

Pengantar Java Script Komputer IT-2

Target Pencapaian

Semua siswa dapat memahami bagaimana membuat aplikasi dengan menggunakan Java Script

1.1. 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.

1.2. 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>.

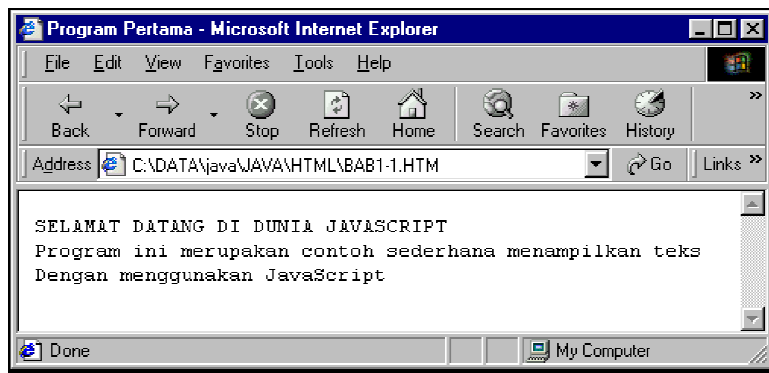
Contoh Script.

1. `<script language="javascript">`
2. `//program anda masukan disini`
3. `</script>`

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.

1.3. Program Pertama Java Script

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



Gambar 1.1. Menampilkan Teks

Contoh Script.

1. `<html>`
2. `<head>`
3. `<title>Program Pertama</title>`
4. `</head>`
5. `<body bgcolor="#FFFFFF">`
6. `<script language="JavaScript">`
7. `document.writeln("<PRE>");`
8. `document.writeln("SELAMAT DATANG DI DUNIA JAVASCRIPT ");`
9. `document.writeln("Program ini merupakan contoh sederhana menampilkan teks");`
10. `document.writeln("Dengan menggunakan JavaScript");`

```
11. document.writeln("</PRE>");
12. </script>
```

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 1.2. Teks Header Pada JavaScript

Contoh Script.

```
1. <html>
2. <script language="JavaScript">
3. document.writeln("<CENTER>");
4. document.writeln("<H1>UNIVERSITAS KOMPUTER INDONESIA </H1>");
5. document.writeln("<H2>JL.DIPATIUKUR 112 BANDUNG</H2>");
6. document.writeln("</CENTER>");
7. </script>
8. </html>
```

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:

Contoh Menuliskan Komentar

1. `// ini komentar, jadi perintah ini tidak akan ditampilkan`
2. `/* yang ini juga sama komentar */`

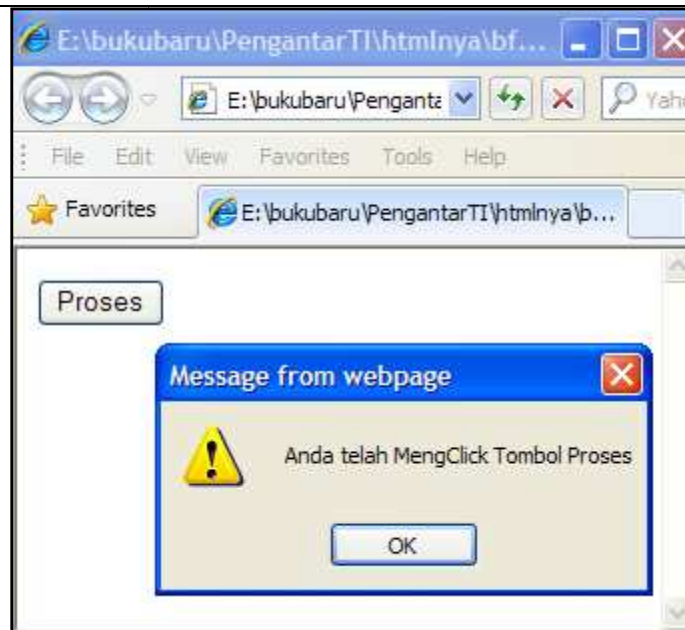
1.5. Event

Event dan event handler merupakan hal yang sangat penting dalam pemrograman JavaScript. Event adalah sesuatu yang terjadi karena aksi user. Contohnya jika user meng-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**:

Menangani Event:

1. `<form>`
2. `<input type="button" value="Proses" onClick="alert('Anda telah MengClick Tombol Proses')">`
3. `</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. Mari kita tulis sebuah script yang mengeluarkan teks tertentu sebanyak tiga kali. Perhatikan contoh berikut ini:

Contoh Membuat Fungsi:

```
1. <form>
2. <script language="JavaScript">
3. document.write("Belajar Java Script<br>");
4. document.write("Belajar Java Script<br>");
5. document.write("Belajar Java Script<br>");
6. document.write("Belajar Java Script<br>");
7. </script>
8. </html>
```

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:

Contoh Membuat Fungsi:

```
1. <html>
2. <script language="JavaScript">
3. function myFunction()
4. {
5.     document.write("Belajar Java Script<br>");
6. }
7. myFunction();
8. myFunction();
9. myFunction();
10. myFunction();
11. </script>
12. </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.

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

Contoh Membuat Fungsi:

```
1. <html>  
2. <script language="JavaScript">  
3. function calculation() {  
4.     var x = 12;  
5.     var y = 5;  
6.     var result = x + y;  
7.     alert(result);  
8. }  
9. </script>  
10. <form>  
11. <input type="button" value="Calculate" onClick="calculation()">  
12. </form>
```

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 11. Perintah **alert(result)** artinya sama dengan **alert(11)**, yaitu popup window akan muncul dengan isi angka 11.

1.7. 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, tulislah 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;
```

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 "Halo". "Mei", "Juli", "Ah Kamu", "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 menunjukan 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)

1.9. Operator

Operator pada JavaScript dibagi menjadi enam, yaitu

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

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 kekakan 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).
```

1.9.4. 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

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.

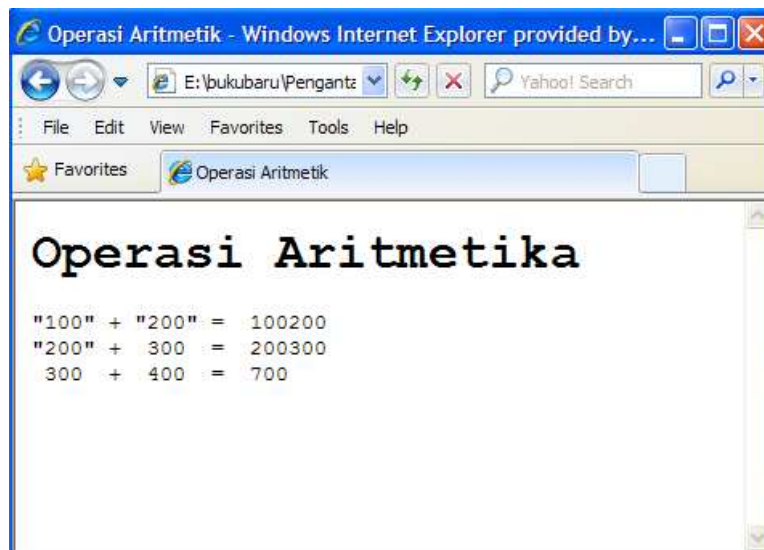
Contoh:

