
switch (variabel) {
  case nilai_penguji1: //statement-statement pilihan 1
    break;
  case nilai_penguji2: //statement-statement pilihan 2
    break;
  default : //statement-statement default
};
```

Statement-statement pilihan 1 hanya akan dieksekusi bila didapati nilai variabel yang diuji sama dengan nilai penguji 1, dan statement-statement pilihan 2 pun hanya akan dieksekusi bila nilai dari variabel yang diuji sama dengan nilai penguji 2, demikian seterusnya. Yang perlu diperhatikan di sini adalah di akhir penulisan statement-statement pada tiap pilihan harus diakhiri oleh statement break.

Apabila nilai dari variabel yang diuji tidak memiliki kesamaan dengan salah satu sari nilai-nilai penguji yang ada, maka statement-statement yang akan dieksekusi adalah statement-statement pada blok default.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>switch ... case ...</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var hari = 5;
      document.write("nilai variabel hari = " +
        hari + "<BR>");
      document.write("hari yang dipilih: ");
      switch(hari) {
        case 1: document.write("senin");
          break;
        case 2: document.write("selasa");
          break;
        case 3: document.write("rabu");
          break;
        case 4: document.write("kamis");
          break;
        case 5: document.write("jumat");
          break;
        case 6: document.write("sabtu");
          break;
        case 7: document.write("minggu");
          break;
        default: document.write("tidak terdaftar");
      }
    </SCRIPT>
```

```

</SCRIPT>
</BODY>
</HTML>

```



Mengenal Operator ... ? ... : ...

Operator ... ? ... : ... digunakan untuk melakukan suatu proses percabangan sederhana. Operator ini biasa digunakan untuk menyederhanakan statement if ... else ..., hanya saja pada operator ini hanya mampu menangani satu statement untuk pengecekan kondisi yang menghasilkan nilai true dan satu statement untuk pengecekan kondisi yang menghasilkan nilai false.

Bentuk umum dari penggunaan operator ... ? ... : ... adalah:

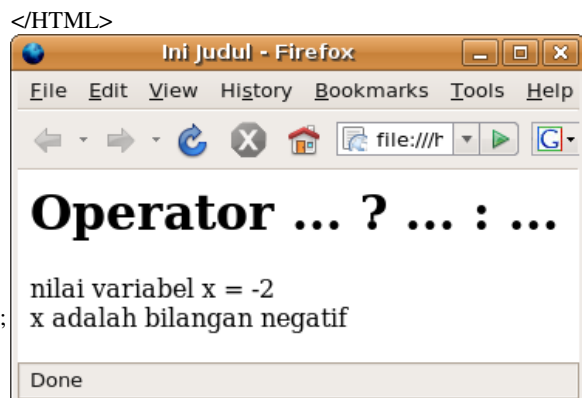
kondisi ? statement_kondisi_true : statement_kondisi_false;

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```

<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Operator ... ? ... : ...</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var x, hasil;
      x = -2;
      document.write("nilai variabel x = " + x + "<BR>");
      hasil = (x>0) ? "positif" : "negatif";
      document.write("x adalah bilangan " + hasil);
    </SCRIPT>
  </BODY>

```



Mengenal Statement while

Statement while merupakan salah satu statement yang bisa digunakan untuk melakukan suatu proses perulangan / looping. Bentuk umum dari statement while adalah:

```

while (kondisi) {
    //statement-statement yang diulang
}

```

Statement-statement yang berada di antara penanda kurung kurawal buka dan kurung kurawal tutup akan diulang berkali-kali selama kondisi yang dicek menghasilkan nilai true. Proses perulangan akan berakhir ketika didapati kondisi yang dicek menghasilkan nilai false.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```

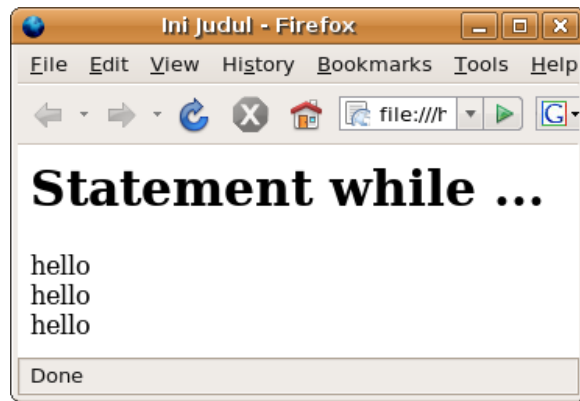
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>

```

```

<BODY>
  <H1>Statement while ...</H1>
  <SCRIPT LANGUAGE="JavaScript">
    var a=4;
    while(a>1) {
      document.write("hello<BR>");
      a--;
    }
  </SCRIPT>
</BODY>
</HTML>

```



Mengenal Statement do ... while

Selain menggunakan statement while, proses perulangan / looping juga bisa dilakukan dengan menggunakan statement do ... while. Bentuk umum dari statement do ... while adalah:

```

do {
    //statement-statement yang diulang
} while (kondisi);

```

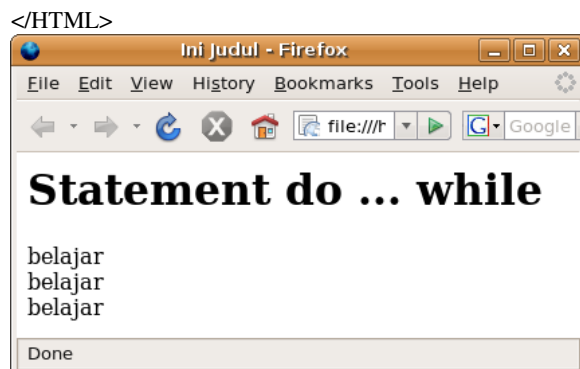
Sedikit berbeda dengan statement while, pada statement do ... while pengecekan kondisi dilakukan di akhir penanda kurung kurawal. Artinya statement-statement yang berada di antara penanda kurung kurawal akan dieksekusi terlebih dahulu baru kemudian dilakukan pengecekan kondisi. Apabila kondisi yang dicek menghasilkan nilai true maka statement-statement tersebut akan dieksekusi kembali secara berulang-ulang sampai ditemui hasil pengecekan kondisi yang menghasilkan nilai false.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```

<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Statement do ... while</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var a=4;
      do {
        document.write("belajar<BR>");
        a--;
      } while(a>1);
    </SCRIPT>
  </BODY>
</HTML>

```



Mengenal Statement for

Statement for juga merupakan salah satu statement yang bisa digunakan untuk melakukan perulangan / looping. Bentuk umum dari statement for adalah:

```

for (inisialisasi; kondisi; penaikan_penurunan) {
    //statement-statement yang diulang
}

```

Statement for banyak digunakan untuk melakukan perulangan di mana banyaknya jumlah perulangan sudah diketahui dengan pasti sebelumnya. Ini sialisasi dalam konteks for merupakan proses pemberian nilai awal pada variabel pencacah. Variabel inilah yang akan selalu diuji nilainya pada bagian pengecekan kondisi. Apabila hasil pengecekan kondisi bernilai benar maka statement-statement yang berada di antara kurung

kurawal buka dan kurung kurawal tutup akan dieksekusi. Diakhir eksekusi statement-statement tersebut, nilai dari variabel pencacah tersebut akan mengalami proses penaikan ataupun penurunan. Setelah itu variabel ini akan kembali dicek nilainya di bagian kondisi. Dan apabila didapati hasil true dari proses pengecekan tersebut maka statement-statement pada blok kurung kurawal akan kembali diulang. Dan demikian seterusnya proses perulangan ini berlangsung sampai didapati nilai false dari hasil pengecekan kondisi.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>Statement for</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var a;
      for(a=1;a<7;a++) {
        document.write("semangat!!!<BR>");
      }
    </SCRIPT>
  </BODY>
</HTML>
```



Mengenal Message Box

Dalam Java Script terdapat tiga macam message box standar, yaitu alert, confirm, dan prompt. Dalam JavaScript ketiga message box ini merupakan bagian dari objek window.

Alert

Alert digunakan untuk menampilkan pesan / informasi yang sifatnya penting dan mendesak. Misalkan saja informasi yang menyatakan bahwa suatu halaman web membutuhkan shockwave player yang terinstal pada web browser.

Bentuk umum dari message box alert adalah:

```
window.alert(pesan);
```

Untuk lebih jelasnya dapat mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>ALERT</H1>
    <SCRIPT LANGUAGE="JavaScript">
      window.alert("JavaScript memang hebat");
    </SCRIPT>
  </BODY>
</HTML>
```



Confirm

Confirm digunakan untuk meminta konfirmasi dari pengunjung web. Nilai yang mampu ditampung oleh

message box ini adalah nilai true atau false. Kedua nilai ini dihasilkan dari penekanan tombol Ok atau Cancel. Penekanan tombol Ok akan menghasilkan nilai true sedangkan penekanan tombol Cancel akan menghasilkan nilai false.

Bentuk umum dari message box confirm adalah:

```
window.confirm(pertanyaan);
```

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>CONFIRM</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var komentar;
      komentar = window.confirm("Apakah Anda vegetarian?");
      if (komentar) {
        document.write("Anda seorang vegetarian");
      } else {
        document.write("Anda bukan seorang vegetarian");
      }
    </SCRIPT>
  </BODY>
</HTML>
```



Prompt

Prompt dapat digunakan untuk menampung nilai yang diinputkan oleh penunjung web.