```
Nama = "Java" + "Script";
Akan menghasilkan "JavaScript" pada variabel Nama
```

Penggunaan Operator String

```
<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 1.4. Output Program

1.10. Memasukan Data

Untuk memasukan data dari keyboard dapat dilakukan dengan menggunakan perintah input dengan type objek yang berbeda-beda

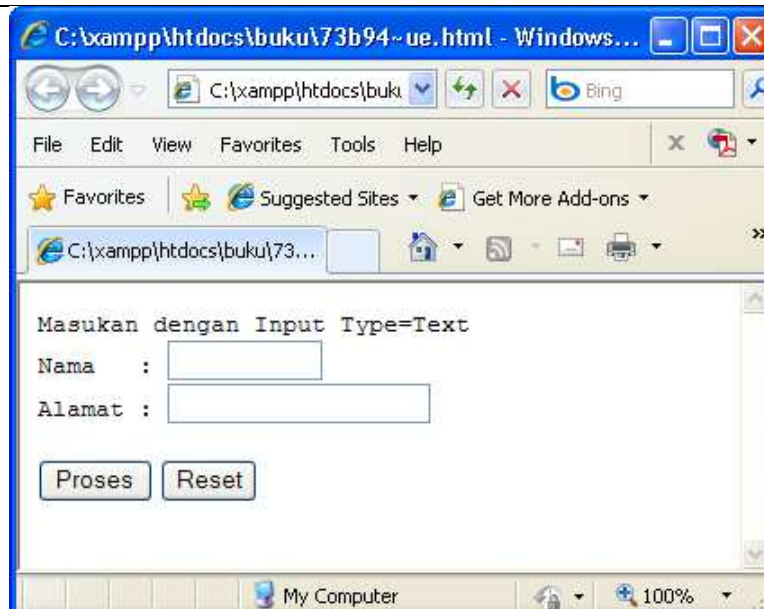
1.10.1. Objek Text

Dalam Halaman HTML dimungkinkan juga untuk memasukan data dari keyboard (user) yang mengakses ke halaman Anda. Contoh aplikasi seperti ini banyak diterapkan dalam Web Interaktif, dengan menggunakan CGI ataupun PHP

Untuk memasukan data dapat digunakan objek text yang telah disediakan dengan contoh sintak seperti berikut:

Penggunaan Objek Text

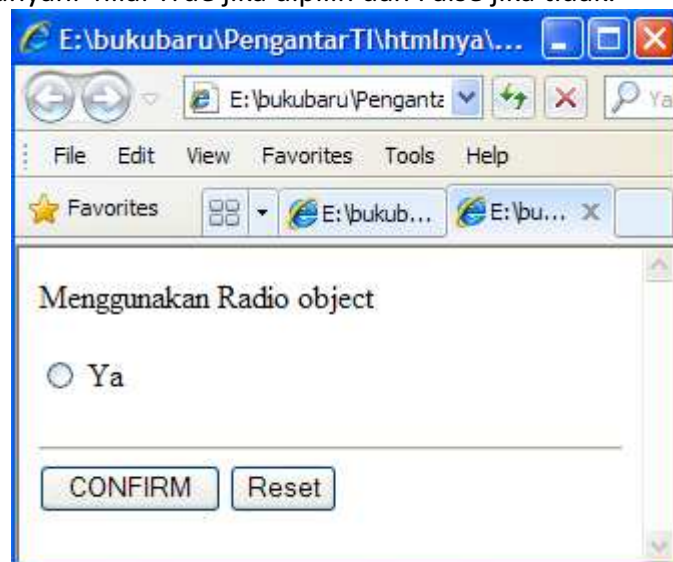
```
<form name=form>
<script>
function proses()
{
var nama2=document.form.nama.value;
var alamat2=document.form.alamat.value;
alert("Nama : "+nama2+ " Alamat: " +alamat2);
}
</script>
<PRE>
Masukan dengan Input Type=Text
Nama : <input type="text" size="10" name="nama">
Alamat : <input type="text" size="20" name="alamat">
</PRE>
<input type=button value=Proses onclick="proses()">
<input type=reset>
```



Gambar 1.5. Memasukan Data dengan Text Objek

1.10.2. Objek Radio

Objek radio menyimpan informasi tentang tombol radio. Karena selalu berupa array, untuk mengakses satu tombol radio digunakan `radio[indeks]`. Disamping itu objek radio juga mempunyai nilai `True` jika dipilih dan `False` jika tidak.



Gambar 1.6. Objek Radio

Penggunaan Objek Radio

```
<html>
<script language="JavaScript">
function radio_box(form)
{
    var ket="";
    if (form.pilih.checked == true)
        ket="Memilih Ya ";
    else
        ket= "Tidak memilih ";
    alert(' Anda ' +ket);
}
</script>
Menggunakan Radio object
<form>
<input type="radio" name="pilih" value="Memilih Ya"> Ya </p>
<hr>
<input type="button" value="CONFIRM"
onclick="radio_box(this.form)"> <input type="reset"
value="Reset">
</form>
```

Dari hasil eksekusi program diatas ada dua buah output yaitu:

1. Jika anda tidak meng-click Radio Button Ya, maka akan ditampilkan pesan *Anda Tidak memilih*
2. Jika anda meng-click Radio Button Ya, maka akan ditampilkan pesan *Anda Anda memilih ya.*

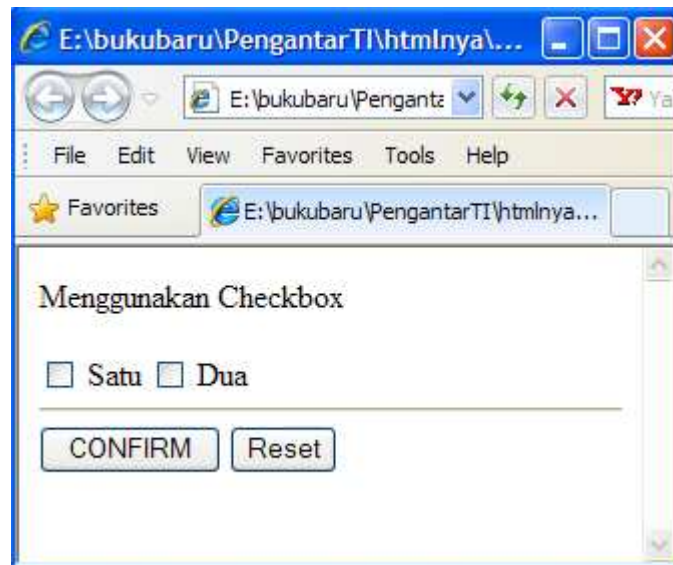
Berikut hasil output program diatas.



Gambar 1.1. Aplikasi Objek Radio

1.10.3. Objek Checkbox

Objek checkbox menyimpan informasi tentang elemen form yang berupa kotak cek. Penggunaanya hampir sama dengan objek Radio.