Bentuk umum dari message box prompt adalah:

```
window.prompt(pesan, nilai_default);
```

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
  </HEAD>
  <BODY>
    <H1>PROMPT</H1>
    <SCRIPT LANGUAGE="JavaScript">
      var nama;
      nama = window.prompt("Masukkan nama Anda", "Mbah Boedy");
      document.write("Selamat Datang " + nama);
    </SCRIPT>
  </BODY>
</HTML>
```



Mengenal function

Penggunaan function memungkinkan penulisan suatu kode program secara modular atau yang sering kali disebut sebagai modular programming. Penulisan function pada JavaScript umumnya diletakkan pada bagian HEAD dalam dokumen HTML.

Bentuk umum dalam penulisan function adalah:

```
function nama_function() {  
    //statement-statement yang akan dieksekusi oleh function  
}
```

Ketika suatu function ingin digunakan yang perlu dilakukan adalah memanggil nama functionnya. Untuk lebih jelasnya bisa mempelajari contoh berikut ini

```
<HTML>  
  <HEAD>  
    <TITLE>Ini Judul</TITLE>  
    <SCRIPT LANGUAGE="JavaScript">  
      function salam() {  
        document.write("<H1>Hallo Apa Kabar?</H1>");  
      }  
    </SCRIPT>  
  </HEAD>  
  <BODY>  
    <SCRIPT LANGUAGE="JavaScript">  
      salam();  
      document.write("Header tersebut dihasilkan dari pemanggilan function salam");  
    </SCRIPT>  
  </BODY>  
</HTML>
```



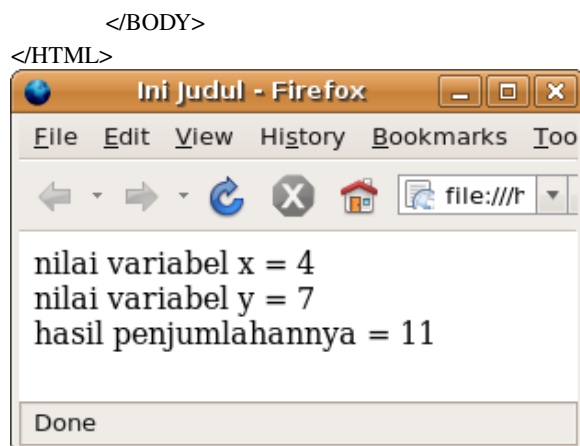
Suatu function juga bisa melewatkan suatu parameter ketika function tersebut dipanggil.

Bentuk umum penulisannya adalah:

```
function nama_function(parameter1, parameter2, ...) {
    //statement-statement yang akan dieksekusi oleh function
}
```

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
    <SCRIPT LANGUAGE="JavaScript">
      function penjumlahan(a, b) {
        var c;
        c = a + b;
        document.write("hasil penjumlahannya = " + c);
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      var x, y;
      x = 4; y = 7;
      document.write("nilai variabel x = " + x + "<BR>");
      document.write("nilai variabel y = " + y + "<BR>");
      penjumlahan(x, y);
    </SCRIPT>
```

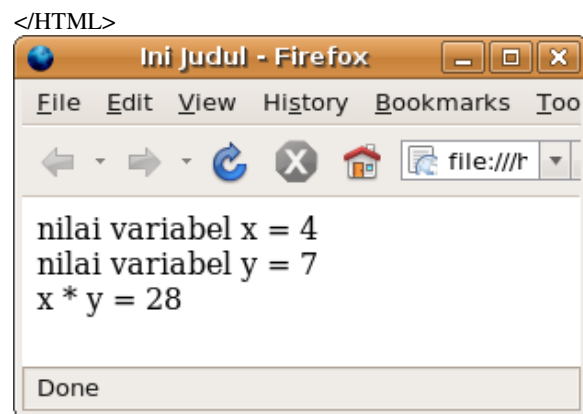


Suatu function juga bisa menghasilkan suatu nilai ketika function tersebut dipanggil. Untuk menjadikan function yang dibuat dapat menghasilkan / mengembalikan suatu nilai ketika dipanggil, yang perlu dilakukan adalah dengan menyertakan kata kunci return di depan nilai yang akan dikembalikan oleh function tersebut. Berikut ini adalah bentuk umum penulisannya.

```
function nama_function() {  
    //statement-statement yang akan dieksekusi oleh function  
    return nilai_kembali;  
}
```

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>  
<HEAD>  
  <TITLE>Ini Judul</TITLE>  
  <SCRIPT LANGUAGE="JavaScript">  
    function perkalian(a, b) {  
      var c;  
      c = a * b;  
      return c;  
    }  
  </SCRIPT>  
</HEAD>  
<BODY>  
  <SCRIPT LANGUAGE="JavaScript">  
    var x, y, z;  
    x = 4; y = 7;  
    document.write("nilai variabel x = " + x + "<BR>");  
    document.write("nilai variabel y = " + y + "<BR>");  
    z = perkalian(x, y);  
    document.write("x * y = " + z);  
  </SCRIPT>  
</BODY>
```



Mengenal Event

Event adalah kejadian yang terjadi pada suatu halaman web. Kejadian ini bisa bermacam-macam, ada kejadian penekanan tombol, kejadian pengubahan nilai pada textbox, dan bermacam-macam kejadian lainnya. Kejadian-kejadian tersebut bisa diarahkan untuk memicu pemanggilan suatu function tertentu. Konsep pemrograman di mana suatu function dipanggil berdasarkan suatu event tertentu dikenal dengan istilah event handling programming.

Event onClick

Event onClick merupakan kejadian penekanan tombol mouse (click) pada komponen form. Kejadian ini bisa berupa penekanan tombol, pemilihan checkbox maupun radio button, dan pemilihan suatu link, serta penekanan tombol submit ataupun reset.

Pengaturan terhadap event ini dilakukan dengan menuliskan atribut onClick pada tag HTML yang bersangkutan. Atribut ini dapat diberi nilai berupa nama function yang akan dipicu oleh event tersebut.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>  
<HEAD>  
  <TITLE>Ini Judul</TITLE>
```

```

<SCRIPT LANGUAGE="JavaScript">
    function salam() {
        window.alert("Selamat Datang\n" + formku.nama.value);
    }
</SCRIPT>
</HEAD>
<BODY>
    <FORM NAME="formku">
        <TABLE>
            <TR><TD>Nama:</TD><TD><INPUT TYPE="TEXT" NAME="nama"></TD></TR>
            <TR><TD></TD><TD><INPUT TYPE="BUTTON" VALUE="Proses" onClick="salam()"></TD></TR>
        </TABLE>
    </FORM>
</BODY>
</HTML>

```



Dalam dokument HTML untuk membuat suatu tombol bebas (selain tombol submit dan reset) dapat menggunakan tag `<INPUT>` dengan menyertakan atribut `TYPE` yang diberi nilai `BUTTON`.

Label pada tombol bisa diatur menggunakan atribut `VALUE`.

Untuk mengakses nilai yang tertampung dalam suatu komponen, yang perlu dilakukan adalah menyebutkan nama form tempat komponen tersebut berada, selanjutnya baru sebutkan nama komponennya dan diakhiri dengan kata kunci `value`. Penyebutan itu dilakukan dengan menyertakan tanda `."` (titik) sebagai pemisah. Hal ini berlaku juga ketika JavaScript ingin menempatkan nilai yang dihasilkan ke dalam suatu komponen dalam form HTML.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```

<HTML>
    <HEAD>
        <TITLE>Ini Judul</TITLE>
        <SCRIPT LANGUAGE="JavaScript">
            function penjumlahan(a, b) {