Gambar 1.8. Objek Checkbox

Penggunaan Objek Checkbox

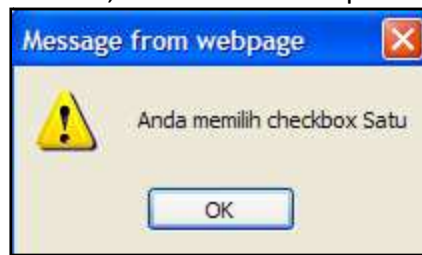
```
<html>
<script language="JavaScript">
function check_box(form)
{
    var ket1="";
    var ket2="";

    if (form.satu.checked == true) ket1="Satu";
    if (form.dua.checked == true) ket2="Dua";
    alert('Anda memilih checkbox ' + ket1 + ' ' + ket2);
}
</script>
```

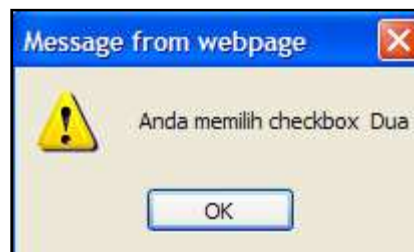
Menggunakan Checkbox

```
<form>
  <input type="checkbox" name="satu"> Satu
  <input type="checkbox" name="dua"> Dua
  <hr>
  <input type="button" value="CONFIRM"
  onclick="check_box(this.form)">
  <input type="reset" value="Reset">
</form>
```

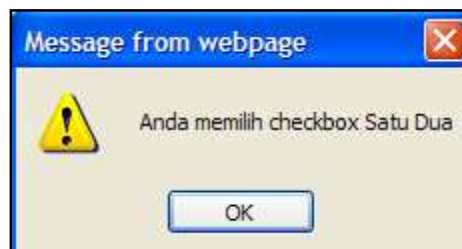

Dari hasil eksekusi program diatas, akan ditampilkan output seperti berikut:
Jika anda meng-click Checkbox Satu, maka akan ditampilkan output



Jika anda meng-click Checkbox Dua, maka akan ditampilkan output

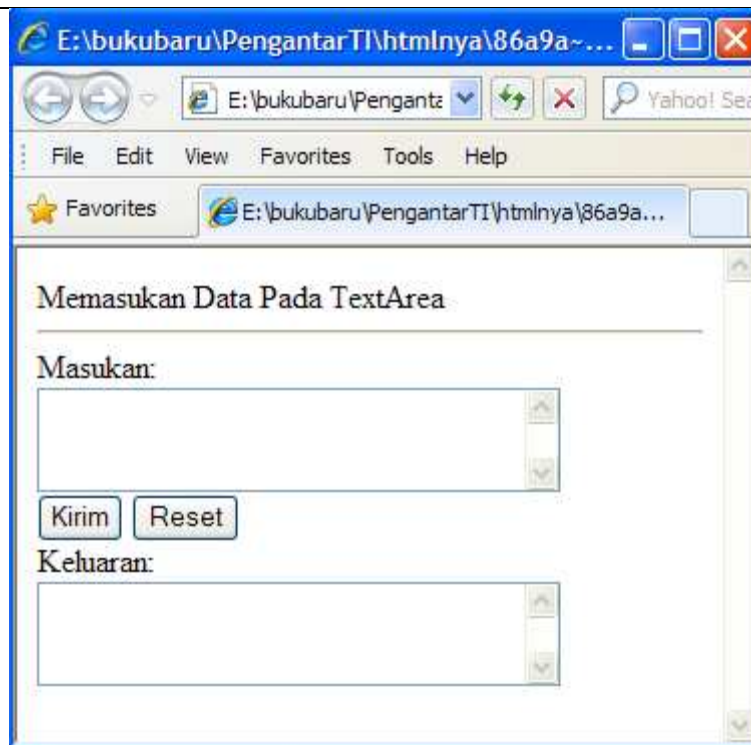


Jika anda meng-click Checkbox Satu dan Dua, maka akan ditampilkan output



1.10.4. Objek TextArea

Objek textarea digunakan apabila anda ingin memasukan data yang memiliki jumlah karakter yang tidak terbatas, dan menyimpan informasi yang dimasukan tadi kedalam form yang berupa kotak teks dengan banyak baris.

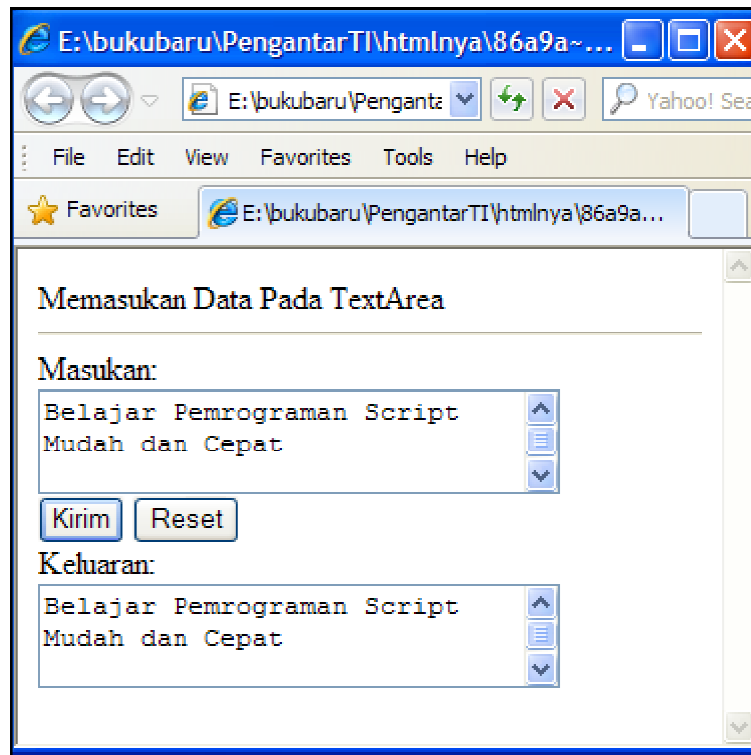


Gambar 1.9. Objek Textarea

Penggunaan Objek Textarea

```
<html>
<script language="JavaScript">
function EvenTekan()
{
    var ketstr=(document.fForm.ket.value);
    document.fForm.ketstr.value=ketstr;
}
</script>
<form name="fForm">
    Memasukan Data Pada TextArea
    <hr>
    Masukan:
    <br>
    <textarea name="ket" rows="3" cols="30">
    </textarea>
    <br>
    <input type="button" value="Kirim" onclick="EvenTekan()">
    <input type="reset" value="Reset">
    <BR>
```

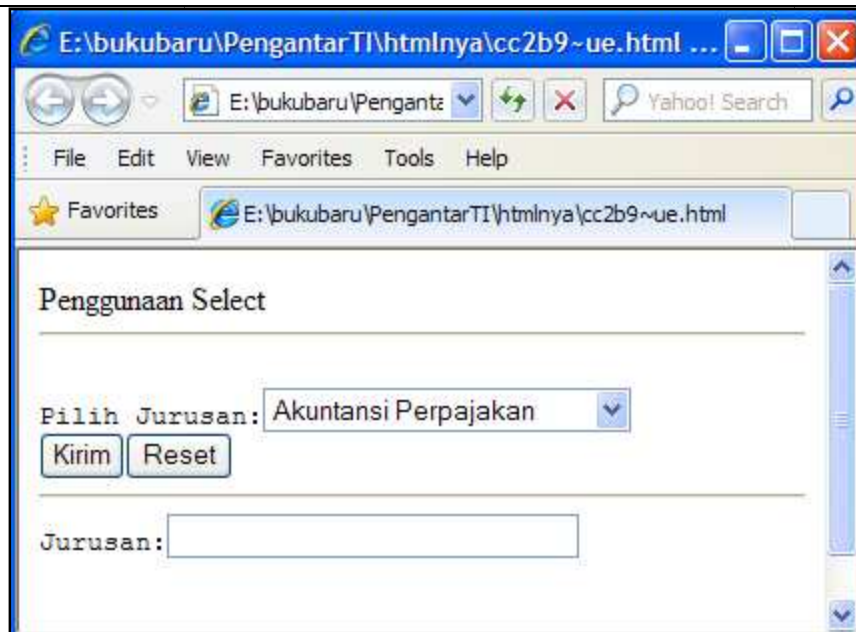
```
Keluaran:<br>
<textarea name="ketstr" rows="3" cols="30">
</textarea>
</form>
```



Gambar 1.10. Objek Textarea

1.10.5. Objek Select

Objek Select digunakan untuk menyimpan informasi tentang elemen form yang berupa kotak daftar pilihan, anda dapat memilih salah satu pilihan dari daftar yang ada di kotak tersebut.



Gambar 1.11. Objek Select

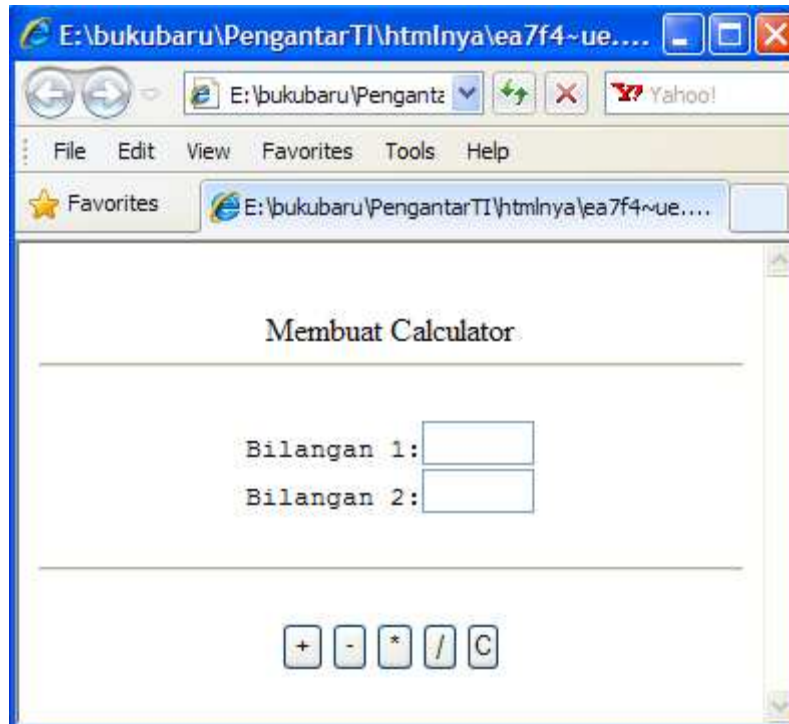
Penggunaan Objek Select

```
<html>
<script language="javascript">
function EvenTekan()
{
    var jurusanstr = (document.fform.jurusan.value);
    document.fform.jurusanstr.value=jurusanstr;
}
</script>
<form name="fform">
Penggunaan Select
<hr><pre>
Pilih Jurusan:<select name="jurusan" size="1">
    <option value="Akuntansi Perpajakan">Akuntansi Perpajakan
    <option value="Manajemen Informatika">Manajemen Informatika
    <option value="Teknik Informatika">Teknik Informatika
    <option value="Public Relation">Public Relation
    <option value="Akuntansidan Perpajakan">Akuntansi dan Perpajakan
</select>
<input type="button" value="Kirim" onclick="EvenTekan()"><input type="reset">
<hr>
```

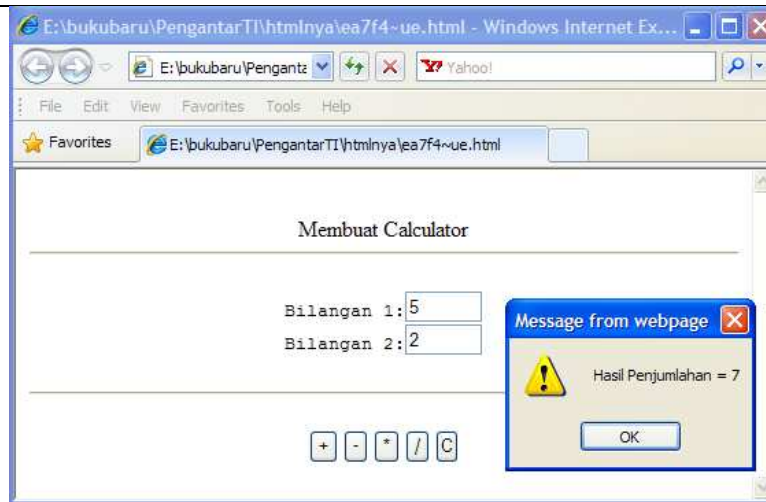
```
Jurusan:<input type="text" size="30" name="jurusanstr">  
</form>
```

Latihan

Dengan menggunakan JavaScript, buatlah program untuk menampilkan proses perhitungan perkalian, pembagian, pengurangan dan penjumlahan, seperti berikut:



Gambar 1.12. Form Input Kalkulator



Gambar 1.13. Output Jika tombol + ditekan

Buatlah Halaman seperti berikut, yang telah mencakup semua materi yang ada di dalam bab ini:

Formulir Pendaftaran Calon Mahasiswa Baru	
Nama :	<input type="text"/>
Alamat :	<input type="text"/>
Asal Sekolah :	<input type="text" value="SMA"/> ▼
Jurusan dipilih:	<input type="checkbox"/> Teknik Informatika <input type="checkbox"/> Manajemen Informatik
Jenjang :	<input type="radio"/> Diploma <input type="radio"/> Sarjana
<input type="button" value="Kirim"/> <input type="button" value="Reset"/>	
Nama :	<input type="text"/>
Alamat :	<input type="text"/>
Asal Sekolah:	<input type="text"/>
Jurusan :	<input type="text"/>
Jenjang :	<input type="text"/>

Gambar 1.14. Membuat Form Lengkap

Jika Anda Click tombol kirim,. Maka dihalaman bawah akan ditampilkan data sesuai dengan yang di inputkan dari form sebelah atas

1.11. Percabangan

Seperti dalam bahasa pemrograman lainnya dalam JavaScript juga ada percabangan. Pada dasarnya dalam JavaScript terdapat dua macam pernyataan percabangan, yaitu pernyataan if..else dan switch.

1.11.1. if..else

Pernyataan ini digunakan untuk menguji sebuah kondisi dan kemudian mengeksekusi pernyataan tertentu bila kondisi tersebut terpenuhi, dan mengeksekusi pernyataan lain bila kondisi tersebut tidak terpenuhi.

Contoh:

```
If..(kondisi)
{
    //lakukan pernyataan yang ada disini
    //jika kondisi terpenuhi
}
else
{
    //lakukan pernyataan yang ada disini
    //jika kondisi tidak terpenuhi
}
```

1.11.2. if..else if

Pada kasus tertentu mungkin Anda akan meletakkan pernyataan if lain setelah else.

Contoh:

```
If (x>0)
{
    document.writeln(' X adalah Bilangan positif');
}
else if(x<0)
{
    document.writeln(' X adalah bilangan negatif');
}
else
{
    document.writeln(' X adalah Nol ');
}
```

1.11.3. switch

Kegunaan pernyataan ini sama dengan yang ada pada C/C++ dan Java, yaitu untuk menangani percabangan majemuk, Dengan kata lain pernyataan switch dapat digunakan untuk menyederhanakan pernyataan if..else if yang terlalu banyak.

Contoh.

```
if (buffer==0) value ="Minggu";  
else if (buffer==1) value ="Senin";  
else if (buffer==2) value ="Selasa";  
else if (buffer==3) value ="Rabu";  
else if (buffer==4) value ="Kamis";  
else if (buffer==5) value ="Jumat";  
else if (buffer==6) value ="Sabtu";
```

Dapat disederhanakan dengan pernyataan switch menjadi sebagai berikut:

```
Switch (buffer)  
{  
case 0: value="Minggu";break;  
case 1: value="Senin"; break;  
case 2: value="Selasa"; break;  
case 3: value="Rabu"; break;  
case 4: value="Kamis"; break;  
case 5: value="Jumat"; break;  
default: value="Sabtu"  
}
```


1.12. Pengulangan

Perintah pengulangan biasanya digunakan apabila anda ingin menampilkan satu atau beberapa baris perintah secara berulang.

1.12.1. Pengulangan For

Pengulangan dengan For paling sering digunakan jika anda sudah tahu akhir dari perulangan tersebut.

Contoh:

```
for(nilaiawal;kondisi;penambahan)
{
    //ulang pernyataan ini;
}
```

Contoh dalam program:

```
For(x=1;x<=10;x++)
    document.writeln("Java Script Ok");
```

Dengan pernyataan ini artinya akan menampilkan tulisan "Java Script Ok" sebanyak sepuluh kali kelayar

1.12.2. Pengulangan while

Pengulangan ini digunakan apabila anda belum tahu pasti berapa banyak pengulangan akan dilakukan. Berakhirnya suatu perulangan dalam while ditentukan oleh suatu kondisi. Bila kondisi sudah terpenuhi maka pengulangan akan dihentikan.

Contoh:

```
while (kondisi)
{
    //ulang pernyataan ini;
}
```

Contoh dalam program:

```
while (x>0)
{
    x=x - 1;
    y=y + 1;
}
```

1.12.3. Pengulangan do..while

Pengulangan ini hampir sama dengan while, digunakan apabila anda belum tahu pasti berapa banyak pengulangan akan dilakukan. Bedanya dengan while, pernyataan do..while pengujian kondisi dilakukan pada akhir kalang.

Contoh:

```
do
{
    //ulang pernyataan ini;
} while (kondisi);
```

Contoh dalam program:

```
do
{
    x=x - 1;
    y=y + 1;
} while (x>)
```

1.13. Fungsi

Fungsi merupakan bagian program yang dapat melakukan tugas tertentu. Beberapa fungsi juga ada yang dapat mengembalikan nilai, contohnya adalah fungsi parseFloat yang sudah digunakan sebelumnya.

1.13.1. Fungsi Buatan Sendiri

Dalam contoh pembahasan sebelumnya sebenarnya Anda sudah menggunakan fungsi. Untuk mendefinisikan fungsi harus diawali dengan kata kunci function.

Sintaknya adalah sebagai berikut:

Function namafungsi(param1,param2,.....)

```
{
    //pernyataan
}
```



Gambar 1.15. Membuat Fungsi.

Program Membuat Fungsi Sendiri

```
<html>
<head>
<title>Membuat Fungsi Sendiri</title>
</head>
<body bgcolor="#FFFFFF">
<p><script language="JavaScript">
function Halo()
{
    document.writeln("Halo saya adalah fungsi buatan");
}

function Tulis(Teks)
{
    document.writeln(Teks);
}

function Kali(a,b)
{
    return (a*b);
}

document.writeln("<PRE>");
document.writeln("<h1>Membuat Fungsi Sendiri</h1>");
Halo();
Tulis("Ini adalah fungsi dengan parameter");
var A = Kali(10,5);
Tulis(A);
document.writeln("</PRE>");
</script></p>
</body>
</html>
```

1.13.2. Fungsi Bawaan

Dalam JavaScript telah disediakan beberapa fungsi bawaan yang akan sangat berguna sekali bagi Anda, disini yang akan dibahas yaitu mengenai fungsi eval, parseInt, parseFloat, isNaN

- **Fungsi eval**

Digunakan untuk mengevaluasi ekspresi atau mengeksekusi pernyataan JavaScript. Contoh.

eval(str)

fungsi eval digunakan sebagai pengevaluasi ekspresi seperti contoh berikut:

```
var A=eval("10*2+3");  
memberikan nila 23 ke variabel A  
var garis = "document.write("<HR>");  
jika anda tuliskan seperti ini:  
eval(garis);  
maka jendela browser akan ditampilkan garis
```

- **Fungsi parseInt dan parseFloat**

Kedua fungsi ini hampir sama, keduanya meminta sebuah parameter bertipe string dan mengkonversikannya ke numerik.

Contoh

```
parseInt(str,[radiks])
```

```
parseFloat(str);
```

contoh penggunaan

```
var A = parseInt("173",8); // 123 bilangan basis 8
```

```
var B = parseFloat("3.14"); // 3.14
```

```
var C = parseFloat(0.314E1"); //3.14
```

- **Fungsi isNaN**

Digunakan untuk menguji apakah suatu variabel adalah bilangan atau bukan, jika bilangan maka akan mengembalikan nilai true, jika bukan maka bernilai false

Contoh

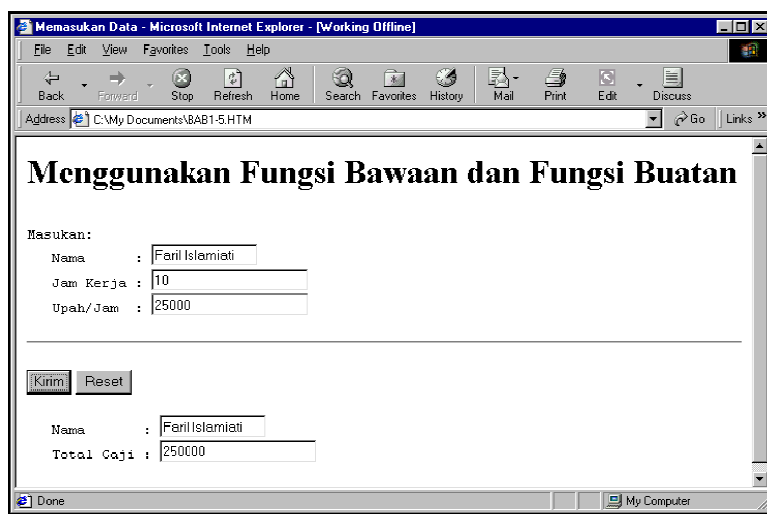
```
var X=parseInt("123");
```

```
if (isNaN(X))
```

```
X = -10;
```

Artinya jika X bukan bilangan maka X adalah -10

Berikut adalah contoh sebuah halaman yang telah menggunakan fungsi bawaan dan fungsi buatan.



Gambar 1.16. Menggunakan Fungsi

Program Menggunakan Fungsi Bawaan dan Fungsi Buatan

```
<html>
<head>
<title>Memasukan Data </title>
<script language="JavaScript">
function EvenTekan()
{
    var NamaAnda = (document.fForm>Nama.value);
    var JamKerja = parseInt (document.fForm.JamKer.value);
    var UpahPerJam = parseInt (document.fForm.Upah.value);
    var Total = JamKerja * UpahPerJam;
    document.fForm>NamaAnda.value>NamaAnda;
    document.fForm.Total.value=Total;
    alert("Total " + Total);
}
</script>
</head>
<body>
<form name="fForm">
<H1>
    Menggunakan Fungsi Bawaan dan Fungsi Buatan
</H1>
<PRE>
Masukan:
    Nama : <input type="text" size="13" name="Nama">
    Jam Kerja : <input type="text" size="20" name="JamKer">
    Upah/Jam : <input type="text" size="20" name="Upah">
    <hr>
</PRE>
    <input type="button" value="Kirim" onClick="EvenTekan()">
    <input type="reset" value="Reset">
<PRE>
    Nama : <input type="text" size="13" name="NamaAnda">
    Total Gaji : <input type="text" size="20" name="Total">
</PRE>
</form>
</body>
</html>
```

Latihan

Dengan menggunakan fungsi bawaan buatlah sebuah halaman untuk mencari akar persamaan kuadrat



Gambar 1.11. Membuat Program Persamaan Kuadrat

1.14. Kejadian

Even adalah sesuatu yang terjadi pada halaman HTML. Berikut ini ada beberapa bentuk contoh kejadian yaitu jika pengguna memuat dokumen, pengguna memasukan data pada kotak teks, pengguna mengklik tombol, dsb.

Semua kejadian pada JavaScript dapat Anda tangani dengan menentukan kejadiannya. Biasanya penanganan kejadian adalah sebuah fungsi, tetapi pada beberapa kasus, Anda bisa menuliskan pernyataan-pernyataan secara langsung.

Berikut ini adalah daftar kejadian pada JavaScript

Kejadian	Keterangan
Abort	Dibangkitkan bila pengguna menghentikan pemuatan citra (tag); yaitu bila pengguna menekan tombol stop atau mengklik link
Blur	Dibangkitkan bila sebuah elemen form kehilangan fokus masukan; yaitu bila pengguna menekan tombol <Tab> atau mengklik elemen form lainnya.
Change	Dibangkitkan bila informasi masukan pada sebuah elemen form (text, textarea, choice) diubah oleh pengguna
Click	Dibangkitkan bila pengguna mengklik sebuah elemen form atau link
Error	Dibangkitkan bila terjadi kesalahan saat browser memuat dokumen atau citra.
Focus	Dibangkitkan bila sebuah elemen form menerima fokus masukan; yaitu bila pengguna mengklik elemen form ini atau menekan tombol <tab> sehingga fokus masukan berpindah ke elemen ini
Load	Dibangkitkan bila browser selesai memuat dokumen
MouseOut	Dibangkitkan bila kursor mouse keluar dari daerah link atau peta cita
MouseOver	Dibangkitkan bila kursor mouse berada di atas sebuah link. Anda mungkin memperhatikan bila kursor mouse berada diatas sebuah link, browser akan menampilkan keterangan pada status.
Reset	Dibangkitkan bila pengguna menekan tombol reset
Select	Dibangkitkan bila pengguna memilih sebagian atau seluruh teks pada elemen form yang berupa kotak teks
Submit	Dibangkitkan bila pengguna menekan tombol Submit
Unload	Dibangkitkan bila pengguna keluar dari dokumen

Untuk menangani suatu kejadian, tambahkan atribut **onKejadian** pada tag yang sesuai. Kemudian isilah atribut ini dengan pernyataan-pernyataan JavaScript. Biasanya pernyataan yang diisikan berupa pemanggilan ke suatu fungsi.

Sintak penanganan kejadian adalah sebagai berikut:

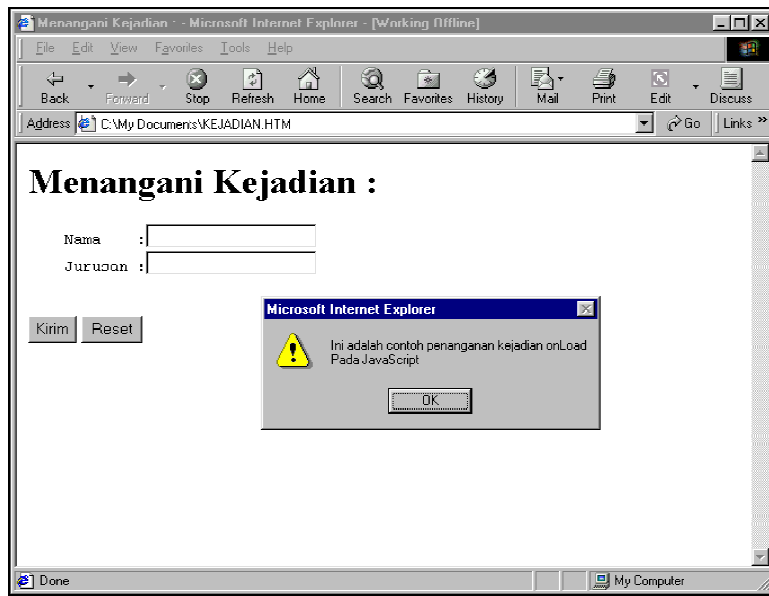
OnKejadian = "Pernyataan-pernyataan"

Contoh:

<BODY ONLOAD = "evLoad()">

<FORM NAME=:fForm" onSubmit="evSubmit()">

Pada saat pertama kali halaman dimuat, program akan menampilkan dialog seperti gambar berikut:



Gambar. 1.18. Kejadian onLoad

Kemudian masukan nama dan jurusan Anda, setelah itu baru di Click tombol Kirim , maka akan ditampilkan hasil seperti gambar berikut:



Gambar. 1.19. Kejadian onClick

Kotak dialog di atas muncul karena dalam program terdapat definisi penanganan tombol kejadian **onClick**:

```
<input type="button" value="Kirim" onClick ="evClick()">
```

Menangani Kejadian OnClick

```
<html>
<head>
<title>Menangani Kejadian :</title>
</head>
<body onload="evLoad()"
<p><script language="JavaScript">
function evLoad()
{
    alert("Ini adalah contoh penanganan kejadian onLoad\n" +
        "Pada JavaScript");
}
function evClick()
{
    alert("Halo " + document.fmForm.nama.value + "\nSelamat datang di Jurusan " +
document.fmForm.jurusan.value);
}

</script></p>

<FORM NAME="fmForm">
<h1>Menangani Kejadian :</h1>

<PRE>
    Nama :<input type="text" size="20" name="nama">
    Jurusan :<input type="text" size="20" name="jurusan"></p>
```

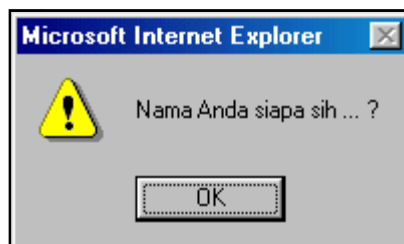
```
</PRE>
<p><input type="button" value="Kirim" onClick ="evClick()">
  <input type="reset" name="B2" value="Reset"></p>
</form>
</body>
</html>
```

Berikut akan dibuat sebuah halaman, apabila pertama kali dibuat akan ditampilkan kotak isian seperti gambar berikut:



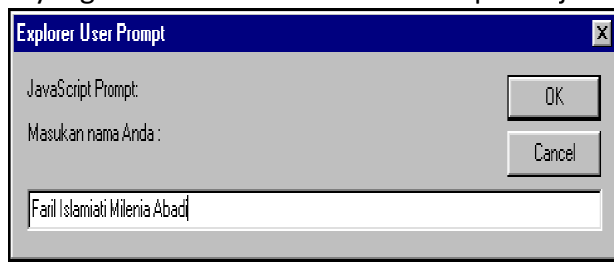
Gambar. 1.20. User Prompt

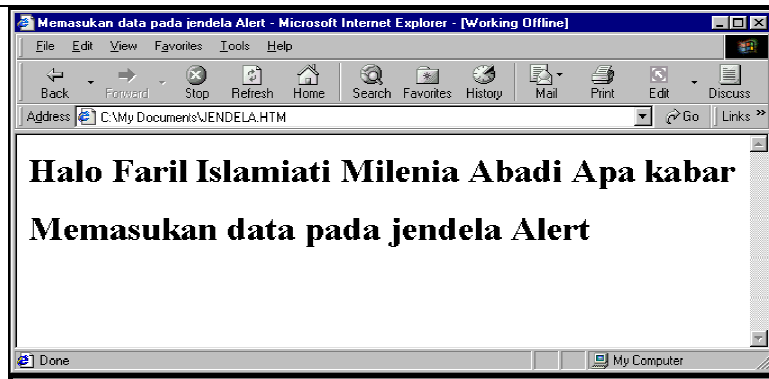
Kemudian jika sudah diisi anda tekan tombol Ok, jika nama Anda tidak diisi maka akan ditampilkan pesan seperti berikut:



Gambar. 1.21. Kotak Informasi

Tetapi jika ada nama yang dimasukan maka akan ditampilkan jendela seperti berikut:





Gambar. 1.22. Hasil keluaran

Penggunaan Alert

```
<html>
<head>
<title>Memasukan data pada jendela Alert</title>
<script language="JavaScript";
var nama=" ";
while ((nama==null) || (nama==" "))
    nama= prompt ("Masukan nama Anda : ", " ");
    if ((nama== null) || (nama==" "))
    {
        alert("Nama Anda siapa sih ... ? ");
    }
}
document.writeln("<h1>Halo " + nama + " Apa kabar </h1>");
</script></p>
</head>
<h1>Memasukan data pada jendela Alert</h1>

</FORM>
</body>
</html>
```

1.15. Mendefinsikan Objek

Kata kunci yang digunakan untuk mendefinisikan objek sama dengan kata kunci yang digunakan untuk mendefinisikan fungsi, yaitu function. Sebagai contoh, disini akan didefinisikan objek mahasiswa yang terdiri dari tiga properti, yaitu nama,alamat,jurusan. Untuk itu didefinisikan fungsi seperti contoh berikut:

```
function Mahasiswa (Nama,Alamat,Jurusan)
{
    this.Nama=Nama;
    this.Alatat=Alamat;
    this.Jurusan=Jurusan
}
```

properti **this** merupakan penunjuk objek ini. Anda memerlukan properti this karena pernyataan di atas hanyalah definisi objek, Anda belum menciptakan instan untuk objek tersebut. Untuk menciptakan instan dari objek mahasiswa di atas digunakan kata kunci **new**.

Contoh:

Sintia = new Mahasiswa("Sintia", "Jl. Panorama No. 80", "Sekretaris");

Untuk mengakses suatu objek, operator yang digunakan adalah operator titik (.).

Contoh

```
document.writeln(Sintia>Nama);
```

atau

```
var Nama=Sintia>Nama;
```

1.15.1. Pernyataan for..in

Pernyataan for..in digunakan untuk melakukan pengulangan berdasarkan properti-properti sebuah objek

Contoh:

```
for (Var x in Sintia)
    document.writeln(Sintia[x]);
```

Contoh halaman untuk memperjelas pendefinisian objek

Contoh Penggunaan for..in

```
<html>
<p><script language="JavaScript">
function Mahasiswa>Nama,Alamat,Jurusan)
{
    this>Nama>Nama;
    this.Alatat=Alamat;
    this.Jurusan=Jurusan;
}
function Tulis(objek)
{
    for (var x in objek)
        document.writeln(objek[x]);
        document.writeln(" ");
}

</script></p>
</body>
<p><script language="JavaScript">document.writeln("<h1>Membuat Objek </h1> ");
document.writeln("<PRE>");

//menciptakan instan objek mahasiswa
var Sintia = new Mahasiswa ("Sintia Ratna Dewi", "Jl. Panorama III No. 80", "Sekretaris");
var Sisca = new Mahasiswa ("Sisca Nawangwulan", "Jl. Ciwaruga No. 72", "Akuntansi");
var Sarah = new Mahasiswa ("Sarah Galabagan", "Jl. Bagdad No. 76", "Informatika");

Tulis(Sintia);
```

```
Tulis(Sisca);
Tulis(Sarah);
document.writeln("</PRE>");
</script></p>
</body>
</html>
```

1.15.2. Mendefinisikan Metode

Objek merupakan pengkapsulan properti/variabel bersama-sama dengan metode /fungsi. Untuk mendefinisikan metode, pertama-tama Anda harus mendefinisikan fungsi biasa.

Contoh:

```
function Anu()
{
}
```

Kemudian pada definisi objek, tambahkan sebuah metode yang menunjuk fungsi Anu.

Contoh

```
this.metode=Anu;
```

Sekarang anda sudah memiliki sebuah metode yang bernama metode.

Berikut ini adalah sebuah halaman yang telah menggunakan Metode untuk menuliskan objek.



Gambar 1.23. Pemakaian Metode

Pemakaian Metode dalam Java Script

```
<html>
<head>
<p><script language="JavaScript">
function Mahasiswa(Nama,Alamat,Jurusan)
{
    this.Nama=Nama;
    this.Alamat=Alamat;
    this.Jurusan=Jurusan;
    //deklarasi metode
    this.Tulis=TulisObjek;
    this.Ubah=UbahObjek;
}

function TulisObjek()
{
    document.writeln("Nama : " + this.Nama);
```

```
document.writeln("Alamat : " + this.Alat);
document.writeln("Jurusan : " + this.Jurusan);
document.writeln(" ");
}
```

```
function UbahObjek(Nama,Alamat,Jurusan)
{
    this.Nama      = Nama;
    this.Alat      = Alamat;
    this.Jurusan   = Jurusan;
}
```

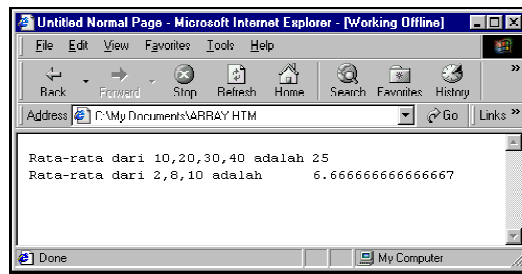
```
document.writeln("<h1>Membuat Objek </h1> ");
document.writeln("<PRE>");

//menciptakan instan objek mahasiswa
Mhs = new Mahasiswa ("Sintia Ratna Dewi", "Jl.Panorama III No. 80", "Sekretaris");
Mhs.Tulis();

//mengubah properti objek
Mhs.Ubah("Sisca Si Jenat", "Jl.Ciwaruga 74", "Perbankan");
Mhs.Tulis();
document.writeln("</PRE>");
</script></p>
</body>
</html>
```

1.16. Array

Array pada JavaScript merupakan sekumpulan elemen, dimana masing-masing elemen dapat bertipe apa saja. Jadi konsep array dapat dikatakan sebagai penggabungan beberapa variabel menjadi satu kesatuan. Contoh JavaScript untuk array:



Gambar 1.24. Menampilkan Array

Program Menampilkan Array

```
<html>
<p><script language="JavaScript">
function RataRata()
{
```

```

        var Jumlah=0.0;
        var Total = RataRata.arguments.length;
        for (var i=0;i<Total;i++)
        Jumlah += RataRata.arguments[i];
        return Jumlah/Total;
    }

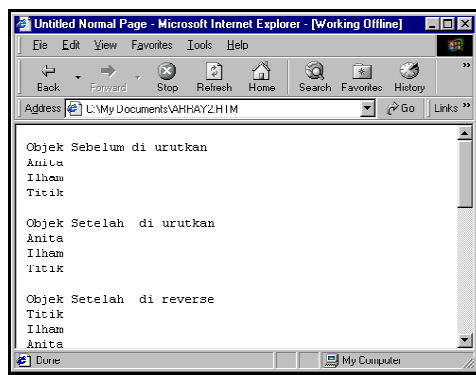
    document.writeln("<PRE>");
    document.writeln("Rata-rata dari 10,20,30,40 adalah " + RataRata(10,20,30,40));
    document.writeln("Rata-rata dari 2,8,10 adalah    " + RataRata(2,8,10));
    document.writeln("</PRE>");
</script></p>
</body>
</html>

```

Objek array mempunyai beberapa properti; salah satu properti yang penting adalah length. Properti length menyatakan jumlah elemen yang dimiliki oleh Array.

Objek Array mempunyai beberapa metode untuk memanipulasi array, yaitu

- join (mengkonversikan semua array ke string),
- reverse (membalik elemen-elemen array).
- sort untuk mengurutkan elemen-elemen array
- split, mengkonversikan string ke dalam array
- concat, menggabungkan dua buah array



Gambar 1.25. Penggunaan Objek Array

Penggunaan Objek Array

```

<html>
<p><script language="JavaScript">
function Tulis(A,str)
{
    document.writeln(str)
    for (var i=0;i<A.length;i++)
    document.writeln(A[i]);
    document.writeln(" ");
}
</script></p>

```

```
<p><script language="JavaScript"><!--
document.write ("<PRE>");
var Mahasiswa = new Array("Anita","Ilham","Titik");
Tulis (Mahasiswa,"Objek Sebelum di urutkan ");
Mahasiswa.sort();
Tulis (Mahasiswa,"Objek Setelah di urutkan ");
Mahasiswa.reverse();
Tulis (Mahasiswa,"Objek Setelah di reverse ");
Tulis (Mahasiswa,"Objek Setelah join" + Mahasiswa.join('*'));
document.write("<PRE>");
</script></p>
<p><script language="JavaScript">
document.write("<PRE>");
var MahasiswaBaru = ["Kemala","Adinda"];
Tulis(MahasiswaBaru,"Objek Mahasiswa Baru: ");
var MahasiswaGabungan = Mahasiswa.concat(MahasiswaBaru);
Tulis(MahasiswaGabungan,"Objek setelah digabung");
MahasiswaGabungan = MahasiswaGabungan.slice(1,3);
Tulis(MahasiswaGabungan,"Setelah object di slice (1,3): ");
document.write("<PRE>");
></script></p>
</body>
</html>
```

1.17. Frame dalam Java Script

Untuk membuat menu dengan menggunakan frame dapat dilakukan dengan menggunakan kombinasi perintah html dan java script, dimana framenya di buat di html sedangkan actionnya dibuat dengan menggunakan java script.

Contoh, misalnya Anda akan membuat sebuah halaman seperti berikut:



Gambar 1.26. Halaman Utama Frame

Jika Anda Click tombol **Yahoo**, kemudian Anda tekan tombol **Tampilkan**, maka pada jendela sebelah kanan akan ditampilkan halaman seperti berikut:

Gambar 1.21. Menampilkan website yahoo.com

Jika Anda Click pilihan **Google**, maka akan ditampilkan halaman seperti berikut pada jendela sebelah kanan:

Gambar 1.21. Menampilkan Website google.com

Jika Anda Click pilihan **Detik**, maka akan ditampilkan halaman detik.com, seperti berikut pada jendela sebelah kanan:

Gambar 1.28. Menmapilkan Website Detik.com

Program utama.html

```
<html>
<frameset cols = "20%,*">
<frame src = "menu.html" name = "fresatu">
<frame src = "awal.html" name = "fredua">
</frameset>
</html>
```

Program menu.html

```
<html>
<head>
<title></title>
</head>
<p><script language="javascript">

function evkananatas()
{
    var str = "http://yahoo.com";
    if (document.fmform.rbkananatas[1].checked)
        str = "http://google.com";
    else if (document.fmform.rbkananatas[2].checked)
        str = "http://www.detik.com";
    parent.fredua.location.href=str;
}

</script></p>
<p>pilih salah satu dibawah ini:</p>
<form name = "fmform">
    <p><input type="radio" checked name="rbkananatas" value= "1" >Yahoo</p>
    <p><input type="radio" name="rbkananatas" value="2">Google</p>
    <p><input type="radio" name="rbkananatas" value="3">Detik</p>
    <p><input type="button" value="tampilkan" onclick="evkananatas()"> </p>
</form>
</body>
</html>
```

Program awal.html

```
<html>
<head>
<title></title>
</head>
```

```
<body background="gray.jpg">
<p align="center">&nbsp;</p>
<h1 align="center">Halaman ini dibuat dengan menggunakan</h1>
<h1 align="center"><font size="7">Frame </font></h1>
<p align="center"><strong>Design By Fahra Ragita</strong></p>
</body>
</html>
```

Daftar Pustaka:

JavaScript, Antony Pranata

Java Script Tutorial, w3c

Dan dari semua yang tidak dapat disebutkan satu persatu

Sekian. Selamat Mencoba, sampai jumpa lagi pada materi dan latihan lainnya.

Nb.nanti diteruskan lagi yah.....

Taryana Suryana. M.Kom

Teknik Informatika

Universitas Komputer Indonesia

taryanarx@yahoo.com

Materi ini dapat di download di <http://nilaionline.unikom.ac.id>