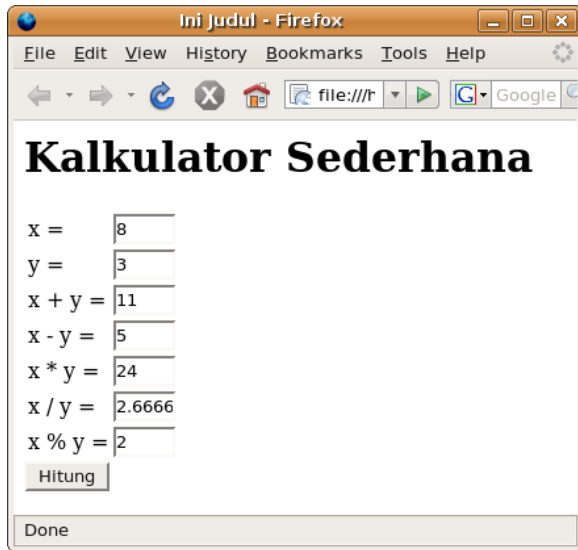
```

```

        return (a + b);
    }
    function pengurangan(a, b) {
        return (a-b);
    }
    function perkalian(a, b) {
        return (a * b);
    }
    function pembagian(a, b) {
        return (a / b);
    }
    function sisa_bagi(a, b) {
        return (a % b);
    }
    function proses() {
        var p, q;
        p = parseInt(form_hitung.x.value);
        q = parseInt(form_hitung.y.value);
        form_hitung.jumlah.value = penjumlahan(p,q);
        form_hitung.kurang.value = pengurangan(p,q);
        form_hitung.kali.value = perkalian(p,q);
        form_hitung.bagi.value = pembagian(p,q);
        form_hitung.modulus.value = sisa_bagi(p,q);
    }
</SCRIPT>
</HEAD>
<BODY>
    <H1>Kalkulator Sederhana</H1>
    <FORM NAME="form_hitung">
        <TABLE>
            <TR>
                <TD>x =</TD>
                <TD><INPUT TYPE="TEXT" NAME="x" SIZE="3"></TD>
            </TR>
            <TR>
                <TD>y =</TD>
                <TD><INPUT TYPE="TEXT" NAME="y" SIZE="3"></TD>
            </TR>
            <TR>
                <TD>x + y =</TD>
                <TD><INPUT TYPE="TEXT" NAME="jumlah" SIZE="3" READONLY></TD>
            </TR>
            <TR>
                <TD>x - y =</TD>
                <TD><INPUT TYPE="TEXT" NAME="kurang" SIZE="3" READONLY></TD>
            </TR>
            <TR>
                <TD>x * y =</TD>
                <TD><INPUT TYPE="TEXT" NAME="kali" SIZE="3" READONLY></TD>
            </TR>
            <TR>
                <TD>x / y =</TD>
                <TD><INPUT TYPE="TEXT" NAME="bagi" SIZE="3" READONLY></TD>
            </TR>
            <TR>
                <TD>x % y =</TD>
                <TD><INPUT TYPE="TEXT" NAME="modulus" SIZE="3" READONLY></TD>
            </TR>
        </TABLE>
        <INPUT TYPE="BUTTON" VALUE="Hitung" onClick="proses()">
    </FORM>
</BODY>

```

</HTML>



Atribut READONLY yang terdapat pada suatu textbox menjadikan nilai pada textbox tersebut tidak dapat diubah lewat halaman web browser.

Selain atribut READONLY terdapat atribut lain yaitu atribut DISABLED yang berfungsi untuk menonaktifkan suatu komponen form.

Event onLoad dan onUnload

Event onLoad berkaitan dengan kejadian pembukaan suatu file dokumen HTML ke dalam web browser sedangkan event onUnload berkaitan dengan kejadian penutupan suatu file dokumen HTML oleh suatu web browser. Kejadian penutupan file dokumen HTML yang dimaksud di sini bukan hanya penutupan web browser tetapi lebih mengarah ke pergantian pembukaan file dokumen HTML. Contoh paling nyata dari kejadian ini adalah ketika terjadi pemilihan link di mana web browser akan menutup file dokumen HTML yang saat ini sedang dibuka dan kemudian membuka file dokumen HTML yang lain sesuai dengan yang disebutkan pada atribut HREF pada link yang bersangkutan.

Baik event onLoad maupun event onUnLaod keduanya hanya dapat disertakan sebagai atribut pada tag <BODY>.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

nama file: hal1.html

```
<HTML>
<HEAD>
  <TITLE>Ini Judul</TITLE>
  <SCRIPT LANGUAGE="JavaScript">
    function salam() {
      window.alert("Selamat Datang");
    }
    function bye() {
      window.alert("Selamat Tinggal");
    }
  </SCRIPT>
</HEAD>
<BODY onLoad="salam()" onUnload="bye()">
  <H1>Mengenal Event onLoad dan onUnload</H1>
  <A HREF="hal2.html">ke halaman2</A>
</BODY>
```

</HTML>

nama file: hal2.html

```
<HTML>
<HEAD>
  <TITLE>Ini Judul</TITLE>
  <SCRIPT LANGUAGE="JavaScript">
    function salam() {
      window.alert("Selamat Datang di halaman 2");
    }
  </SCRIPT>
</HEAD>
<BODY onLoad="salam()">
  <H1>Halaman 2</H1>
</BODY>
```

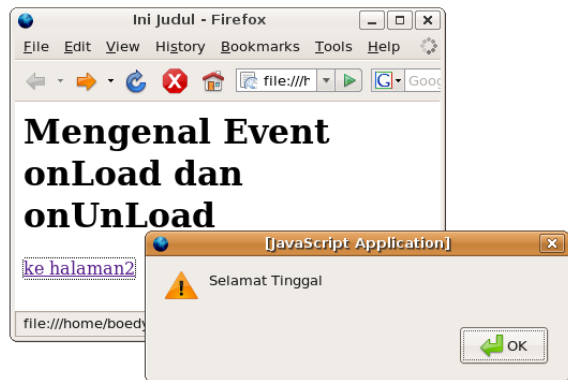
</HTML>



1) Ketika file hal1.html dibuka oleh web browser



2) Ketika web browser ditutup



3) Ketika link dipilih



4) Ketika memasuki hal2.html

Event onFocus dan onBlur

Event onFocus berkaitan dengan kejadian mengarahkan fokus kursor ke suatu komponen form tertentu. Contoh nyata dari kejadian ini adalah ketika melakukan pemindahan kursor ke dalam suatu textbox baik itu dilakukan dengan menggunakan bantuan mouse ataupun menggunakan tombol tabulasi pada keyboard.

Sedikit berbeda dengan event onFocus, event onBlur berkaitan dengan kejadian ketika kursor beralih meninggalkan suatu komponen form tertentu.

Baik event onFocus maupun event onBlur keduanya bisa diimplementasikan pada komponen textbox dan text area.

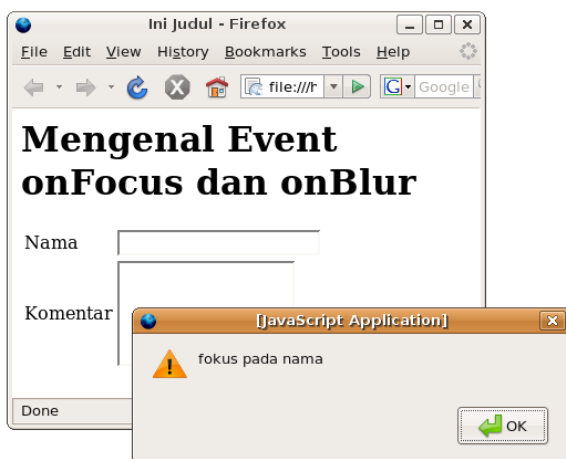
Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```
<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
    <SCRIPT LANGUAGE="JavaScript">
      function fokus_nama() {
        window.alert("fokus pada nama");
      }
      function fokus_komentar() {
        window.alert("fokus pada komentar");
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM>
      <INPUT TYPE="text" VALUE="Nama" />
      <INPUT TYPE="text" VALUE="Komentar" />
    </FORM>
  </BODY>
</HTML>
```

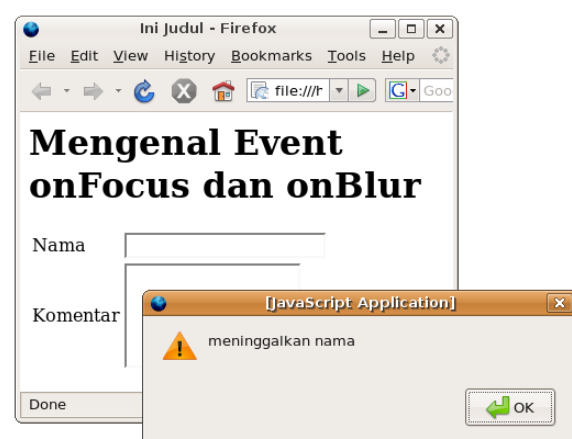
```

    }
    function bye_nama() {
        window.alert("meninggalkan nama");
    }
    function bye_komentar() {
        window.alert("meninggalkan komentar");
    }
</SCRIPT>
</HEAD>
<BODY>
    <H1>Mengenal Event onFocus dan onBlur</H1>
    <FORM NAME="formku">
    <TABLE>
        <TR>
            <TD>Nama</TD>
            <TD><INPUT TYPE="TEXT" NAME="nama" onFocus="fokus_nama()" onBlur="bye_nama()"></TD>
        </TR>
        <TR>
            <TD>Komentar</TD>
            <TD><TEXTAREA NAME="kometar" COLS="20" ROWS="5" onFocus="fokus_komentar()"
                onBlur="bye_komentar()"></TEXTAREA></TD>
        </TR>
    </TABLE>
    </FORM>
</BODY>
</HTML>

```



Ketika fokus kursor pada textbox nama



Ketika fokus kursor meninggalkan textbox nama

Event onChange

Event onChange merupakan suatu kejadian perubahan nilai pada suatu komponen. Contoh nyata dari event ini adalah pada kejadian pengisian atau perubahan nilai pada textbox, text area, combo box, dan listbox. Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```

<HTML>
    <HEAD>
        <TITLE>Ini Judul</TITLE>
        <SCRIPT LANGUAGE="JavaScript">
            function ubah_nama() {
                window.alert("Nama Anda " + formku.nama.value);
            }
            function ubah_kota() {
                window.alert("Anda berada di kota " + formku.kota.value);
            }
        </SCRIPT>

```

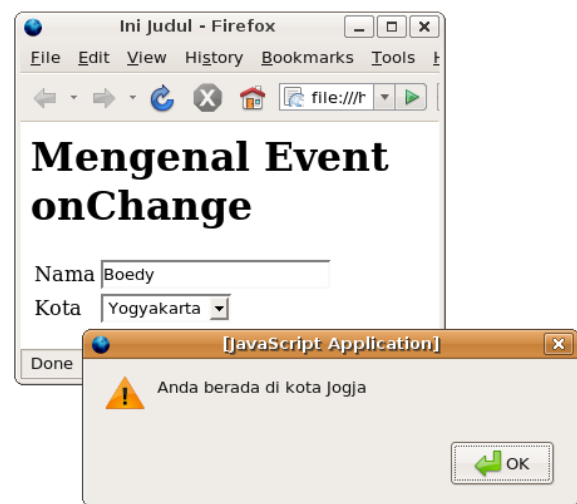
```

</HEAD>
<BODY>
  <H1>Menenal Event onChange</H1>
  <FORM NAME="formku">
    <TABLE>
      <TR>
        <TD>Nama</TD>
        <TD><INPUT TYPE="TEXT" NAME="nama" onChange="ubah_nama()"></TD>
      </TR>
      <TR>
        <TD>Kota</TD>
        <TD><SELECT NAME="kota" onChange="ubah_kota()">
          <OPTION VALUE="Jakarta">Jakarta</OPTION>
          <OPTION VALUE="Bandung">Bandung</OPTION>
          <OPTION VALUE="Jogja">Yogyakarta</OPTION>
        </SELECT></TD>
      </TR>
    </TABLE>
  </FORM>
</BODY>
</HTML>

```



Setelah melakukan perubahan pada textbox nama



Setelah melakukan perubahan pada combo box kota

Event onSelect

Event onSelect berkaitan dengan kejadian pemilihan teks pada suatu komponen form seperti pemilihan teks yang berada pada suatu textbox. Event ini bisa diterapkan pada textbox, textarea, combo box, dan list box. Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```

<HTML>
  <HEAD>
    <TITLE>Ini Judul</TITLE>
    <SCRIPT LANGUAGE="JavaScript">
      function select_nama() {
        window.alert("Anda memilih teks \n pada textbox nama");
      }
      function select_komentar() {
        window.alert("Anda memilih teks \n pada textarea komentar");
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <H1>Menenal Event onSelect</H1>

```

```

<FORM NAME="formku">
<TABLE>
    <TR>
        <TD>Nama</TD>
        <TD><INPUT TYPE="TEXT" NAME="nama" onSelect="select_nama()"></TD>
    </TR>
    <TR>
        <TD>Komentar</TD>
        <TD>
            <TEXTAREA NAME="komentar" COLS="20" ROWS="5" onSelect="select_kota()">
            </TEXTAREA>
        </TD>
    </TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```



Event onMouseOver dan onMouseOut

Event onMouseOver dan onMouseOut berkaitan dengan kejadian pergerakan kursor mouse pada suatu link.

Event onMouseOver terjadi ketika kursor mouse berada di atas suatu link sedangkan event onMouseOut terjadi ketika kursor mouse bergerak meninggalkan suatu link.

Untuk lebih jelasnya bisa mempelajari contoh berikut ini:

```

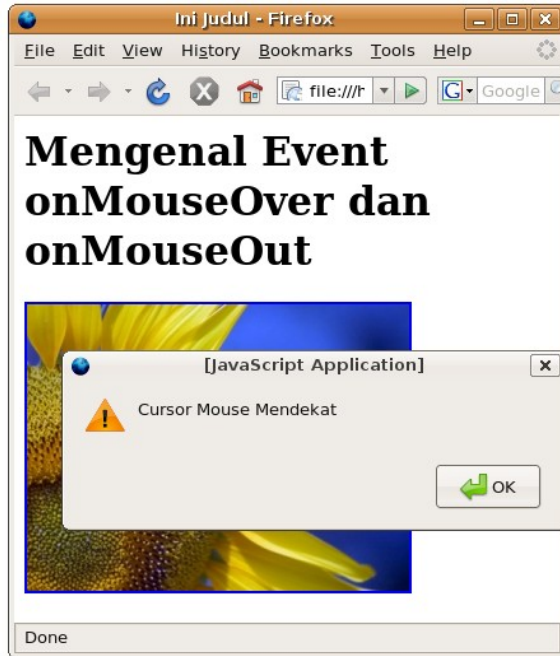
<HTML>
    <HEAD>
        <TITLE>Ini Judul</TITLE>
        <SCRIPT LANGUAGE="JavaScript">
            function dekat() {
                window.alert("Cursor Mouse Mendekat");
            }
            function jauh() {
                window.alert("Cursor Mouse Menjauh");
            }
        </SCRIPT>
    </HEAD>
    <BODY>
        <A href="#" onMouseOver="dekat()" onMouseOut="jauh()">Klik Disini</A>
    </BODY>
</HTML>

```

```

    }
</SCRIPT>
</HEAD>
<BODY>
    <H1>Menenal Event onMouseOver dan onMouseOut</H1>
    <A HREF="http://ubuntu.com" onMouseOver="dekat()" onMouseOut="jauh()"><IMG SRC="pic1.jpg"></A>
</BODY>
</HTML>

```



Mengenal Array

Bagi setiap orang yang pernah belajar suatu bahasa pemrograman tertentu pasti pernah mengenal konsep array. Array sering kali digambarkan sebagai suatu variabel yang mampu menampung banyak nilai oleh karenanya sering kali juga dikenal dengan sebutan variabel array.

Deklarasi Array

Dalam JavaScript terdapat dua cara untuk mendeklarasikan suatu array.

Bentuk umum dari cara yang pertama adalah:

```
var nama_variabel_array = new Array(isi_array);
```

contoh:

```
var hobi = new Array("memancing","membaca","melukis","menyanyi");
```

Pada contoh tersebut dideklarasikan suatu variabel array bernama hobi. Variabel array ini berisi nilai-nilai bertipe data String, yaitu "memancing", "membaca", "melukis", dan "menyanyi".

Pada cara pendeklarasian seperti ini suatu variabel array akan langsung diberi nilai awal dan ukuran dari variabel array yang terbentuk akan disesuaikan dengan banyaknya nilai awal yang diisikan.

Bentuk umum dari cara yang kedua adalah:

```
var nama_variabel_array = new Array(ukuran);
```

contoh:

```
var negara = new Array();
```

Pada cara pendeklarasian kedua seperti di atas akan menghasilkan suatu variabel array bernama negara.

Yang menarik dari array yang dimiliki oleh Java Script adalah ukurannya yang dinamis. Dalam Java Script seseorang yang ingin membuat suatu array tidak perlu menentukan ukuran array yang akan dibuat. Ukuran

array dalam Java Script akan secara otomatis menyesuaikan dengan jumlah data yang ditampung di dalamnya. Untuk menyimpan data pada suatu array ataupun untuk mengakses data yang tersimpan di dalam suatu array dapat dilakukan dengan menyebutkan nama array dan diikuti dengan index tempat data tersebut berada dalam array yang bersangkutan.

Misalnya pada array negara akan dimasukkan data-data negara: “indonesia”, “jepang”, “belanda”, dan “malaysia”, maka dapat dilakukan dengan:

```
negara[0] = “indonesia”;  
negara[1] = “jepang”;  
negara[2] = “belanda”;  
negara[3] = “malaysia”;  
var coba = negara[1];
```

Pada contoh di atas, array negara dengan index 0 akan berisi data string “indonesia”, dan pada index 2 akan berisi data string “belanda”.

Variabel coba pada contoh di atas akan berisi data yang dicopykan dari array negara index 1. Dengan kata lain variabel coba pada contoh di atas akan berisi data string “jepang”.

Suatu array dalam Java Script juga bisa menggunakan index berupa String dan tidak terbatas pada penggunaan index numerik.

```
var hobi = new Array();  
hobi[“satu”] = “bernyanyi”;  
hobi[“andi”] = “membaca”;
```

Pada contoh di atas akan dihasilkan suatu variabel array bernama hobi. Array hobi tersebut akan menampung dua buah data string, yaitu data string “bernyanyi” dan data string “membaca”. Data string “bernyanyi” akan disimpan dalam array hobi pada index string “satu” sedangkan data string “membaca” akan disimpan dalam array hobi pada index string “andi”.

Untuk lebih jelasnya bisa mempelajari contoh lengkap berikut ini: