



PENGANTAR JAVASCRIPT

- Latar Belakang
- Kelebihan dan Kekurangan Javascript

Latar Belakang

Seperti halnya dengan bahasa pemrograman lainnya, yakni Pascal, C++, Basic dan sebagainya, java merupakan bahasa pemrograman yang umurnya relatif masih muda bila dibandingkan dengan lainnya.

Java merupakan bahasa pemrograman yang menjadi kelanjutan dari bahasa C++. Java dikembangkan oleh Sun Microsystem, Inc. pada tahun 1995 dan telah mengalami revolusi (perkembangan) yang sangat pesat sampai sekarang ini.

Bahasa pemrograman ini memiliki beberapa keunggulan dibandingkan dengan bahasa pemrograman lainnya yaitu ukuran filenya kecil, akses data cepat, bersifat kompiler-interpretor dan dapat dibaca oleh berbagai system operasi seperti windows ataupun linux.

Java memiliki sifat *platform independent* (terbuka) yang menjadi ciri khas bahasa pemrograman tersebut. Hal ini menyebabkan bahasa pemrograman ini dapat di baca seluruh komputer tanpa harus mengkonversinya sehingga ketika user sedang berkomunikasi dengan menggunakan fasilitas komputer, user tidak perlu mengkhawatirkan perbedaan software maupun hardware dari masing-masing komputer.

Javascript merupakan bahasa pemrograman yang dibuat untuk memudahkan bagi setiap orang untuk mempelajari bahasa Java. Berbeda dengan java, javascript tidak memiliki kompiler seperti halnya java. Sifat yang sama dengan bahasa java adalah javascript bersifat interpretor, yaitu script yang telah dibuat di text editor seperti notepad atau wordpad akan dibaca perbaris dari baris awal sampai akhir.

Bahasa pemrograman javascript dibuat secara khusus untuk membuat suatu halaman web yang interaktif maupun dinamis. Jika java merupakan bahasa pemrograman dan HTML (Hyper Text Markup Language) merupakan bahasa kode maka javascript hibrid yang menjadi perpaduan antara bahasa java dengan HTML.

Beberapa keunggulan javascript dalam mengolah dan mengkondisikan sebuah web antara lain, karena ukuran filenya kecil maka ketika seorang merequest sebuah web dari webserver dimana didalam web tersebut memuat script javascript maka aplikasi dari javascript tersebut akan cepat tampil di browser tanpa harus diolah terlebih dahulu di server. Hal ini disebabkan karena javascript memiliki sifat client side yaitu script yang ketika dipanggil oleh browser akan langsung tampil tanpa harus diolah di server disertai script aslinya sehingga user dapat melihat sourcenya dengan mudah tanpa terenkripsi oleh server.

Sedangkan ada beberapa bahasa pemrograman web yang ketika dipanggil browser maka script program tersebut akan diolah dan bekerja di server, sedangkan ketika web tersebut akan ditampilkan di browser maka scriptnya tidak akan disertai di browser karena scriptnya hanya akan bekerja di server dan tidak bekerja di client. Bahasa pemrograman ini sifatnya server side. Contohnya antara lain adalah PHP, ASP, JSP dan masih banyak lainnya. Beberapa jenis software yang menjadi webserver di antaranya adalah apache yang sekarang ini sangat digandrungi oleh banyak sebagian besar masterweb untuk dijadikan webservernya.

Untuk permasalahan server side ini, kita akan membahasnya pada bab selanjutnya yaitu mengenai MySQL dan PHP. Pada bab ini kita hanya membahas peran javascript sebagai aplikasi tambahan di browser yang bersifat client side.

Beberapa keuntungan dan keunggulan javascript sebagai bahasa pemrograman di web sedikit telah kita uraikan di paragraph sebelumnya. Javascript juga mampu membuat aplikasi yang sangat menarik yang dapat kita letakkan di web kita. Seperti program kalkulator sederhana yang dipakai di web www.bhinneka.com, pembuatan link-link dinamis yang hanya mendekatkan kursor ke link tersebut maka akan keluar-menu-menu lainnya. Jenis ini bisa dilihat di webnya www.microsoft.com. Ada juga yang mampu membuat animasi gambar dan text yang cukup banyak digunakan di web-web yang terkenal seperti

www.detik.com. Bahkan ada juga yang bisa membuat game dari javascript dan juga membuat homepage online. Jika kita sering browsing di internet maka kita akan mudah menemukan aplikasi-aplikasi javascript yang terdapat di web-web yang interaktif dan dinamis.

Dalam pembuatan javascript, text editor yang memungkinkan untuk digunakan adalah notepad, wordpad dan sebagainya. Ada beberapa Browser yang mendukung untuk menjalankan javascript dan juga yang tidak mendukung javascript, untuk browser yang mendukung javascript ada beberapa jenis seperti internet explorer dan netscape.

Pada awalnya perkembangan javascript dengan pesat disebabkan semakin menjamurnya web-web yang ada di internet. Perkembangan World Wide Web yang sangat pesat menyebabkan munculnya banyak homepage dengan halaman-halaman yang sangat menarik. Halaman Web yang dulunya hanya berupa teks sekarang menjadi cerdas dan interaktif.

Oleh karena itu muncullah CGI (Common Gateway Interface). Dengan CGI, halaman web bisa menampilkan pencacah pengunjung, animasi sederhana dan masih banyak lagi sayangnya CGI meletakkan beban pemrosesan di server, bukan di client. Akibatnya dengan semakin banyaknya pengguna yang mengakses server lalu lintas jaringan semakin padat. Di samping CGI ada bahasa pemrograman lain yang juga digunakan untuk meningkatkan interaktivitas halaman web yaitu java, yang pemrosesannya dilakukan di client namun sayangnya java sangat sulit dipelajari terutama oleh pemula. Karena itu dibuatlah suatu bahasa pemrograman web yang khusus dibuat untuk memberikan tampilan yang interaktif dan dinamis di sebuah web sehingga memberi kesan bagi pengunjung agar pengunjung tersebut sering membuka web tersebut yang mengakibatkan informasi yang kita letakkan di dalam web tersebut dapat dibaca oleh banyak pemakai jasa internet di dunia.

Kelebihan dan Kekurangan Javascript

Kelebihan Javascript

1. Ukuran file kecil

Script dari javascript memiliki ukuran yang kecil sehingga ketika web yang memiliki javascript ditampilkan di browser maka akses tampilannya akan lebih cepat dibandingkan ketika browser membuka suatu web yang memiliki script java. Hal ini juga sangat berkepentingan dengan daya kerja server. Semakin kecil space suatu web yang disimpan dalam suatu server maka daya kerja server ketika di browsing oleh user di internet akan tidak terlalu berat, selain itu sifat javascript client side yang tidak perlu lagi di olah oleh server ketika browser memanggil web dari sebuah server.

2. Mudah untuk dipelajari

Javascript merupakan bahasa semi pemograman yang merupakan gabungan antara bahasa pemograman java dengan bahasa kode HTML sehingga disebut bahasa hybrid. Walaupun javascript merupakan turunan dari java namun javascript tidak memiliki aturan yang serumit java.

3. Terbuka

Javascript tidak terikat oleh hardware maupun software tertentu bahkan system operasi seperti windows maupun unix. Karena ia bersifat terbuka, maka ia dapat dibuat maupun di baca di semua jenis komputer.

Kekurangan Javascript

1. Script tidak terenkripsi

Karena javascript bersifat client side, maka script yang kita buat di text editor dan telah dijadikan web di server, ketika user me-request web dari server tersebut maka sintak javascript akan langsung ditampilkan di browser. User bisa melihat dan menirunya dari sourcenya.

2. Kemampuan terbatas

Walaupun javascript mampu membuat bentuk web menjadi interaktif dan dinamis, namun javascript tidak mampu membuat program aplikasi sendiri seperti java.

3. Keterbatasan Objek

Javascript tidak mampu membuat kelas-kelas yang bisa menampung objek-objek tambahan seperti java karena javascript telah memiliki objek yang built-in pada struktur bahasanya.

STRUKTUR JAVASCRIPT

- Struktur dan Sintak Dasar Javascript
- Posisi Script Javascript di HTML

Struktur dan Sintak Dasar Javascript

Ditinjau dari segi pembuatan dan penulisannya, javascript tidak jauh beda dengan HTML seperti pada pertemuan awal kita, sehingga file yang menggunakan sintak javascript di dalam script HTML dapat menggunakan ekstension HTML juga.

Script dasar dalam javascript diawali dengan sintak `<script>` dan *diakhiri* dengan sintak `</script>`. Agar text editor dapat mengidentifikasi bahwa sintak tersebut merupakan sintak javascript yang digunakan untuk aplikasi di web, maka perlu penambahan sintak di dalam sintak script yaitu :

```
<script language="javascript">
```

Sedangkan ruang yang ada di antara sintak javascript tersebut kita dapat mencantumkan ekspresi, fungsi, pernyataan dan lain sebagainya sesuai keinginan kita.

Di dalam penulisan javascript untuk menampilkan suatu pernyataan atau kalimat kita dapat menggunakan sintak :

```
Document.write("kalimat yang akan di tampilkan");
```

Dan pada setiap akhir pernyataan umumnya disertai dengan tanda titik koma (;) kecuali pada kondisi tertentu saja yang akan kita bahas pada bab-bab berikutnya.

Terkadang ada beberapa browser yang lama tidak dapat mengidentifikasi script javascript secara baik sehingga tampilan di browser tersebut akan menjadi kurang indah untuk dilihat. Oleh karena itu kita harus mengantisipasinya dengan menggunakan tanda "`<!--`" setelah penulisan `<script language="javascript">` dan tanda "`//>`" sebelum penulisan `</script>`.

Di dalam script javascript juga dikenal tanda komentar yang apabila kita bubuhkan tanda tersebut maka kalimat yang ada di samping ataupun di antara tanda tersebut berfungsi sebagai komentar saja dan tidak ditampilkan di browser. Tanda ini sangat membantu bagi pemula atau yang baru mempelajari javascript agar lebih memahami fungsi maupun peran sintak yang di terangkan oleh tanda keterangan tersebut. Tanda tersebut ada beberapa jenis antara lain:

1. `“//”`, Tanda tersebut apabila kita letakkan di text editor yang menjadi wahana pembuatan script javascript maka kalimat apapun yang ada di sampingnya akan menjadi keterangan dan tidak dapat ditampilkan di browser. Tanda tersebut biasanya digunakan untuk keterangan yang hanya satu baris saja.
2. `“/* */”`, tanda tersebut merupakan tanda komentar yang apabila ada kalimat di antaranya maka kalimat tersebut akan menjadi keterangan biasa dan tidak ditampilkan oleh browser. Tanda ini biasanya digunakan pada keterangan yang sbanyaknya beberapa baris.

Dari segi peletakan posisi sintak javascript, kita dapat membaginya menjadi 2 kondisi, sebagai berikut :

1. Sintak javascript ditulis dengan file yang berbeda.
2. Sintak javascript terintegrasi dengan sintak HTML.

Di bawah ini kita akan menguraikan secara rinci perbedaan antara kedua kondisi

di atas yaitu :

1. Sintak javascript ditulis dengan file yang berbeda

Tata cara penulisan seperti ini yaitu memisahkan sintak javascript dengan sintak HTML dapat memudahkan bagi pembuat web apabila kita ingin membuat web yang berisi berbagai aplikasi javascript di setiap halaman yang kita buat. Penulisan secara manual sintak javascript di setiap lembar web akan cukup merepotkan bagi kita terlebih lagi hal tersebut dapat memperberat kerja server karena besarnya file yang harus di kerjakan di server akibat banyaknya script javascript yang berulang-ulang di setiap file web tersebut. Hal ini dapat di atasi dengan membuat suatu file khusus bagi

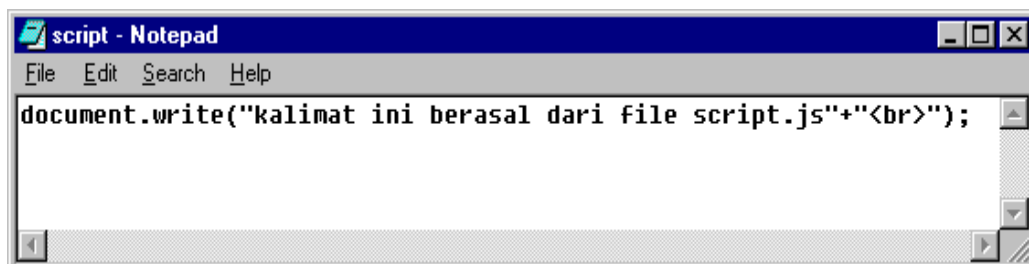
script javascript yang sama dari beberapa file yang menggunakan script tersebut sehingga file utama akan mencari file tersebut dengan sintak tertentu.

Script javascript yang kita buat tersebut kemudina kita simpan dalam suatu file dengan ekstension “.js”. Dalam file tersebut kita tidak perlu mencatumkan sintak pembuka `<script>` dan sintak penutup `</script>`. File utama terserbut akan mencari file yang berekstention “.js” tersebut dengan sintak :

```
<script src="nama_file.js"> </script>
```

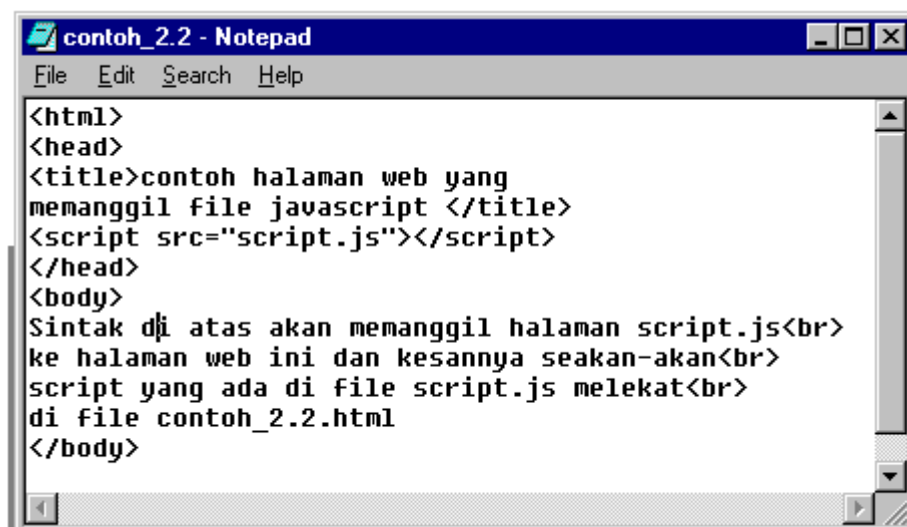
Dengan menggunakan sintak terserbut, kita dapat memanggil kapanpun dan diposisi manapun kita ingin menampilkan aplikasi javascript. Metode ini sangat memudahkan dan cukup praktis bagi pembuat web.

Contoh file Script.js



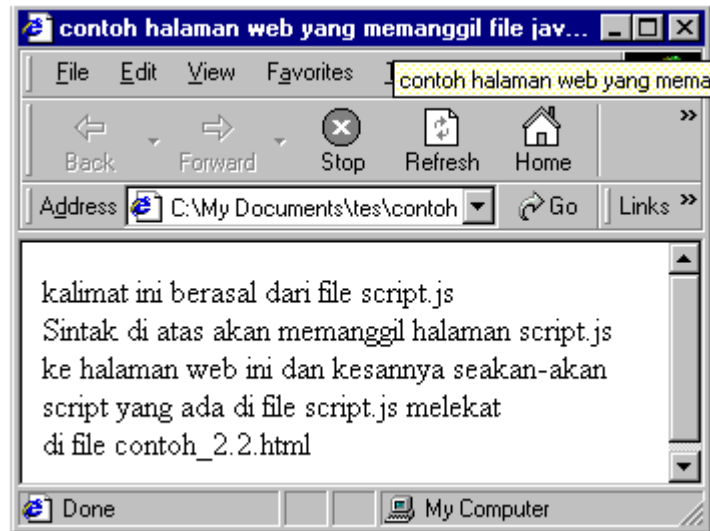
```
document.write("kalimat ini berasal dari file script.js"<br>);
```

Contoh_2.2.html



```
<html>
<head>
<title>contoh halaman web yang
memanggil file javascript </title>
<script src="script.js"></script>
</head>
<body>
Sintak di atas akan memanggil halaman script.js<br>
ke halaman web ini dan kesannya seakan-akan<br>
script yang ada di file script.js melekat<br>
di file contoh_2.2.html
</body>
```

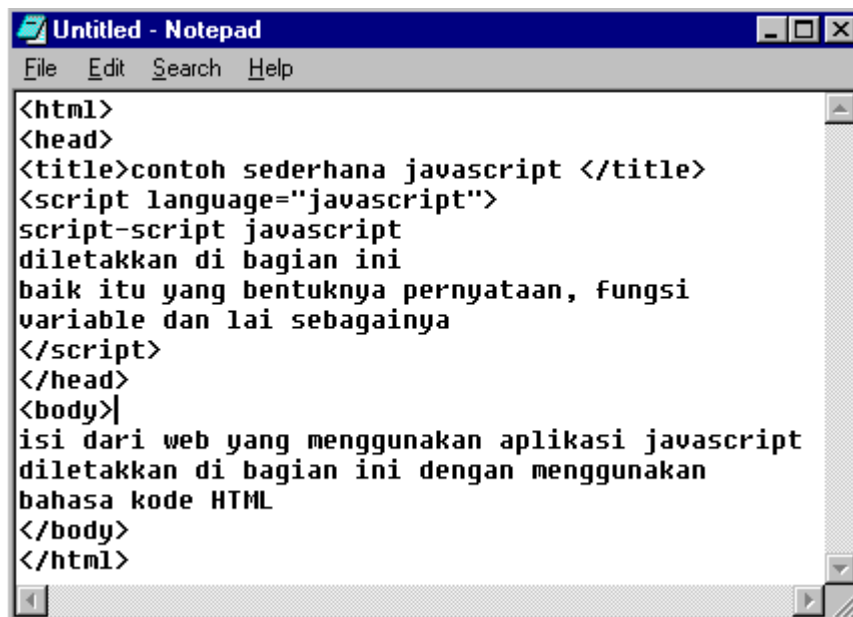
Bila kita buka browser dengan nama file contoh_2.2.html, maka yang tampak di browser adalah :



2. Sintak javascript terintegrasi dengan sintak HTML

Metode ini sudah cukup banyak digunakan oleh para pembuat web di dunia dan cakupan script yang digunakan tidaklah terlalu banyak sehingga tidak terlalu memberi efek berarti bagi daya kerja server untuk mengirim web yang beraplikasi javascript ke browser. Sintak-sintak javascript yang menggunakan metode ini biasanya tidak terlalu sering dibuat oleh sipembuat web karena bila ada script javascript yang ingin digunakan secara berulang-ulang maka akan lebih efisien kalau dibuat dengan cara seperti cara nomor di atas. Bila kita letakkan sintak javascript bersamaan dengan sintak HTML biasa dan kita buat secara berulang-ulang di setiap lembar web yang kita buat maka hal tersebut akan menjadikan ukuran web kita akan menjadi lebih besar dibandingkan bila kita menggunakan cara pertama seperti di atas.

Contoh :



```
<html>
<head>
<title>contoh sederhana javascript </title>
<script language="javascript">
script-script javascript
diletakkan di bagian ini
baik itu yang bentuknya pernyataan, fungsi
variable dan lai sebagainya
</script>
</head>
<body>
isi dari web yang menggunakan aplikasi javascript
diletakkan di bagian ini dengan menggunakan
bahasa kode HTML
</body>
</html>
```

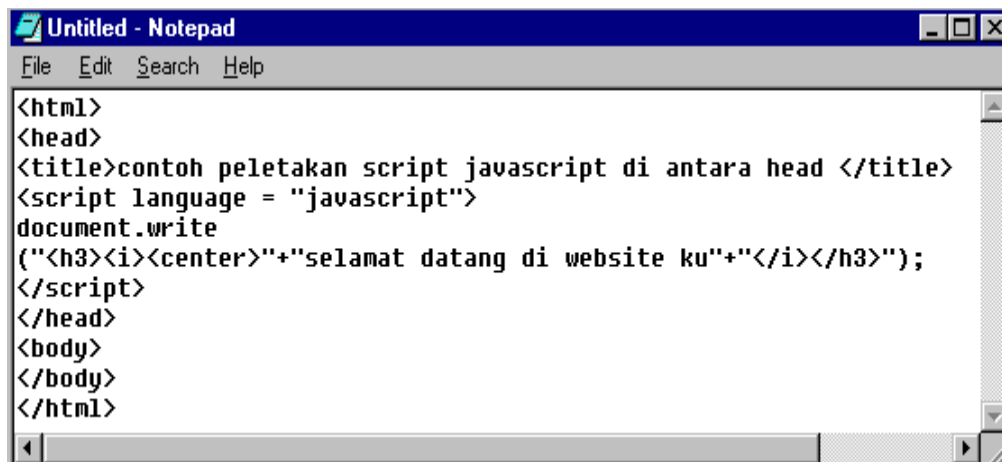
Posisi Script Javascript di File HTML

Dari segi penulisan javascript, kita dapat menulis sintak (script) javascript di beberapa posisi pada bagian tertentu di sintak HTML yang telah ada. Di bawah ini kita akan membahas berbagai posisi yang dapat diletakkan sintak javascript di bagian file HTML seperti pada bagian HEAD maupun BODY.

1. Script javascript diletakkan di antara <HEAD> </HEAD>

Posisi ini sering sekali kita jumpai hampir disetiap web. Posisi ini cukup membantu kecepatan tampilan web yang menggunakan aplikasi javascript. Karena sifatnya interpreter, maka ketika browser pertama sekali membaca sebuah web yang dipanggil olehnya, apabila script javascript tersebut diletakkan di antara sintak <HEAD>....</HEAD> maka browser akan pertama sekali membaca dan mengidentifikasi sintak tersebut sebelum browser membaca isi dari <BODY>. Hal ini sangat membantu dari segi kecepatan tampilan sebuah web di browser.

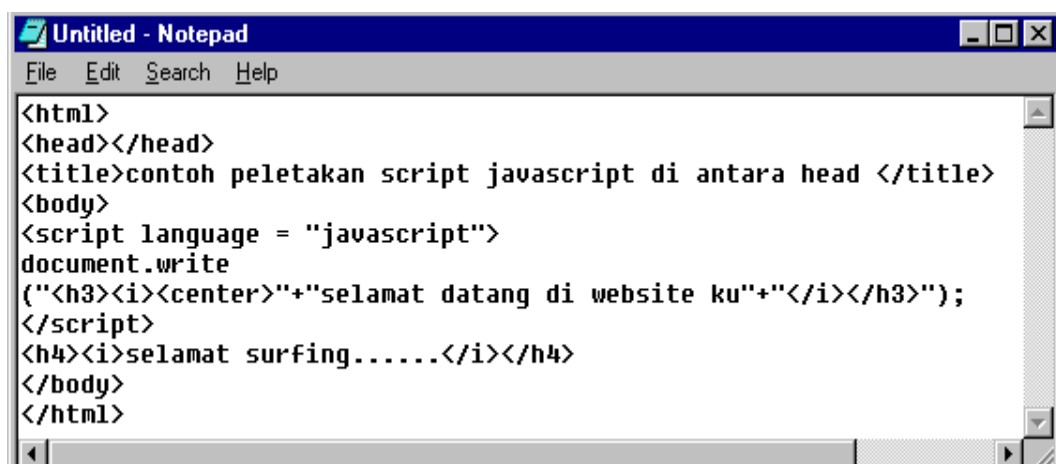
Contoh_2.4.html



```
<html>
<head>
<title>contoh peletakan script javascript di antara head </title>
<script language = "javascript">
document.write
("<h3><i><center>""selamat datang di website ku""</i></h3>");
</script>
</head>
<body>
</body>
</html>
```

2. Script javascript diletakkan di antara <BODY> </BODY>

Penempatan script javascript di antara sintak BODY merupakan salah satu metode posisi pengetikan script javascript selain seperti yang di atas. Jika kita meletakkan script javascript di dalam sintak BODY maka ketika browser menerima sebuah web dari salah satu web server maka sintak javascript dan sintak HTML lainnya akan bersamaan diproses sehingga kecepatannya akan sedikit berkurang bila dibandingkan sintak javascript diletakkan di antara HEAD. Hal ini disebabkan javascript bersifat interpreter. Penempatan posisi ini juga dapat mengatur posisi aplikasi javascript pada tubuh website yang kita buat dengan mengaturnya melalui table agar preposisi javascript dapat dengan mudah diletakkan. Contoh_2.5.html



```
<html>
<head></head>
<title>contoh peletakan script javascript di antara head </title>
<body>
<script language = "javascript">
document.write
("<h3><i><center>""selamat datang di website ku""</i></h3>");
</script>
<h4><i>selamat surfing.....</i></h4>
</body>
</html>
```

NILAI, VARIABLE DAN KONSTANTA

- Nilai Dalam Javascript
- Variable
- Konstanta/Literal

Nilai Dalam Javascript

Nilai merupakan besaran yang digunakan untuk melakukan perhitungan, menulis dan sebagainya. Di dalam javascript terdapat beberapa macam nilai, yaitu

1. String

String adalah karakter yang bisa berupa huruf , kata symbol atau angka. String ditulis diantara tanda kutip ganda (“) atau tanda kutip tunggal (‘). Jika terdapat string lain gunakan tanda kutip tunggal. misal :

```
<INPUT NAME = age on focus = "windows.status= ' ketikkan umur anda ' ">  
name = "firdaus 'alikomeini' adnan "
```

sedang untuk menulis tanda kutip ganda dalam string digunakan tanda (\ “).
Misalnya :

```
name = "Institut \"Teknologi \" Bandung ”
```

2. Numerik

Berupa bilangan atau angka yang mempunyai sifat matematis (bisa dijumlah, tambah dsb), misalnya : 17, 8, 1945

3. Boolean

Nilai yang hanya memiliki dua nilai yaitu *true* (benar) dan *false* (salah)

4. Null

Adalah nilai yang tidak memiliki nilai sama sekali . Null tidak sama dengan karakter kosong atau nilai nol (0).

Variabel

Aturan penamaan variable

Java Script memiliki aturan untuk memberi nama variabel :

1. Harus diawali dengan karakter (huruf atau garis bawah) tidak boleh diawali dengan angka atau symbol.

Contoh :

Variable	Keterangan
Contoh 3.1	Benar
<u>contoh</u> 3.2	Benar
2001_angkatan	Salah
\$sql	Salah

2. Tidak boleh menggunakan spasi , untuk memisahkan antar karakter digunakan garis bawah.

Contoh :

Variable	Keterangan
Firdaus adnan	Salah
Firdaus_adnan	benar

3. Kapitalisasi (huruf besar dan huruf kecil) dibedakan pada nama variabel.

Contoh :

Firdaus tidak sama dengan *firdaus*

4. Tidak boleh menggunakan reserved word , atau nama yang sama dengan perintah yang ada pada Java Script. Reserved word dalam javascript adalah :

Abstract, Boolean, break, byte, case, catch, char, class, const, continue, default, do, double, else, extends, false, final, finally, float, for, function, goto, if, implements, import, in, instace of, int, interface, long, native, new, null, package, private, protected, public, return, short, static, super, switch, synchronized, this, throw, throws, transient, true, try, var, void, while, with.

Memberikan nilai pada variabel

Untuk memberikan nilai pada variabel digunakan sintaks :

var *namavariabel* = nilai (value)

contoh :

1. Memberikan nilai string untuk variable :

```
var nama = "firdaus adnan"
```

2. Memberi nilai numerik untuk variable :

```
var angka = 1000.12
```

3. Memberi nilai Boolean untuk variable :

```
var x = true
```

```
var y = false
```

4. Memberi nilai null untuk variable :

```
var nama = null
```

Konstanta/Literal

Konstanta/literal adalah suatu nilai pasti yang tidak bisa diubah-ubah. Pengertian ini relevan dengan kata dasarnya yaitu konstan yang artinya tetap. Jika isi variable masih bisa diubah, maka konstanta tidak.

Antara variable dan konstanta dapat disatukan. Perhatikan contoh di bawah ini :

```
total = subtotal + 100
```

Nilai 100 di atas adalah konstanta/literal sedangkan total dan subtotal merupakan variable.

Beberapa jenis literal pada Java Script :

1. Literal Integer,

yaitu suatu bilangan bulat tanpa pecahan

Literal	Bilangan
2001	Desimal
702	Octal
A33ff	Heksadesimal

2. Literal Floating Point,

yaitu suatu bilangan pecahan atau berpangkat.

Literal	Arti
3.14	3,14
2E3	2×10^3

-2E3	-2 x 10 ³
2E-3	2 x 10 ³

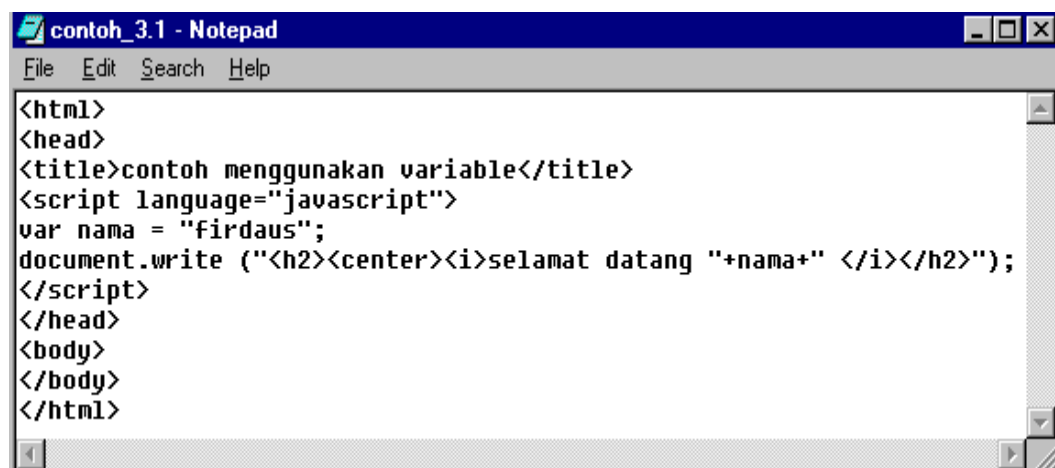
3. Literal String,

adalah suatu karakter yang berisi huruf , angka atau symbol-simbol lainnya. Literal string harus diawali dan diakhiri dengan tanda kutip ganda (“) atau tunggal (‘).

Literal	Keterangan
“firdaus adnan”	Benar
“firdaus adnan’	Salah

4. Literal Boolean,

adalah suatu literal yang memiliki dua buah nilai yaitu *true* (benar) dan *false* (salah)
contoh_3.1.html



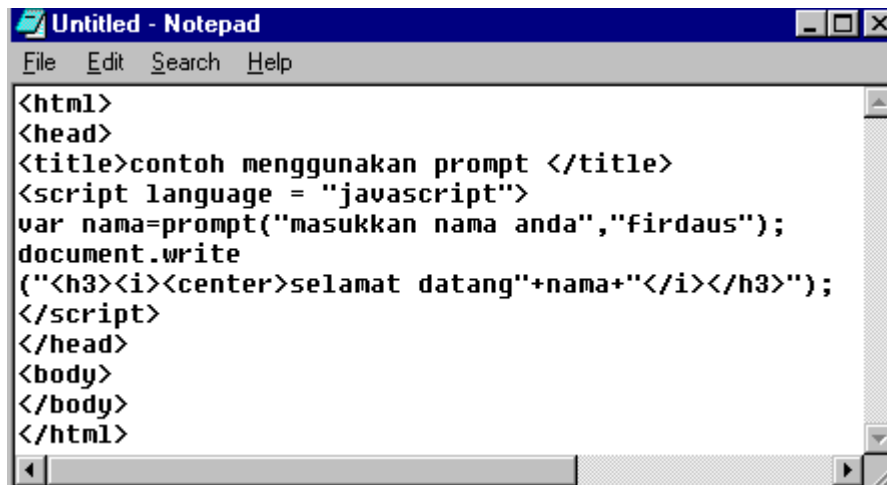
```

<html>
<head>
<title>contoh menggunakan variable</title>
<script language="javascript">
var nama = "firdaus";
document.write ("<h2><center><i>selamat datang "+nama+" </i></h2>");
</script>
</head>
<body>
</body>
</html>

```

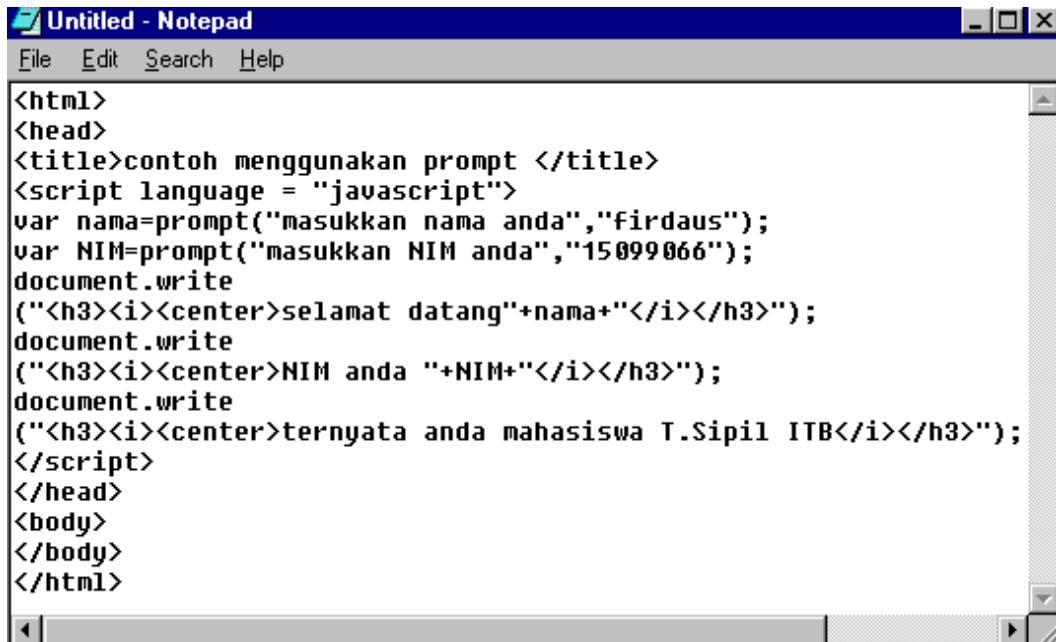
contoh di atas menjelaskan bahwa variable yang diidentifikasi dengan nama menyatakan sebuah string yang value-nya adalah firdaus. Ketika kita ingin memanggil value tersebut kedalam browser maka kita harus mencantumkan nama variable yang telah kita definisikan di atas yaitu *nama* kedalam sintak pernyataan atau sintak yang digunakan untuk menampilkan suatu kalimat yaitu *document.write* dan menambahkan tanda “+” dan tanda kutip dua (“) di samping variable tersebut.

Contoh_3.2.html



```
<html>
<head>
<title>contoh menggunakan prompt </title>
<script language = "javascript">
var nama=prompt("masukkan nama anda","firdaus");
document.write
("<h3><i><center>selamat datang"+nama+"</i></h3>");
</script>
</head>
<body>
</body>
</html>
```

Sintak prompt di atas merupakan script dalam javascript untuk memerintahkan user untuk memasukkan input ke dalam box yang telah tersedia, di mana input tersebut menjadi variable terhadap output yang dideklarasikan di dalam sintak pernyataan yaitu di dalam *document.write*. Beda halnya dengan alert yang akan kita bahas pada bab selanjutnya. Contoh_3.3.html



```
<html>
<head>
<title>contoh menggunakan prompt </title>
<script language = "javascript">
var nama=prompt("masukkan nama anda","firdaus");
var NIM=prompt("masukkan NIM anda","15099066");
document.write
("<h3><i><center>selamat datang"+nama+"</i></h3>");
document.write
("<h3><i><center>NIM anda "+NIM+"</i></h3>");
document.write
("<h3><i><center>ternyata anda mahasiswa T.Sipil ITB</i></h3>");
</script>
</head>
<body>
</body>
</html>
```


OPERATOR

- Pengantar Operator
- Operator Unary
- Operator Binary
- Operator Penugasan
- Perbandingan

Pengantar Operator

Operator merupakan perintah yang memanipulasi nilai atau variable dan memberikan suatu hasil tertentu. Beberapa jenis operator yang sudah umum yaitu operator aritmatika yang sering kita gunakan dalam membahas matematika, diantaranya adalah operator tambah (+), kurang (-), kali (x), bagi (/), persen (%) dan sebagainya.

Data-data yang dimanipulasi oleh operator untuk memberikan suatu hasil yang diinginkan disebut juga dengan *operan*.

Sebagai contoh :

$$X + 3 = 6$$

Keterangan :

- X dan 3 adalah operan
- + adalah operator
- 6 adalah hasil

Dalam modul ini kita akan mempelajari jenis-jenis operator javascript, di antaranya adalah :

1. Operator Unary
2. Operator Binary
3. Operator penugasan
4. Operator penugasan

5. Operator perbandingan
6. Operator logical
7. Operator String

Operator Unary

Operator unary adalah operator yang hanya mengolah satu nilai atau satu operan saja. Operator unary dalam javascript terbagi atas tiga macam, yaitu :

1. Increment (penambahan), dengan symbol ++
2. Decrement (pengurangan), dengan symbol –
3. Negasi (lawan), dengan symbol –

Untuk lebih jelasnya, kita akan membahas satu-persatu beserta contohnya.

4.2.1 Increment (penambahan)

Operator ini berfungsi untuk menambahkan nilai operan (variable) sebenar satu.

Contoh :

a. `total = subtotal++`

Dari contoh di atas kita dapat mendefinisikan bahwa variable *subtotal* dimasukkan ke variable *total* terlebih dahulu, kemudian nilai variable *subtotal* ditambah dengan angka satu.

b. `total =++ subtotal`

Dari contoh di atas kita dapat mendefinisikan bahwa variable *subtotal* ditambah terlebih dahulu, lalu nilai variable *subtotal* di masukkan ke variable *total*.

4.2.2 Decrement (pengurangan)

Operator ini akan mengurangi nilai operan (variable) sebesar satu.

Contoh :

a. `total = subtotal--`

Artinya variable *subtotal* dimasukkan ke variable *total* terlebih dahulu kemudian nilai variable *subtotal* dikurangi Satu

b. `total = --subtotal`

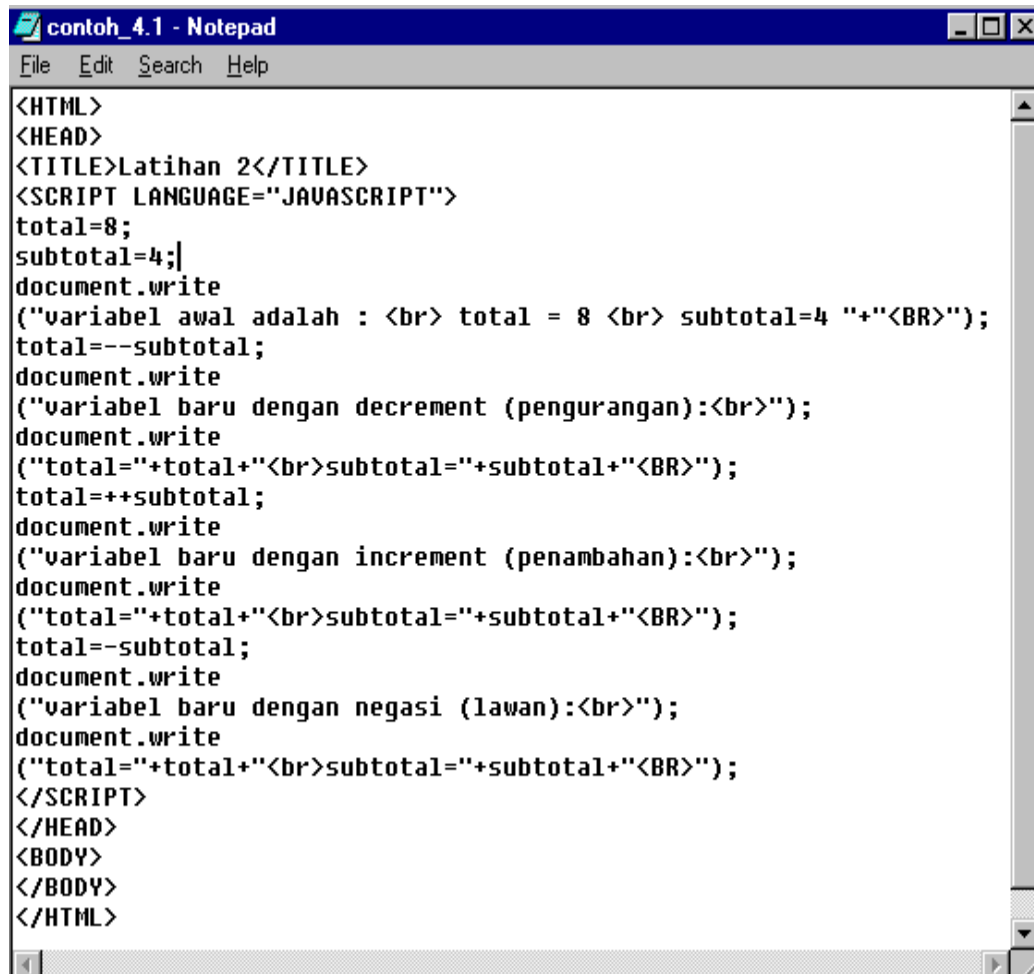
Artinya nilai variable *subtotal* dikurangi satu terlebih dahulu, kemudian nilai variable *subtotal* dimasukkan ke variable *total*.

4.2.3 Negasi (lawan)

Operator ini akan membalikkan nilai operan. Jika operan bernilai positif, operator negasi akan menjadikannya negatif dan sebaliknya.

Perhatikan contoh soal berikut :

Contoh_4.1.html



```
<HTML>
<HEAD>
<TITLE>Latihan 2</TITLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
total=8;
subtotal=4;
document.write
("variabel awal adalah : <br> total = 8 <br> subtotal=4 "+"<BR>");
total--subtotal;
document.write
("variabel baru dengan decrement (pengurangan):<br>");
document.write
("total="+total+"<br>subtotal="+subtotal+"<BR>");
total++subtotal;
document.write
("variabel baru dengan increment (penambahan):<br>");
document.write
("total="+total+"<br>subtotal="+subtotal+"<BR>");
total--subtotal;
document.write
("variabel baru dengan negasi (lawan):<br>");
document.write
("total="+total+"<br>subtotal="+subtotal+"<BR>");
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Operator Binary

Operator Binary adalah operator yang mengolah dua nilai atau dua operan.

Perhatikan contoh di bawah ini :

$$x + y = 5$$

Tanda + adalah operator binary, karena tanda tersebut mengolah dua buah operan yakni x dan y. Semua operator binary pada javascript adalah operator aritmatika. Perhatikan table di bawah ini :

Operator	Fungsi	Contoh
+	Penjumlahan	var x = 2 + 3 ; nilai x = 5
-	Pengurangan	var x = 3 - 2; nilai x=1
*	Perkalian	var x = 2*3; nilai x = 6
/	Perkalian	var x = 3/2; nilai x = 1.5
%	Modulus	var x = 3%2; nilai x = 1

Perhatikan contoh di bawah ini

Contoh_4.2.html

```

<html>
<head>
<title> contoh fungsi operator Binary </title>
<script language="JAVASCRIPT">
document.write("<h2><i>"+<i>"Contoh penggunaan operator binary"</i></h2>");
document.write ("hasil dari 2+3 adalah :"+(2+3)+"<BR>");
document.write ("hasil dari 3-2 adalah :"+(3-2)+"<BR>");
document.write ("hasil dari 3*2 adalah :"+(3*2)+"<BR>");
document.write ("hasil dari 3/2 adalah :"+(3/2)+"<BR>");
document.write ("hasil dari 3 modulus 2 adalah :"+(3%2)+"<BR>");
</script>
</head>
<body>
</body>
</html>

```

Operator Penugasan

Operator penugasan adalah operator yang memberi nilai kepada variable. Jenis operator yang umum digunakan adalah tanda sama dengan (=). Contoh :

```
var nilai = 10
```

```
nilai = matematika
```

Pada contoh di atas, variable *nilai* diberi nilai (value) 10. Pada baris kedua, variable *nilai* diberi nilai (value) yang terdapat pada variable matematika.

Jenis operator lainnya merupakan singkatan dari operator binary.

Perhatikan contoh di bawah ini :

nilai += matematika

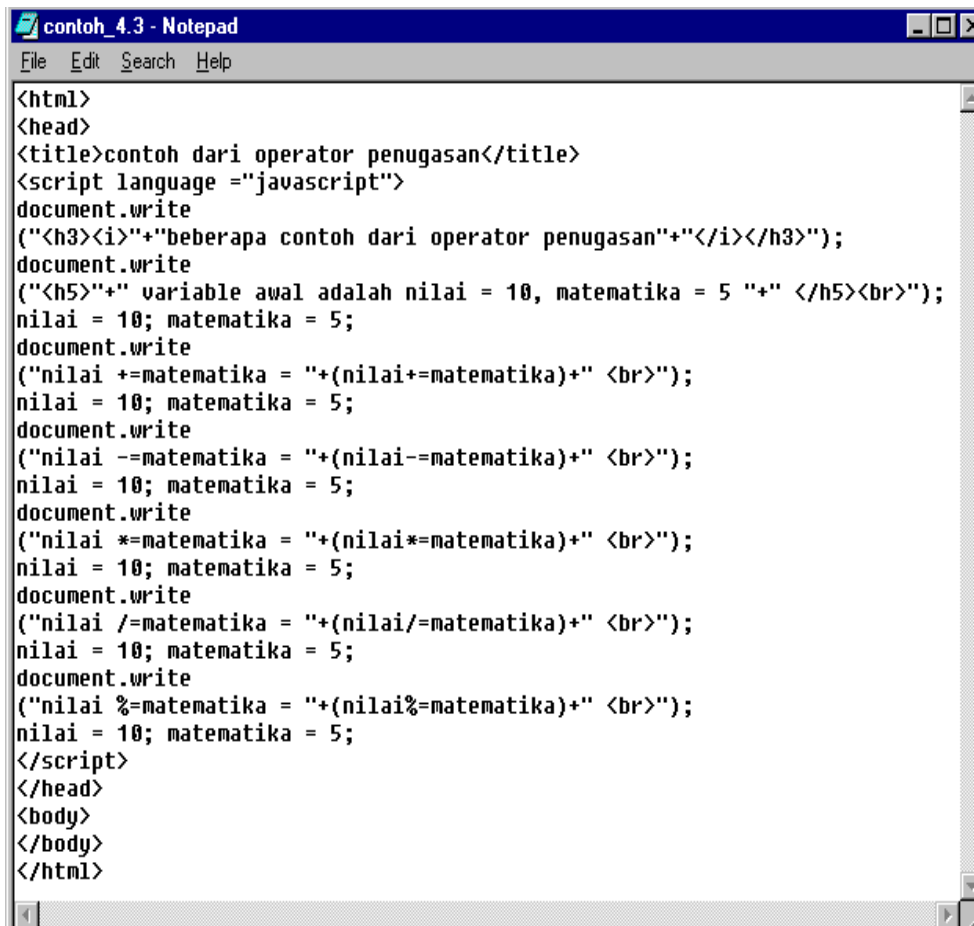
contoh di atas sama artinya dengan :

nilai = nilai + matematika

Beberapa jenis operator penugasan yang terdapat pada javascript adalah :

Operator	Contoh	Arti
=	nilai = 10	value nilai = 10
+=	nilai += matematika	nilai = nilai + matematika
-=	nilai -= matematika	nilai = nilai - matematika
*=	nilai *= matematika	nilai = nilai * matematika
/=	nilai /= matematika	nilai = nilai / matematika
%=	nilai %=matematika	nilai = nilai % matematika

Perhatikan contoh_4.3.html berikut ini :



```
<html>
<head>
<title>contoh dari operator penugasan</title>
<script language = "javascript">
document.write
("<h3><i>"+"beberapa contoh dari operator penugasan"+"</i></h3>");
document.write
("<h5>"+" variable awal adalah nilai = 10, matematika = 5 "+" </h5><br>");
nilai = 10; matematika = 5;
document.write
("nilai +=matematika = "+(nilai+=matematika)+" <br>");
nilai = 10; matematika = 5;
document.write
("nilai -=matematika = "+(nilai-=matematika)+" <br>");
nilai = 10; matematika = 5;
document.write
("nilai *=matematika = "+(nilai*=matematika)+" <br>");
nilai = 10; matematika = 5;
document.write
("nilai /=matematika = "+(nilai/=matematika)+" <br>");
nilai = 10; matematika = 5;
document.write
("nilai %=matematika = "+(nilai%=matematika)+" <br>");
nilai = 10; matematika = 5;
</script>
</head>
<body>
</body>
</html>
```

Perbandingan

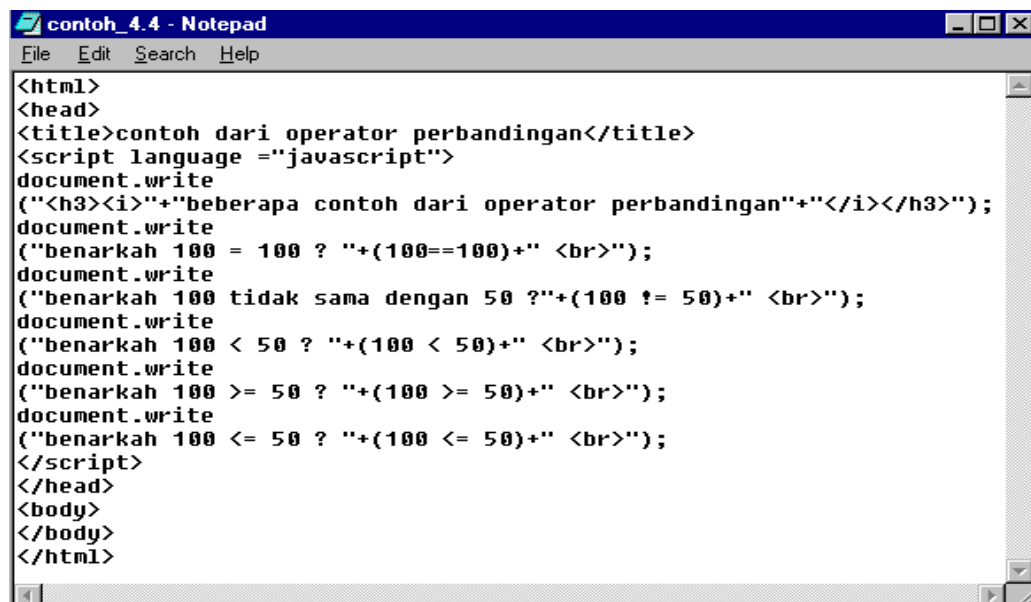
Operator perbandingan adalah operator yang digunakan untuk membandingkan satu nilai dengan nilai yang lain, untuk mendapatkan hasil benar atau salah. Pada umumnya, macam dari jenis operator perbandingan ini adalah sama dengan (==). Contohnya :

Biaya == 100

Jika nilai dari variable biaya di atas besarnya sama dengan 100, maka pernyataan di atas akan memberikan hasil true (benar). Tetapi jika nilai dari variable biaya di atas besarnya tidak sama dengan 100, maka pernyataan di atas akan memberikan hasil false (salah).

Operator	Arti	Contoh
==	Sama dengan	Total==10
!=	Tidak sama dengan	Total !=100
>	Lebih besar dari	Total>100
<	Kurang dari	Total<100
>=	Lebih besar atau sama	Total>=100
<=	Kurang dari atau sama	Total<=100

Perhatikan contoh_4.4.html di bawah ini :



```
contoh_4.4 - Notepad
File Edit Search Help
<html>
<head>
<title>contoh dari operator perbandingan</title>
<script language = "javascript">
document.write
("<h3><i>"+"beberapa contoh dari operator perbandingan"+"</i></h3>");
document.write
("benarkah 100 = 100 ? "+(100==100)+" <br>");
document.write
("benarkah 100 tidak sama dengan 50 ? "+(100 != 50)+" <br>");
document.write
("benarkah 100 < 50 ? "+(100 < 50)+" <br>");
document.write
("benarkah 100 >= 50 ? "+(100 >= 50)+" <br>");
document.write
("benarkah 100 <= 50 ? "+(100 <= 50)+" <br>");
</script>
</head>
<body>
</body>
</html>
```

OPERATOR (lanjutan) DAN EKSPRESI

- Operator Logikal
 - Operator string
- Ekspresi
- Definisi Ekspresi
 - Ekspresi Perbandingan
- Alert

Operator Logikal

Pada dasarnya Operator *logical* hampir sama dengan operator perbandingan yang berfungsi untuk membandingkan dua buah nilai untuk memberikan hasil benar atau salah. Perbedaannya adalah operator logical berfungsi *membandingkan perbandingan*. Perhatikan contoh di bawah ini :

$(total > 10) \ \&\& \ (biaya < 100)$

Pada contoh di atas digunakan operator $\&\&$ yang artinya adalah mengharuskan kedua nilai memenuhi syarat. Jika kedua nilai benar maka ekspresi di atas akan bernilai benar.

Beberapa jenis operator yang terdapat di javascript adalah sebagai berikut :

Operator	Arti	keterangan
$\&\&$	AND (dan)	Kedua nilai yang dibandingkan harus sesuai dengan kriteria
$\ \ $	OR (atau)	Salah satu nilai yang dibandingkan harus sesuai dengan kriteria
$!$	NOT (bukan)	Nilai yang dibandingkan merupakan kebalikannya.

Dari contoh di atas yaitu $(total > 10) \ \&\& \ (biaya < 100)$ dapat kita perkirakan bahwa :

- Jika total = 12 (benar) dan biaya = 98 (benar), maka hasilnya adalah *benar*
- Jika total = 8 (salah) dan biaya = 98 (benar), maka hasilnya adalah *salah*
- Jika total = 8 (salah) dan biaya = 102 (salah), maka hasilnya adalah *salah*

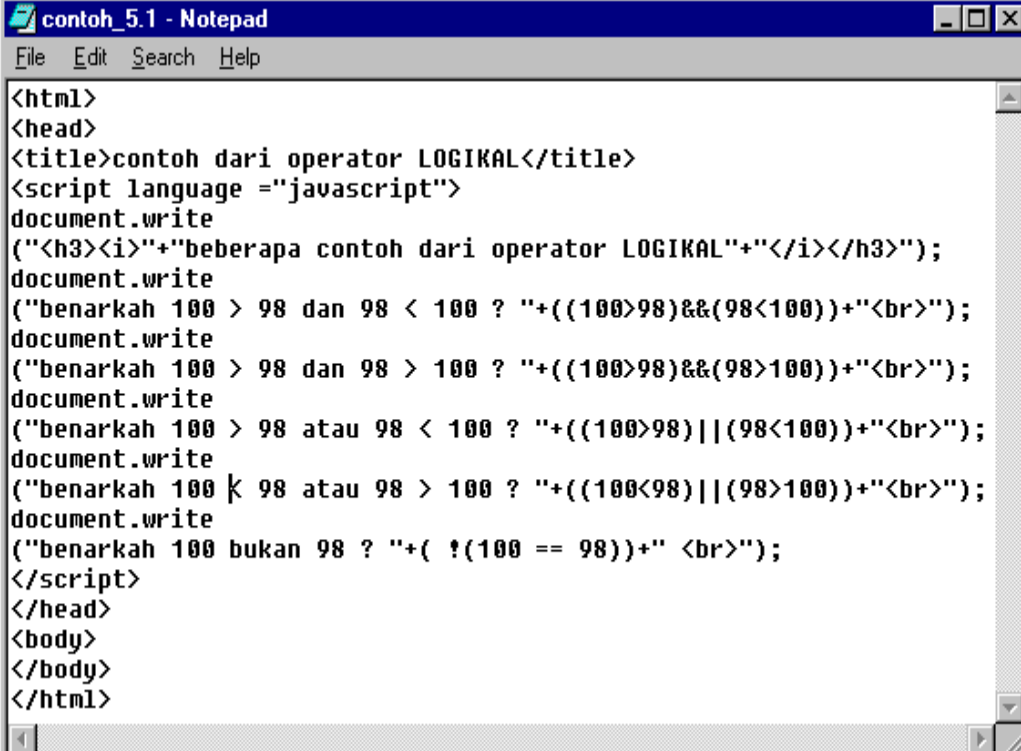
Jika operator contoh di atas kita ganti dengan tanda (||) maka dapat kita ambil perkiraan yaitu :

- Jika total = 12 (benar) dan biaya = 98 (benar), maka hasilnya adalah *benar*
- Jika total = 8 (salah) dan biaya = 98 (benar), maka hasilnya adalah *benar*
- Jika total = 12 (benar) dan biaya = 102 (salah), maka hasilnya adalah *benar*
- Jika total = 8 (salah) dan biaya = 102 (salah), maka hasilnya adalah *salah*

Jika contoh di atas dirubah menjadi `!(total == 100)`, maka dapat kita perkirakan hasilnya sebagai berikut :

- Jika total=100 (benar), maka ekspresi di atas salah
- Jika total=98 (salah), maka ekspresi di atas benar

Untuk lebih jelasnya, perhatikan contoh_5.1.html di bawah ini :



```
<html>
<head>
<title>contoh dari operator LOGIKAL</title>
<script language ="javascript">
document.write
("<h3><i>"+"beberapa contoh dari operator LOGIKAL"+"</i></h3>");
document.write
("benarkah 100 > 98 dan 98 < 100 ? "+((100>98)&&(98<100))+"<br>");
document.write
("benarkah 100 > 98 dan 98 > 100 ? "+((100>98)&&(98>100))+"<br>");
document.write
("benarkah 100 > 98 atau 98 < 100 ? "+((100>98)|| (98<100))+"<br>");
document.write
("benarkah 100 < 98 atau 98 > 100 ? "+((100<98)|| (98>100))+"<br>");
document.write
("benarkah 100 bukan 98 ? "+( !(100 == 98))+" <br>");
</script>
</head>
<body>
</body>
</html>
```

Operator String

Operator string adalah operator yang mengolah nilai string, yaitu nilai yang berupa karakter (bukan angka atau ekspresi aritmatik).

Jenis operator string pada dasarnya sama dengan operator-operator yang telah dijelaskan di atas. Hanya saja, operator-operator tersebut digunakan untuk memanipulasi string. Perhatikan contoh di bawah ini :

```
var nama = "firdaus" + "alikomeini"+"adnan"
```

Dari contoh di atas maka dapat diambil kesimpulan bahwa variable pesan adalah string "firdaus alikomeini adnan".

Beberapa hal yang perlu diperhatikan adalah :

1.String tidak bisa dibandingkan

Jika kita tidak bisa membandingkan antara "roti" == "keju" atau "roti">"keju". Maka hasil keduanya adalah salah atau false.

2. Javascript mengubah string angka menjadi numerik

Apabila string berisi angka, javascript akan mengubahnya menjadi nilai numerik, sehingga dapat dibandingkan. Perhatikan contoh di bawah ini :

```
"100" > "10"
```

Javascript akan mengubah "100" menjadi 100 dan mengubah "10" menjadi 10. Karena 100 lebih besar dari 10, maka ekspresi di atas bernilai true atau benar.

3.String gabungan tidak diubah

Gabungan yang dimaksud adalah gabungan antara numerik dengan kalimat. Misalkan string tersebut berupa "angkatan2001", contoh tersebut tidak akan diubah menjadi nilai numerik. Perhatikan table berikut yang menjadi contoh penerapan operator string pada javascript :

Operator	Arti	Contoh
==	Membandingkan string	"roti" == "keju"
>	Lebih besar dari	"100" > "10"
+	Penjumlahan	nama_awal = "firdaus" nama_akhir = "adnan" Nama = nama_awal + nama_akhir

+=	Penjumlahan ganda	nama = "firdaus" nama+= "adnan"
----	-------------------	------------------------------------

Untuk lebih lengkapnya, perhatikan contoh_5.2.html di bawah ini

```

<html>
<head>
<title>contoh dari operator string</title>
<script language = "javascript">
document.write
("<h3><i>"+ "beberapa contoh dari operator string" + "</i></h3>");
document.write
("benarkah 'roti' = 'keju'? "+ ("roti"=="keju") + "<br>");
document.write
("benarkah '100' = '10'? "+ ("100">"10") + "<br>");
nama_awal = "firdaus";
nama_akhir = "adnan";
nama = nama_awal + nama_akhir;
document.write ("nama_awal = 'firdaus' " + "<br>");
document.write ("nama_akhir = 'adnan' " + "<br>");
document.write ("nama = nama_awal + nama_akhir = " + nama + "<br>");
nama_awal = "firdaus";
document.write ("nama_awal = 'firdaus' " + "<br>");
document.write ("nama_awal += 'adnan' = " + (nama_awal += "adnan"));
</script>
</head>
<body>
</body>
</html>

```

EKSPRESI

Definisi Ekspresi

Ekspresi merupakan kode-kode pemrograman yang digunakan untuk menampilkan perhitungan atau penugasan.

Pada penambahan mengenai operator di atas, anda telah menggunakan ekspresi ini.

Perhatikan contoh berikut :

var nilai = 100

```
var total = nilai1 + nilai2
```

Ekspresi perbandingan

Pada javascript, tidak ada ketentuan tertentu untuk penulisan ekspresi. Tetapi ada satu pengecualian yaitu pada ekspresi perbandingan.

Sintak untuk ekspresi perbandingan ini adalah sebagai berikut :

(kondisi) ? nilaibenar : nilaisalah

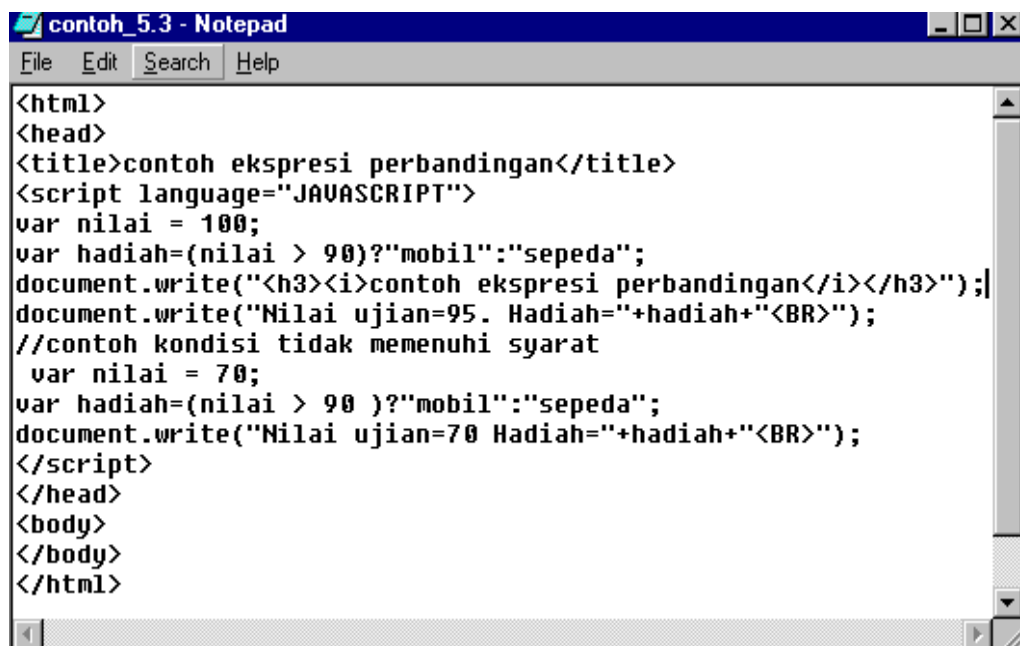
Dari sintak di atas, jika kondisi benar maka akan diberi nilai yang terdapat pada variable *nilaibenar*. Sedangkan jika kondisi salah maka akan memberikan nilai yang terdapat pada variable *nilaisalah*.

Perhatikan contoh di bawah ini :

Jika firdaus ingin mendapatkan mobil maka syaratnya ia harus mendapatkan nilai 90, tetapi jika ternyata nilai yang ia dapat di bawah 90 maka ayahnya akan memberinya sebuah sepeda. Kondisi seperti ini dapat kita terjemahkan ke dalam kalimat :

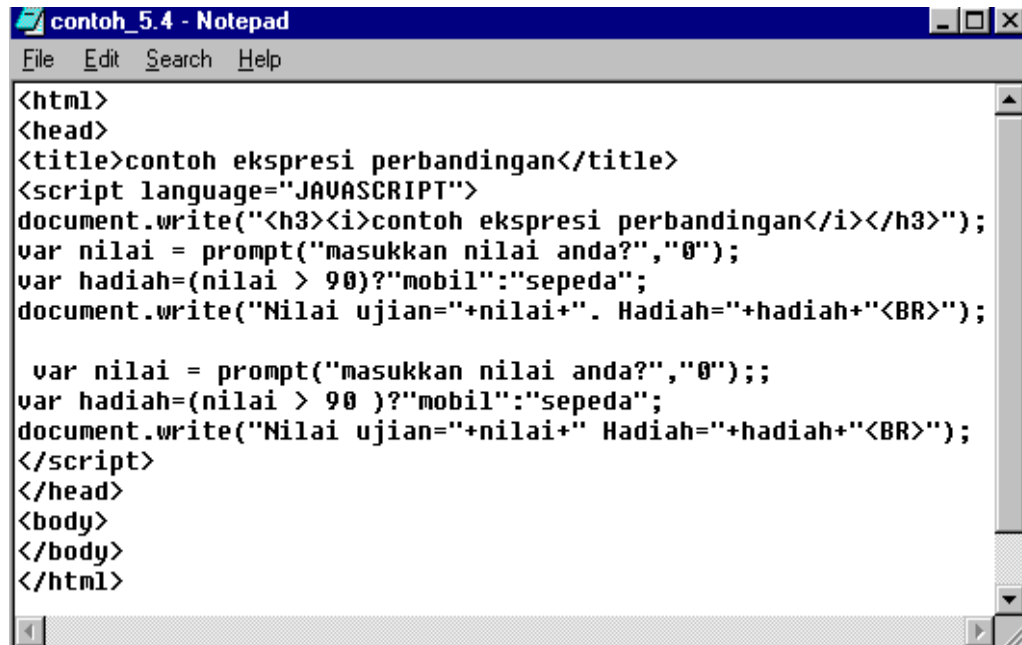
Hadiah = (nilai > 90) ? "mobil" : "sepeda"

Jika kita terjemahkan ke dalam bahasa javascript (contoh_5.3.html) adalah sebagai berikut :



```
contoh_5.3 - Notepad
File Edit Search Help
<html>
<head>
<title>contoh ekspresi perbandingan</title>
<script language="JAVASCRIPT">
var nilai = 100;
var hadiah=(nilai > 90)?"mobil":"sepeda";
document.write("<h3><i>contoh ekspresi perbandingan</i></h3>");
document.write("Nilai ujian=95. Hadiah="+hadiah+"<BR>");
//contoh kondisi tidak memenuhi syarat
var nilai = 70;
var hadiah=(nilai > 90 )?"mobil":"sepeda";
document.write("Nilai ujian=70 Hadiah="+hadiah+"<BR>");
</script>
</head>
<body>
</body>
</html>
```

Perhatikan contoh_5.4.html di bawah ini :

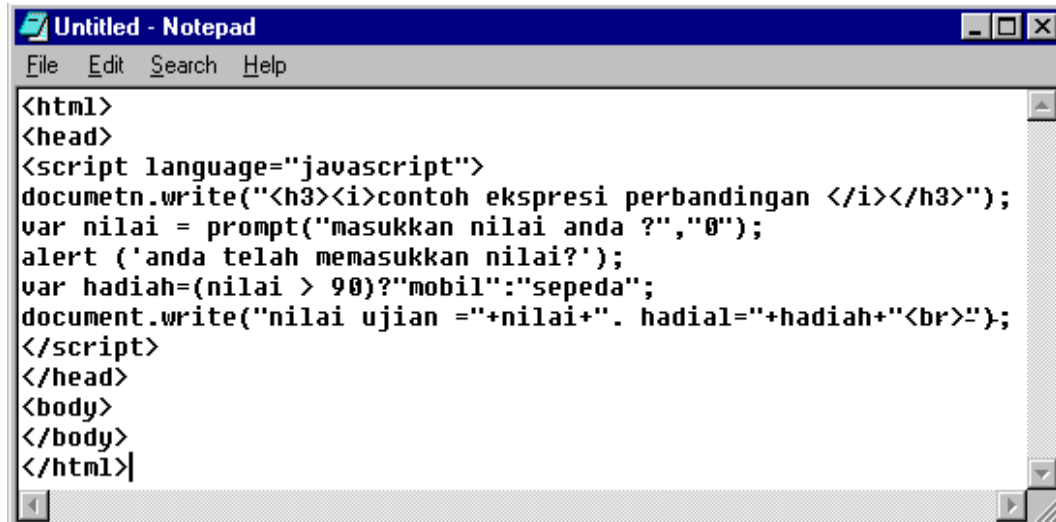


```
<html>
<head>
<title>contoh ekspresi perbandingan</title>
<script language="JAVASCRIPT">
document.write("<h3><i>contoh ekspresi perbandingan</i></h3>");
var nilai = prompt("masukkan nilai anda?","0");
var hadiah=(nilai > 90)?"mobil":"sepeda";
document.write("Nilai ujian="+nilai+" . Hadiah="+hadiah+"<BR>");

  var nilai = prompt("masukkan nilai anda?","0");;
var hadiah=(nilai > 90 )?"mobil":"sepeda";
document.write("Nilai ujian="+nilai+" Hadiah="+hadiah+"<BR>");
</script>
</head>
<body>
</body>
</html>
```

Alert

Alert dalam javascript merupakan script pemberi pesan kepada user ketika user memasukkan input ke dalam kolom data, ketika user sedang menekan tombol yang ada di browser atau peringatan kepada user ketika user salah memasukkan input ke dalam kolom data di mana input tersebut perlawana dengan keinginan script yang ada. Perhatikan contoh_5.5.html :



```
<html>
<head>
<script language="javascript">
document.write("<h3><i>contoh ekspresi perbandingan </i></h3>");
var nilai = prompt("masukkan nilai anda ?", "0");
alert ('anda telah memasukkan nilai?');
var hadiah=(nilai > 90)?"mobil":"sepeda";
document.write("nilai ujian =" +nilai+ ". hadiah=" +hadiah+ "<br>");
</script>
</head>
<body>
</body>
</html>
```

PERNYATAAN

- Pengantar Pernyataan
- Pernyataan if...else
- Pernyataan while
- Pernyataan for
- Pernyataan break

Pengantar Pernyataan

Pernyataan merupakan kata kunci yang digunakan untuk mengendalikan alur program javascript yang kita buat. Pernyataan bisa dibuat sebagai kalimat lengkap yang menyatukan ekspresi, variable, dan nilai-nilai yang telah kita pelajari pada modul-modul sebelumnya.

Pernyataan (statement) dapat terbagi menjadi 9 bagian yang akan kita bahas satu persatu. Kesembilan bagian tersebut adalah :

1. Pernyataan if..else
2. Pernyataan while
3. Pernyataan for
4. Pernyataan break
5. Pernyataan continue
6. Pernyataan for..in
7. pernyataan with
8. Pernyataan var
9. Pernyataan komentar

Dalam modul ini kita akan membahas pernyataan *if..else* sampai *for* , sedangkan pernyataan lainnya akan dibahas pada modul selanjutnya.

Pernyataan if...else

Pernyataan ini digunakan untuk membandingkan suatu kondisi, jika kondisi itu benar, maka akan dijalankan ekspresi tertentu, tetapi jika kondisi itu salah akan dijalankan ekspresi lain. Sintaks dari pernyataan if.....else,

```
If (kondisi)  
    {ekspresi 1}  
else (kondisi)  
    {ekspresi 2}
```

Keterangan :

- Kondisi : kondisi yang dibandingkan
- Ekspresi-1 : ekspresi-ekspresi yang akan dijalankan jika kondisi memenuhi syarat.
- Ekspresi-2 : ekspresi-ekspresi yang akan dijalankan jika kondisi tidak memenuhi syarat.

Contoh :

Agar lebih memahami maksud dari pernyataan di atas, simak kalimat berikut :

Jika nilai ujian saya meningkat, saya akan mendapatkan mobil dari orangtua saya, tetapi jika tetap atau berkurang maka saya tidak mendapatkan mobil (else) melainkan saya hanya mendapatkan sepeda.

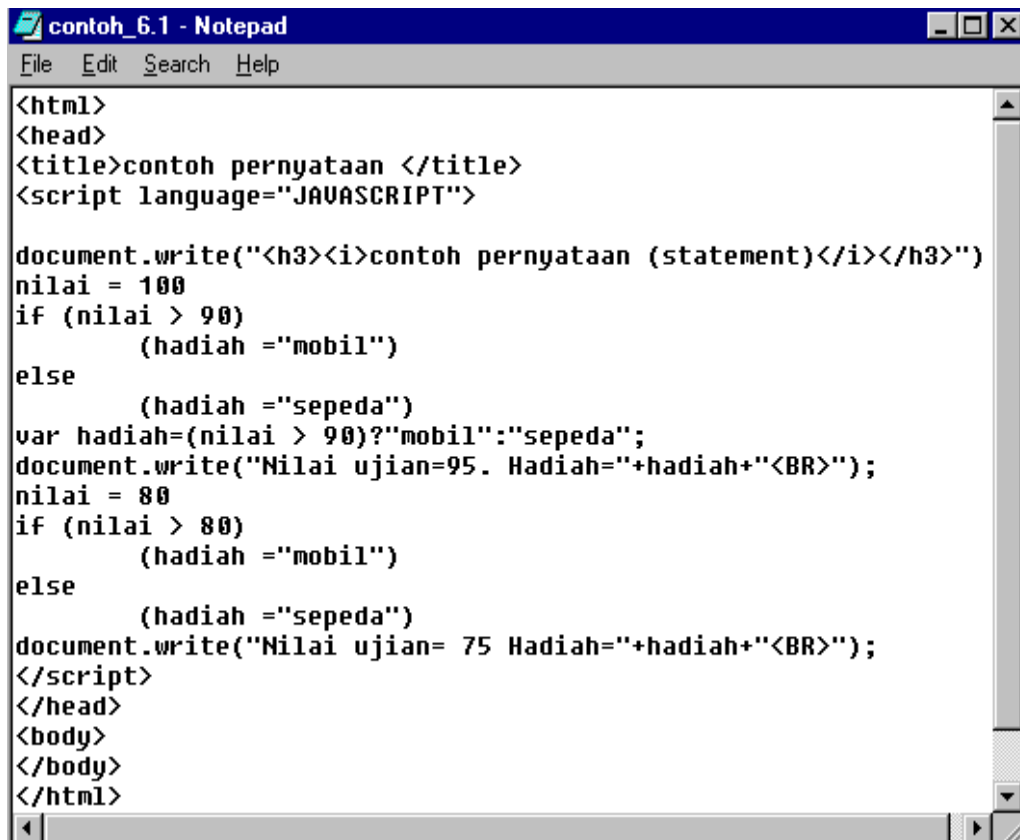
Jika kita terjemahkan ke dalam bahasa javascript maka kalimatnya adalah :

```
If (nilai > 90)  
    (hadiah = "mobil")  
else  
    (hadiah="sepeda")
```

keterangan :

Baris	Keterangan
if (nilai > 90) (hadiah = "mobil")	Jika isi variable <i>nilai</i> bernilai lebih besar dari 90 maka variable <i>hadiah</i> akan diberikan nilai " <i>mobil</i> "
Else (hadiah="sepeda")	Jika isi variable <i>nilai</i> bernilai tidak lebih besar dari 90 maka variable <i>hadiah</i> akan diberikan nilai " <i>sepeda</i> "

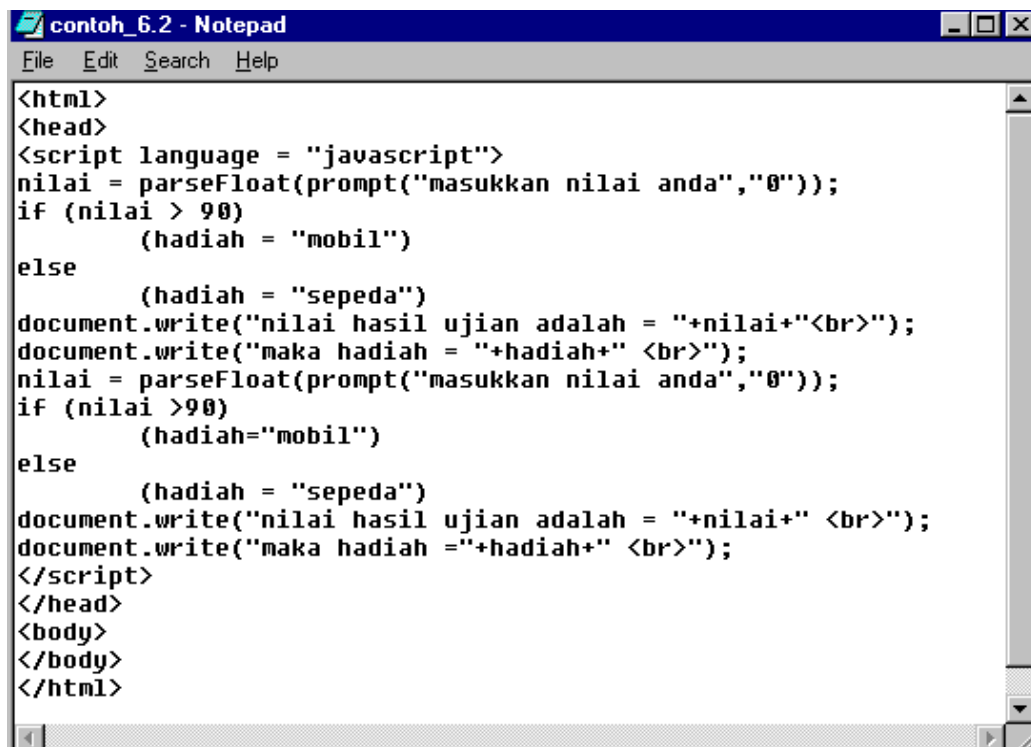
Perhatikan contoh_6.1.html di bawah ini :



```
<html>
<head>
<title>contoh pernyataan </title>
<script language="JAVASCRIPT">

document.write("<h3><i>contoh pernyataan (statement)</i></h3>")
nilai = 100
if (nilai > 90)
    (hadiah ="mobil")
else
    (hadiah ="sepeda")
var hadiah=(nilai > 90)?"mobil":"sepeda";
document.write("Nilai ujian=95. Hadiah="+hadiah+"<BR>");
nilai = 80
if (nilai > 80)
    (hadiah ="mobil")
else
    (hadiah ="sepeda")
document.write("Nilai ujian= 75 Hadiah="+hadiah+"<BR>");
</script>
</head>
<body>
</body>
</html>
```

Perhatikan contoh_6.2.html di bawah ini :



```
<html>
<head>
<script language = "javascript">
nilai = parseFloat(prompt("masukkan nilai anda","0"));
if (nilai > 90)
    (hadiah = "mobil")
else
    (hadiah = "sepeda")
document.write("nilai hasil ujian adalah = "+nilai+"<br>");
document.write("maka hadiah = "+hadiah+" <br>");
nilai = parseFloat(prompt("masukkan nilai anda","0"));
if (nilai >90)
    (hadiah="mobil")
else
    (hadiah = "sepeda")
document.write("nilai hasil ujian adalah = "+nilai+" <br>");
document.write("maka hadiah = "+hadiah+" <br>");
</script>
</head>
<body>
</body>
</html>
```


Dari contoh di atas maka browser akan meminta masukan dari user yang di jadikannya sebagai variable yang akan di cocokkan pada sintak pernyataan yang telah dibuat pada text editor.

Pernyataan while

Pernyataan `while` digunakan untuk melakukan suatu loop atau menjalankan ekspresi berulang-ulang selama (`while`) kondisi yang ditentukan bernilai benar.

Sintak dari pernyataan `while` adalah sebagai berikut :

```
While (kondisi)  
(ekspresi)
```

Keterangan :

- Kondisi : kondisi yang di bandingkan
- Ekspresi : ekspresi-ekspresi yang akan dijalankan jika kondisi memenuhi syarat.

Contohnya :

Jika seorang nasabah bank ingin menabung si suatu bank dengan dana awal Rp.100.100,-. Dia tidak akan berhenti menabung sampai kas saldonya mencapai satu juta. Maka jika dituangkan pada suatu sintak javascript, sebagai berikut :

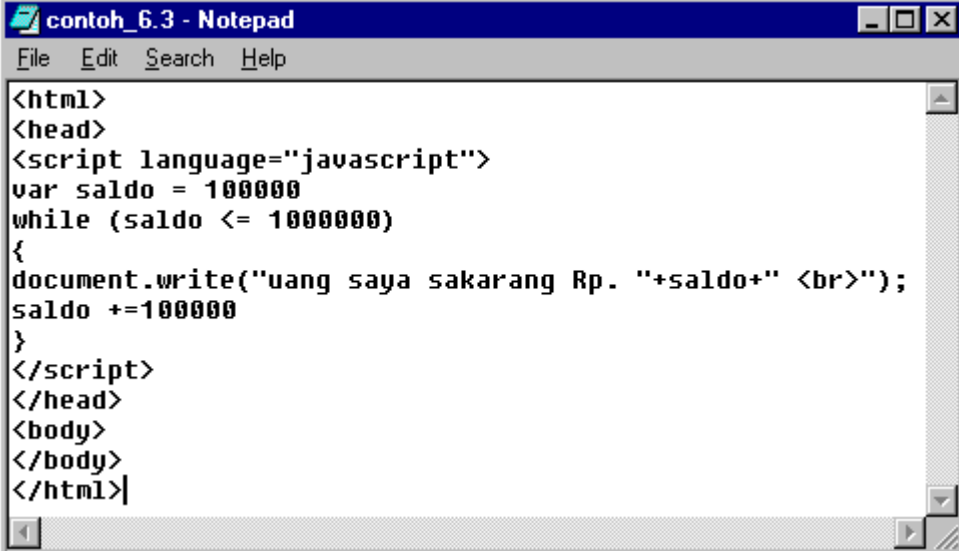
```
var saldo = 100000  
while (saldo <=1000000)  
(saldo+=100000)
```

Dari sintak di atas maka dapat di uraikan sebagai berikut :

- `var saldo = 100000` ; aritnya adalah variable saldo di beri nilai awal 1000
- `while (saldo <= 1000000)`; artinya adalah memeriksa kondisi apakah isi variable `saldo` kurang dari atau sama dengan 1000000. Jika benar, maka pernyataan berikutnya akan dijalankan.

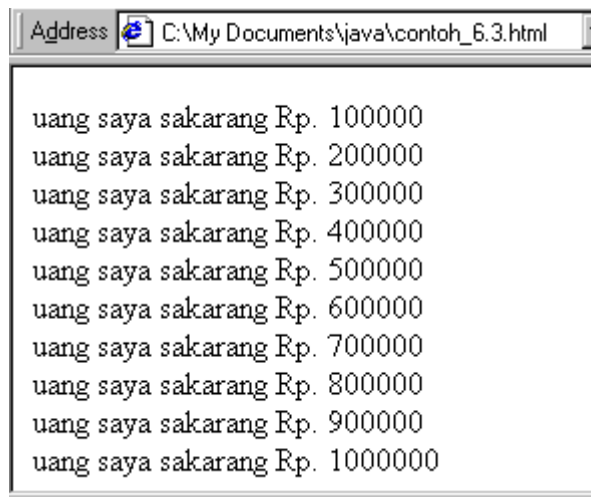
- (saldo +=1000); artinya sama dengan saldo= saldo+1000. Variable saldo yang baru diisi dengan variable uang 1000 maka akan ditambah dengan 1000 sehingga jumlahnya kini menjadi 2000, dan begitu seterusnya.

Perhatikan contoh_6.3.html :



```
<html>
<head>
<script language="javascript">
var saldo = 100000
while (saldo <= 1000000)
{
document.write("uang saya sakarang Rp. "+saldo+" <br>");
saldo +=100000
}
</script>
</head>
<body>
</body>
</html>
```

Di dalam browser akan terlihat seperti di bawah ini :



Pernyataan for

Pernyataan for berfungsi melakukan suatu loop atau menjalankan ekspresi berulang-ulang untuk kondisi yang telah di tentukan sebelumnya. Pernyataan for jika dilihat hampir sama dengan pernyataan while, yang berbeda adalah

pernyataan for menyertakan kondisi awal dan kondisi akhir serta besarnya kenaikan.

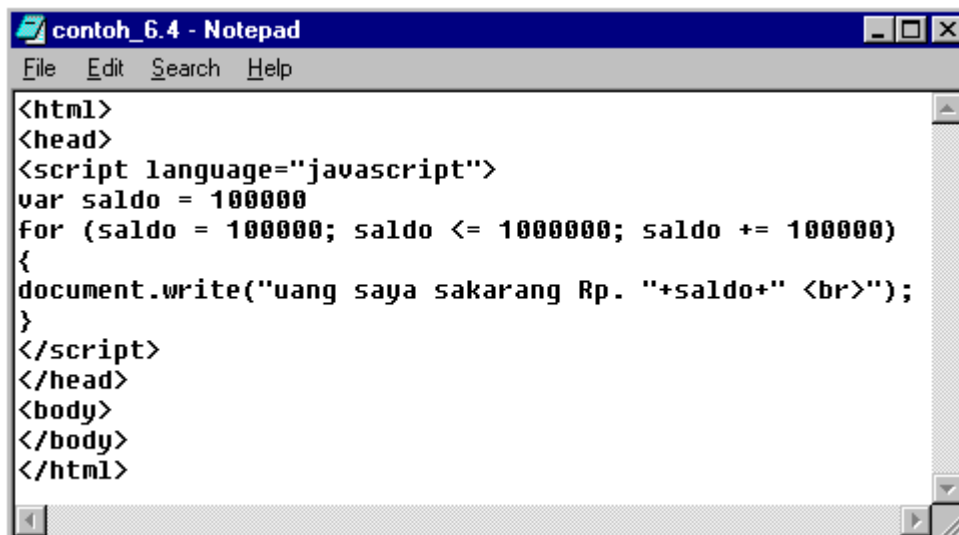
Sintak pernyataan for adalah :

```
for (kondisi_awal; kondisi_akhir; kenaikan)
    (ekspresi)
```

Keterangan :

- *kondisi_awal* : kondisi awal yang memenuhi untuk loop dimulai
- *kondisi_akhir* : kondisi akhir yang memenuhi untuk loop diakhiri
- *kenaikan* : besarnya kenaikan nilai variable untuk setiap iterasi loop

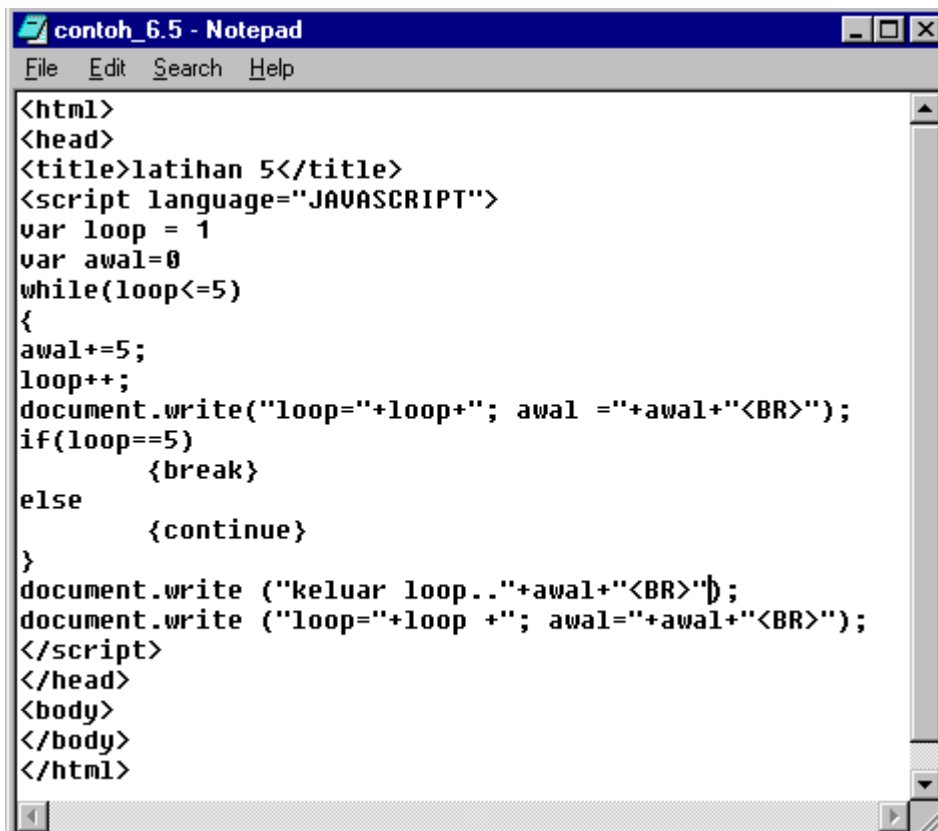
Perhatikan contoh_6.4.html di bawah ini :



```
<html>
<head>
<script language="javascript">
var saldo = 100000
for (saldo = 100000; saldo <= 1000000; saldo += 100000)
{
document.write("uang saya sakarang Rp. "+saldo+" <br>");
}
</script>
</head>
<body>
</body>
</html>
```

Pernyataan Break

Pernyataan (statement) break berfungsi sebagai menghentikan ekspresi loop yang sedang dijalankan, dan melanjutkan ke baris skrip di bawah loop tersebut. Jika pengulangannya belum selesai sampai pada statement yang di syaratkan maka ia akan tetap me loop, dan berhenti pada kondisi yang telah ditetapkan di dalam sintak javascript. Perhatikan contoh_6.5.html di bawah ini :



```
<html>
<head>
<title>latihan 5</title>
<script language="JAVASCRIPT">
var loop = 1
var awal=0
while(loop<=5)
{
awal+=5;
loop++;
document.write("loop="+loop+"; awal =" +awal+"<BR>");
if(loop==5)
    {break}
else
    {continue}
}
document.write ("keluar loop.." +awal+"<BR>");
document.write ("loop="+loop +"; awal="+awal+"<BR>");
</script>
</head>
<body>
</body>
</html>
```

PERNYATAAN (lanjutan) DAN FUNGSI

- Pernyataan Continue
- Pernyataan for...in
- Pernyataan with
- Pernyataan var
- Pernyataan Komentar

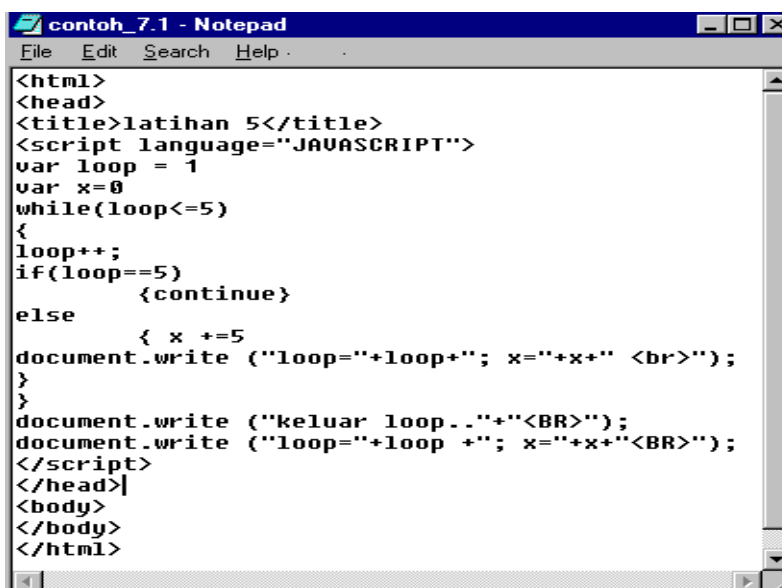
Fungsi

- Pengertian Fungsi
- Pembuatan Fungsi
- Aturan pembuatan Javascript
- Memanggil Fungsi javascript

Pernyataan Continue

Pernyataan *continue* berfungsi untuk menghentikan eksekusi loop. Tetapi hal ini tidak menyebabkan keluar dari loop tetapi hanya melompati satu buah iterasi dan melanjutkan iterasi loop yang selanjutnya.

Perhatikan contoh_7.1.html di bawah ini :



```
contoh_7.1 - Notepad
File Edit Search Help
<html>
<head>
<title>latihan 5</title>
<script language="JAVASCRIPT">
var loop = 1
var x=0
while(loop<=5)
{
loop++;
if(loop==5)
{continue}
else
{ x +=5
document.write ("loop="+loop+"; x="+x+" <br>");
}
}
document.write ("keluar loop.."+<BR>");
document.write ("loop="+loop +"; x="+x+"<BR>");
</script>
</head>
<body>
</body>
</html>
```

Pernyataan for...in

Pernyataan for...in adalah pernyataan yang melakukan loop(pengulangan) di dalam sebuah objek.

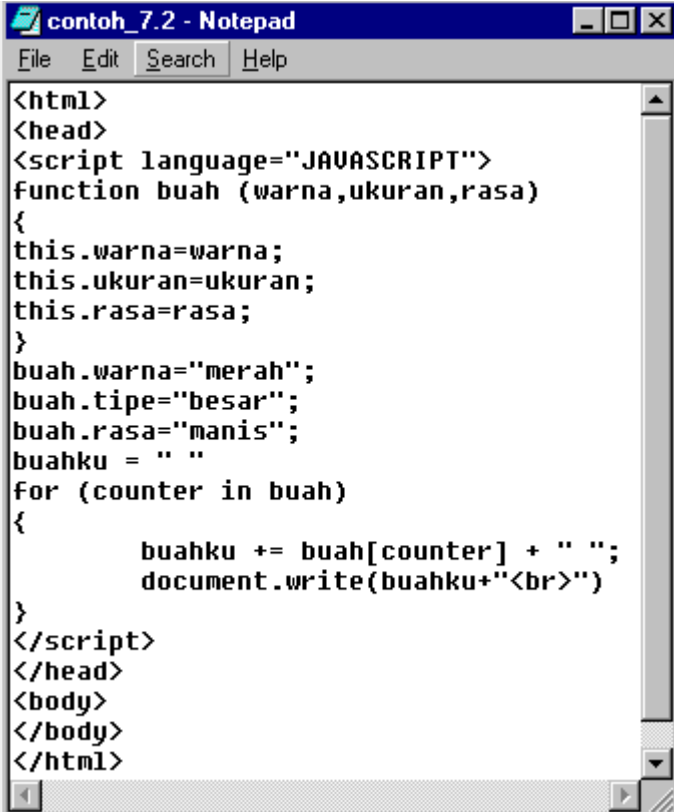
Sintak dari pernyataan for...in adalah :

```
for (counter in nama_object)
    {ekspresi}
```

Keterangan :

- Counter : jumlah counter di dalam object (jumlah property)
- Nama_object : objek yang ingin dilakukan loop
- Ekspresi : ekspresi yang dijalankan selama loop

Perhatikan contoh_7.2.html di bawah ini :



```
contoh_7.2 - Notepad
File Edit Search Help
<html>
<head>
<script language="JAVASCRIPT">
function buah (warna,ukuran,rasa)
{
this.warna=warna;
this.ukuran=ukuran;
this.rasa=rasa;
}
buah.warna="merah";
buah.tipe="besar";
buah.rasa="manis";
buahku = " "
for (counter in buah)
{
    buahku += buah[counter] + " ";
    document.write(buahku+"<br>")
}
</script>
</head>
<body>
</body>
</html>
```

Pada contoh di atas, objeknya adalah buah sedangkan property untuk objek buah adalah warna, tipe dan rasa. Setiap property memiliki nomor index yang dimulai dari 0. Jadi pada objek buah di atas memiliki counter sebagai berikut :

- buah [0] = buah.warna

- buah [1] = buah.besar
- buah[2] = buah.rasa

Pernyataan with

Seperti halnya dengan pernyataan for...in yang di atas, dalam pernyataan ini ada sedikit perbedaan yaitu pernyataan with berfungsi untuk memberi tahu objek yang sedang kita gunakan. Setiap referensi variable di dalam pernyataan berikutnya akan merujuk kepada objek tersebut.

Sintak pernyataan with sebagai berikut :

```
with (nama_object)
    (ekspresi)
```

Keterangan :

- nama_object : nama objek yang ingin di jadikan acuan pernyataan
- ekspresi : ekspresi yang terdapat di dalam objek

contoh :

```
with (buah)
```

```
{
    warna = "merah";
    ukuran = "besar";
    rasa = "manis";
}
```

Sintak lengkapnya, anda cukup memodifikasi contoh seperti di atas (contoh_7.2.html).

Pernyataan var

Pernyataan var berfungsi untuk menugaskan suatu nilai ke dalam variable.

Sintak dari statement (pernyataan) var adalah :

```
var nama_variable = value
```

contoh :

```
var nama = "firdaus"
```

```
var a = 5
```

Pada modul sebelumnya, sebenarnya kita telah mempelajarinya yaitu pada modul variable mengenai nilai dari variable.

Sebenarnya kita bisa memberikan nilai untuk variable tanpa pernyataan var tersebut, seperti :

```
nama="firdaus"
```

Pernyataan komentar

Pernyataan komentar berfungsi untuk menerangkan atau memberi keterangan pada setiap barisnya yang ada di dalam sintak HTML. Pernyataan ini memudahkan bagi user untuk mempelajari script javascript yang kita buat, selain itu komentar yang kita buat dalam script tersebut tidak akan terlihat di browser dan hanya bisa terlihat di source-nya saja, karena browser tidak ikut memproses sintak komentar tersebut.

Pada modul sebelumnya, sebenarnya kita telah mempelajari dan membahasnya.

Sintaknya ada dua yaitu :

1. *//komentar*
2. */* komentar
komentar */*

FUNGSI

Pengertian Fungsi

Fungsi adalah sekumpulan kode-kode javascript yang jika dijalankan akan menghasilkan atau mengerjakan suatu tugas tertentu. Fungsi dapat di sebut sebagai modul atau subprogram dari seluruh skrip atau program javascript.

Pembuatan Fungsi

Dilihat dari segi pembuatan fungsi, maka sintak dasarnya adalah :


```
function nama_fungsi (parameter)
{ .....
    ekspresi
    ekspresi
    .....
}
```

Untuk memposisikan sintak tersebut, maka sintak tersebut harus diletakkan di :

```
<script language="javascript">
function nama_fungsi()
{
    ### isi fungsi ###
}
</script>
```

Aturan Pembuatan Fungsi

Ditinjau dari segi pembuatannya, maka fungsi memiliki beberapa aturan pembuatan yang akan di jabarkan di bawah ini :

1. Diawali dengan kata *function*
2. Nama *fungsi* bebas, asalkan dapat menjelaskan fungsi dari *fungsi* tersebut. Yang terpenting di sini adalah *fungsi* harus diikuti oleh tanda kurung “()” yang digunakan sebagai tempat parameter.
3. Di dalam tanda kurung setelah kata *function*, dapat diletakkan parameter fungsi.

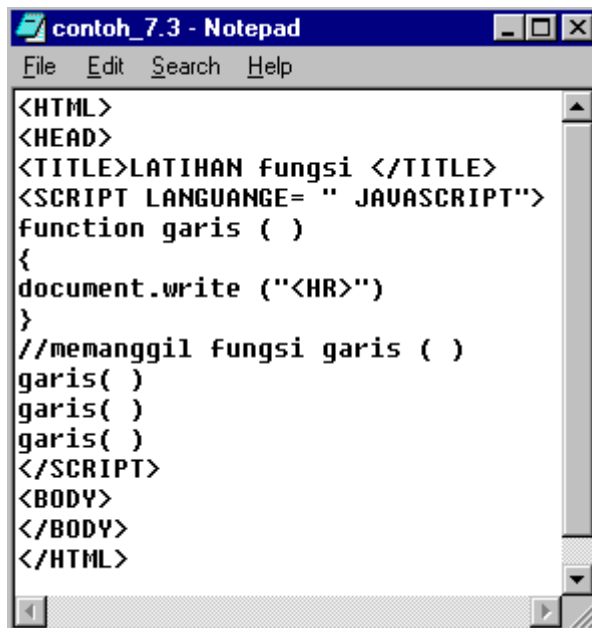
Untuk lebih jelas lagi perhatikan sintak berikut :

```
function jumlah (a,b)
{
    var c = a+b;
    document.write (z)
}
```

Parameter di dalam fungsi bisa digunakan bisa juga tidak digunakan. Parameter merupakan variable atau nilai yang akan diolah ekspresi-ekspresi yang terdapat di dalam fungsi tersebut.

4. Isi fungsi harus di letakkan di dalam tanda kurung kurawal.

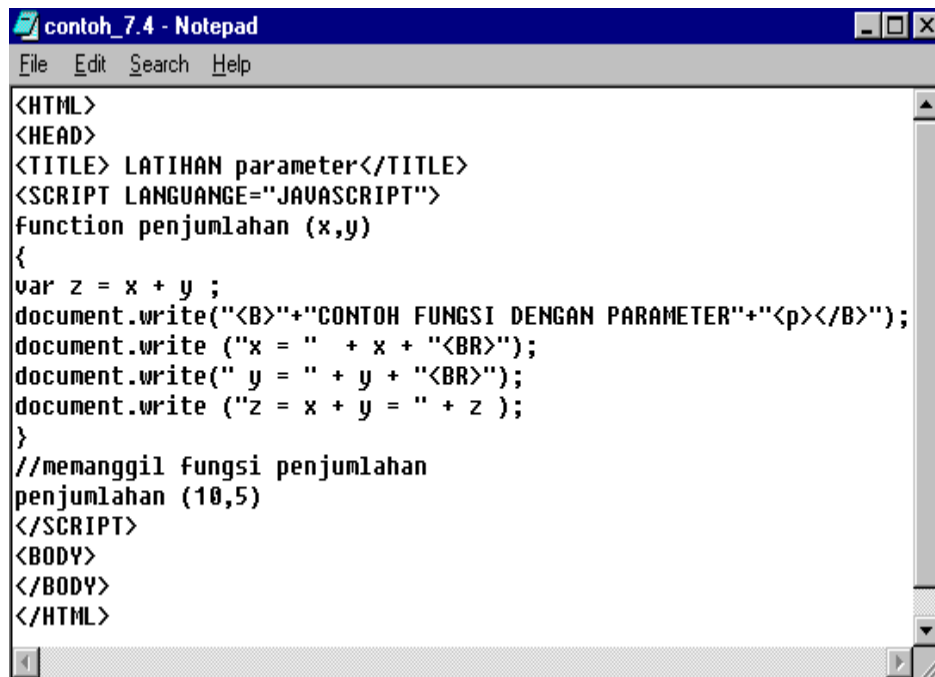
Untuk lebih jelas lagi, perhatikan contoh_7.3.html :



```
<HTML>
<HEAD>
<TITLE>LATIHAN fungsi </TITLE>
<SCRIPT LANGUAGE= " JAVASCRIPT">
function garis ( )
{
document.write ("<HR>")
}
//memanggil fungsi garis ( )
garis( )
garis( )
garis( )
</SCRIPT>
<BODY>
</BODY>
</HTML>
```

Perhatikan contoh di atas. Tampak bahwa dengan menuliskan garis() berarti kita telah memanggil sintak "<hr>" yang di nyatakan di dalam sintak document.write. Ini disebabkan bahwa kata garis merupakan nama sebuah fungsi yang bisa dipanggil pada posisi manapun selama nama fungsi tersebut masih berada di antara sintak javascript.

Pada contoh di atas, tampak bahwa isi parameter masih kosong. Di bawah ini ada script javascript yang memberikan nilai parameter pada fungsi. Perhatikan contoh_7.4.html di bawah ini :



```
<HTML>
<HEAD>
<TITLE> LATIHAN parameter</TITLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
function penjumlahan (x,y)
{
var z = x + y ;
document.write("<B>"+"CONTOH FUNGSI DENGAN PARAMETER"+"<p></B>");
document.write ("x = " + x + "<BR>");
document.write(" y = " + y + "<BR>");
document.write ("z = x + y = " + z );
}
//memanggil fungsi penjumlahan
penjumlahan (10,5)
</SCRIPT>
<BODY>
</BODY>
</HTML>
```

Memanggil Fungsi javascript

Untuk menampilkan fungsi yang telah di buat, maka fungsi tersebut harus dipanggil terlebih dahulu. Ada beberapa cara pemanggilan fungsi tersebut, yaitu :

1. Fungsi tersebut dipanggil dalam skrip javascript
2. Fungsi dipanggil dari dalam fungsi itu sendiri
3. Fungsi dipanggil dari dalam fungsi yang lain.

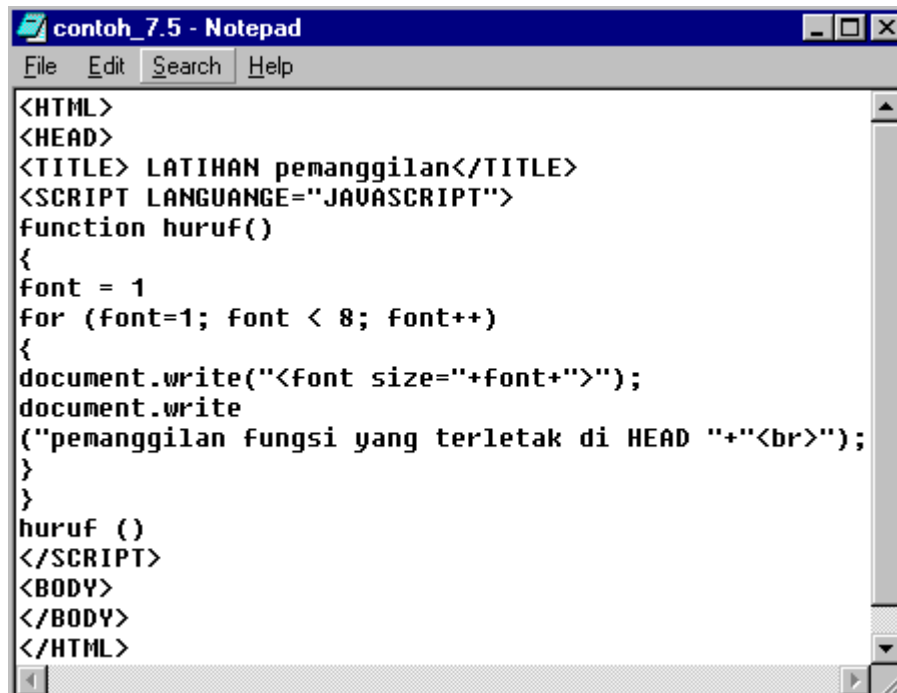
Di bawah ini akan dijelaskan secara lebih rinci mengenai metode pemanggilan fungsi javascript tersebut :

a. Memanggil fungsi dari dalam skrip javascript

Metode ini terbagi dua :

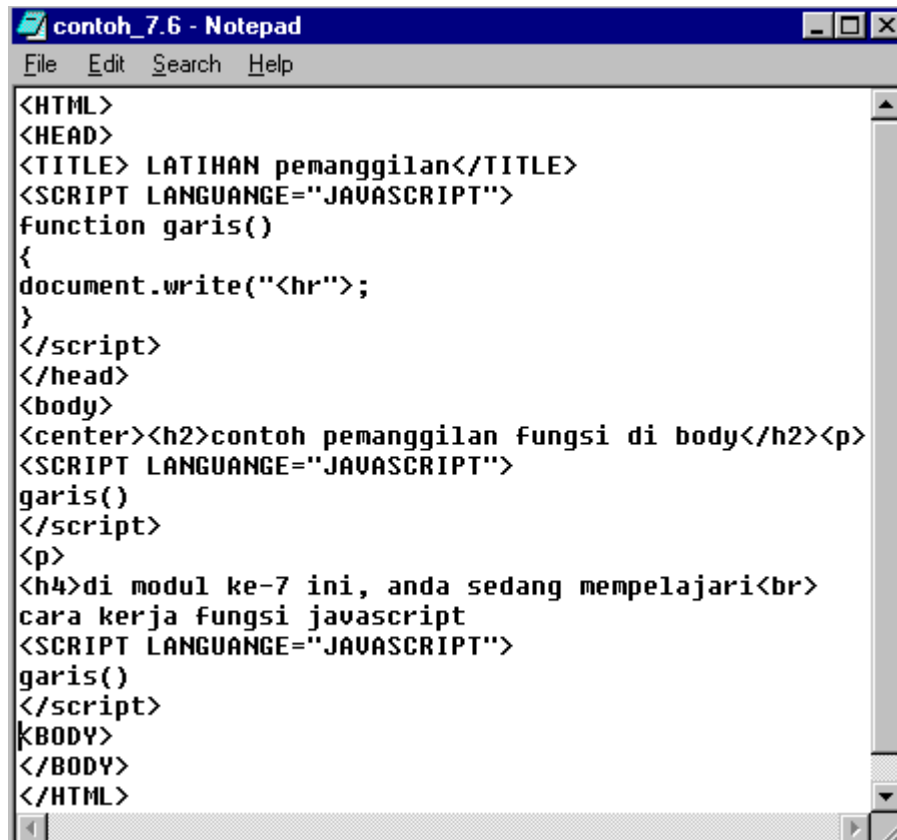
1. Fungsi dipanggil dari skrip yang terletak pada bagian <head>

Perhatikan contoh di bawah ini :



```
<HTML>
<HEAD>
<TITLE> LATIHAN pemanggilan</TITLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
function huruf()
{
font = 1
for (font=1; font < 8; font++)
{
document.write("<font size="+font+"");
document.write
("pemanggilan fungsi yang terletak di HEAD "+"<br>");
}
}
huruf ()
</SCRIPT>
<BODY>
</BODY>
</HTML>
```

2. Fungsi dipanggil dari skrip javascript yang terletak pada bagian <body>
Perhatikan contoh_7.6.html berikut :



```
<HTML>
<HEAD>
<TITLE> LATIHAN pemanggilan</TITLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
function garis()
{
document.write("<hr>");
}
</script>
</head>
<body>
<center><h2>contoh pemanggilan fungsi di body</h2><p>
<SCRIPT LANGUAGE="JAVASCRIPT">
garis()
</script>
<p>
<h4>di modul ke-7 ini, anda sedang mempelajari<br>
cara kerja fungsi javascript
<SCRIPT LANGUAGE="JAVASCRIPT">
garis()
</script>
<BODY>
</BODY>
</HTML>
```

FUNGSI (lanjutan)

- Memanggil Fungsi Dari Dalam Fungsi
- Memanggil Fungsi Dari Fungsi Lain
- Fungsi Built-in

Memanggil Fungsi Dari Dalam Fungsi

Fungsi ini disebut juga dengan fungsi rekursif yaitu fungsi berulang. Maksudnya adalah fungsi tersebut dipanggil dari dalam fungsi itu sendiri.

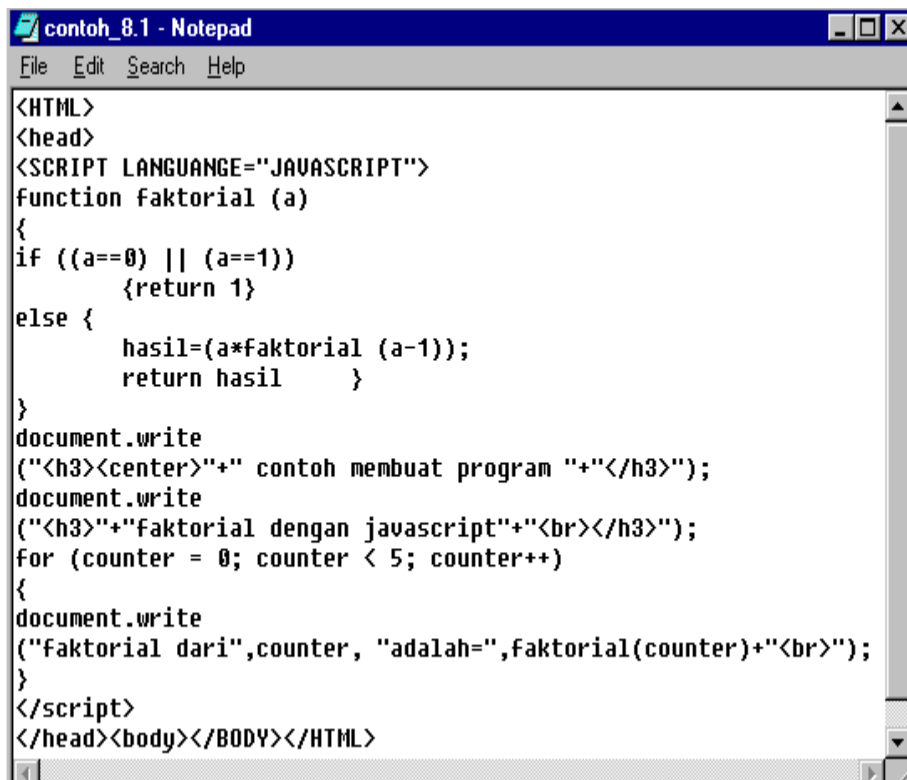
Sintaknya adalah :

```
<script language = "javascript">  
function nama_function()  
{ ...  
  nama_fungsi()  
  ...  
}  
</script>
```

Perhatikan contoh_8.1.html di bawah ini :

Contoh yang paling mudah untuk ditemukan adalah contoh menghitung nilai factorial.

Perhatikan contoh_8.1.html berikut :



```
<HTML>
<head>
<SCRIPT LANGUAGE="JAVASCRIPT">
function faktorial (a)
{
if ((a==0) || (a==1))
    {return 1}
else {
    hasil=(a*faktorial (a-1));
    return hasil    }
}
document.write
("<h3><center>"+ " contoh membuat program "+ "</h3>");
document.write
("<h3>"+ "faktorial dengan javascript"+ "<br></h3>");
for (counter = 0; counter < 5; counter++)
{
document.write
("faktorial dari",counter, "adalah=",faktorial(counter)+"<br>");
}
</script>
</head><body></BODY></HTML>
```

Dari script di atas, dapat dilihat bahwa rumus yang telah ada harus dipanggil kembali.

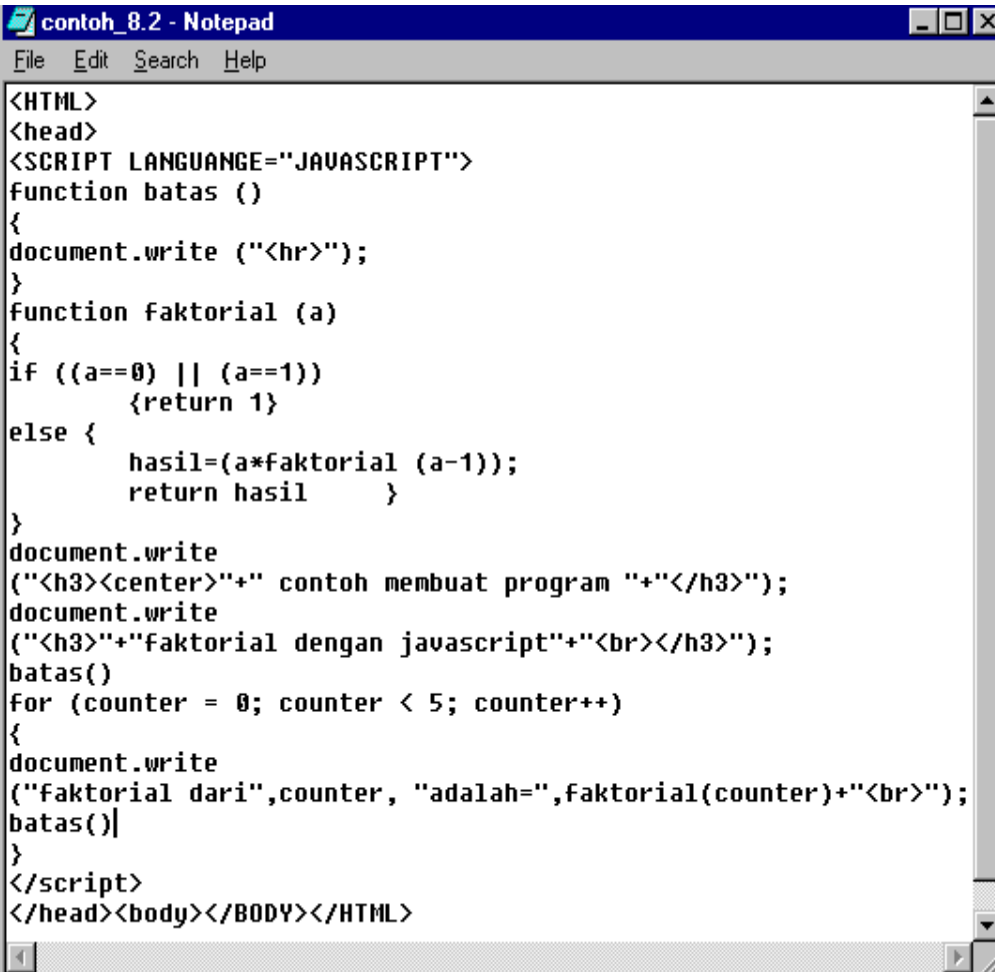
Memanggil Fungsi dari Fungsi Lainnya

Dalam membuat suatu fungsi, kita juga dapat memanggil fungsi lain yang telah kita buat sebelumnya, dengan kata lain kita membuat suatu fungsi terlebih dahulu kemudian membuat fungsi lainnya yang memanggil fungsi sebelumnya.

Sintaknya adalah :

```
<script language="javascript">
function nama_fungsi1 ()
{
...
}
function nama_fungsi2 ()
{
... nama_fungsi1()
}
</script>
```

Untuk lebih jelas lagi, perhatikan contoh_8.2.html di bawah ini :



```
<HTML>
<head>
<SCRIPT LANGUAGE="JAVASCRIPT">
function batas ()
{
document.write ("<hr>");
}
function faktorial (a)
{
if ((a==0) || (a==1))
{return 1}
else {
hasil=(a*faktorial (a-1));
return hasil }
}
document.write
("<h3><center>"+ " contoh membuat program "+ "</h3>");
document.write
("<h3>"+ "faktorial dengan javascript"+ "<br></h3>");
batas()
for (counter = 0; counter < 5; counter++)
{
document.write
("faktorial dari",counter, "adalah=",faktorial(counter)+"<br>");
batas()}
}
</script>
</head><body></BODY></HTML>
```

Dari contoh di atas, tampak bahwa ada dua jenis fungsi yang berbeda, fungsi pertama menyatakan garis, dan fungsi kedua menyatakan nilai factorial. Kondisi di atas, fungsi garis dapat dipanggil di dalam fungsi faktorial sehingga selain menampilkan nilai factorial, maka akan tampil juga fungsi garis di antara fungsi factorial.

Fungsi Built-in

Di dalam javascript juga menyediakan beberapa fungsi built-in yang dapat langsung digunakan. Fungsi-fungsi tersebut adalah :

1. parseInt()

2. parseFloat()
3. isNaN()
4. eval()
5. escape()
6. unescape()

Untuk lebih jelasnya lagi, perhatikan penjabaran dari tiap fungsi di atas yang ada di bawah ini :

1. Fungsi parseInt()

Fungsi parseInt() berguna untuk mengubah dan mengkonversi nilai string menjadi nilai integer dengan menggunakan basis tertentu.

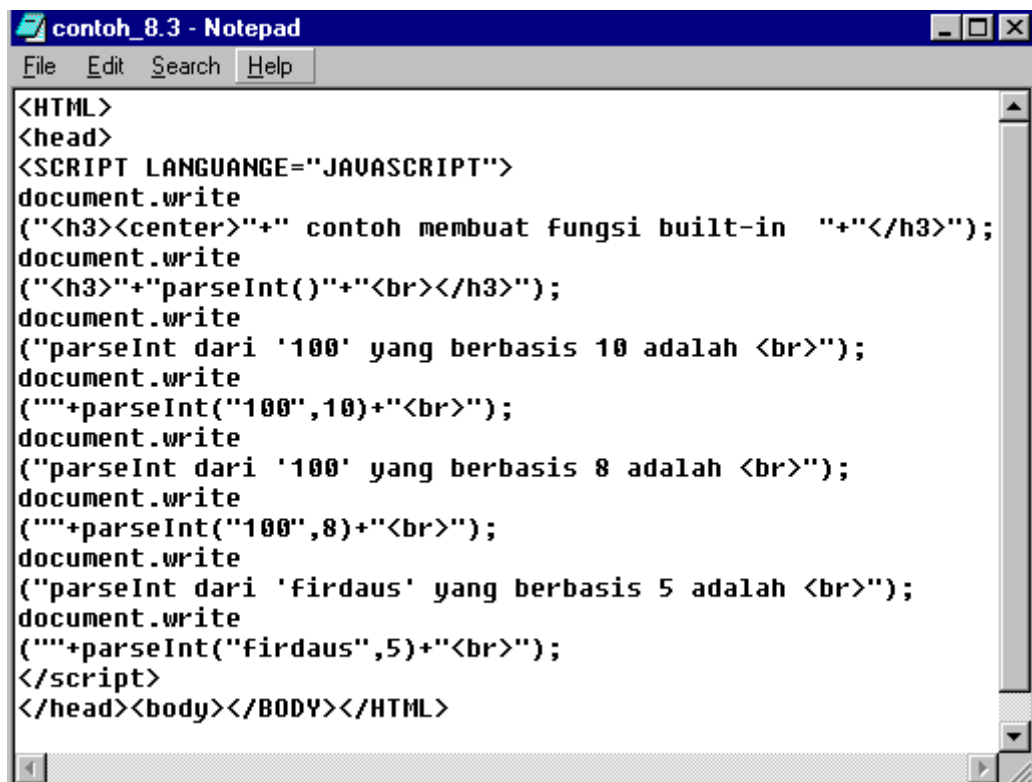
Misalnya :

String "100.12" jika diubah menjadi integer dengan basis sepuluh maka hasilnya adalah 100 karena nilai integer saja yang akan dikonversi.

Sintaknya adalah :

parseInt (string, basis)

Perhatikan contoh_8.3.html di bawah ini :



```
<HTML>
<head>
<SCRIPT LANGUAGE="JAVASCRIPT">
document.write
("<h3><center>"+ " contoh membuat fungsi built-in "+ "</h3>");
document.write
("<h3>"+ "parseInt()"+ "<br></h3>");
document.write
("parseInt dari '100' yang berbasis 10 adalah <br>");
document.write
("'+parseInt("100",10)+'<br>");
document.write
("parseInt dari '100' yang berbasis 8 adalah <br>");
document.write
("'+parseInt("100",8)+'<br>");
document.write
("parseInt dari 'firdaus' yang berbasis 5 adalah <br>");
document.write
("'+parseInt("firdaus",5)+'<br>");
</script>
</head><body></BODY></HTML>
```

- 2 Fungsi parseFloat()

Fungsi dari fungsi `parseFloat()` adalah untuk mengubah atau mengkonversi nilai string menjadi nilai floating-point. Nilai floating-point adalah bilangan pecahan atau pangkat).

Sintak fungsi tersebut adalah :

parseFloat (string)

Perhatikan contoh `_8.4.html` di bawah ini :

```
<HTML><head>
<SCRIPT LANGUAGE="JAVASCRIPT">
document.write
("<h3><center>"+ " contoh membuat fungsi built-in "+ "</h3>");
document.write
("<h3>"+ "parseFloat()"+ "<br></h3>");
document.write
("parseFloat dari '-100.12' adalah <br>");
document.write
("'+parseFloat("-100.12")+ "<br>");
document.write
("parseFloat dari 'smkti' adalah "+parseFloat("smkti")+ "<br>");
</script>
</head><body></BODY></HTML>
```

3. Fungsi `isNaN`

Fungsi `isNaN()` yang merupakan singkatan dari *is Not a Number* digunakan untuk menentukan apakah suatu nilai berupa bilangan numerik atau bukan bilangan numerik.

Dengan kata lain bahwa :

- Jika nilai tersebut bukan bilangan numeric, maka fungsi `isNaN` akan menghasilkan nilai `true` (benar).
- Jika nilai tersebut bilangan numeric, maka fungsi `isNaN` akan menghasilkan nilai `false` (salah).

Sintak dari fungsi tersebut adalah :

isNaN (value)

Perhatikan contoh `_8.5.html` di bawah ini :

```
contoh_8.5 - Notepad
File Edit Search Help
<HTML><head>
<SCRIPT LANGUAGE="JAVASCRIPT">
document.write
("<h3><center>"+ " contoh membuat fungsi built-in "+ "</h3>");
document.write
("<h3>"+ "isNaN()"+ "<br></h3>");
document.write
("isNaN dari '100.12' adalah "+ isNaN("100.12")+ " <br>");
document.write
("isNaN dari 'smkti2001' adalah "+ isNaN("smkti2001")+ " <br>");
document.write
("isNaN dari 'smkti' adalah "+ isNaN("smkti")+ "<br>");
</script>
</head><body></BODY></HTML>
```

4. Fungsi eval()

Fungsi eval() digunakan untuk menghitung suatu ekspresi aritmatika.

Sintaknya adalah

Eval (ekspresi)

Perhatikan contoh_8.6.html di bawah ini :

```
contoh_8.6 - Notepad
File Edit Search Help
<HTML>
<head>
<SCRIPT LANGUAGE="JAVASCRIPT">
var aritmatika = prompt ("masukkan ekspresi arimatika anda","0");
document.write
("<h3><center>"+ " hasilnya adalah "+ "</h3>");
document.write
("<h3>"+ eval(aritmatika)+ "<br></h3>");
</script>
</head>
<body></BODY></HTML>
```

Ketika halaman ini akan tampil , maka akan tampak kolom tempat memasukkan ekspresi artimatika. Jika kita masukkan suatu ekspresi aritmatika seperti :



Angka 10 di atas merupakan variable yang akan di tampilkan oleh sintak pernyataan, maka akan tampak hasilnya seperti di bawah ini :



EVENT

- Pengantar Event
- Menjalankan Event
- Beberapa Aplikasi Event

Pengantar Event

Event adalah suatu aksi yang dilakukan oleh pemakai yang memicu jalannya kode JavaScript anda. sebagian besar event dipicu oleh pemakai yang memanipulasi halaman web pada program browsernya , seperti menggeser pointer keatas link untuk memicu menampilkan alamat link pada baris status.

Macam-macam event :

1. Click , event ini terjadi jika pemakai mengklik tombol mouse pada link atau elemen yang terdapat form.
2. Focus, event ini terjadi jika pemakai mengklik atau meletakkan pointer mouse pada elemen form seperti field teks , kotak cek dan sebagainya.
3. Blur , event ini terjadi jika pemakai menyingkirkan focus (pointer mouse) dari elemen form yang sebelumnya diberi focus.
4. Change , event ini terjadi jika pemakai mengubah input atau masukan pada elemen form.
5. MouseOver, terjadi jika pemakai meletakkan mouse di atas sebuah link .
6. Select, event ini terjadi jika pemakai memilih teks pada elemen form , seperti menggeser pointer mouse pada teks sambil menahan tombol kiri mouse.
7. Submit, event ini terjadi bila mengklik tombol “submit”

Menjalankan Event

Event Handler

Event Handler merupakan kode-kode javascript yang akan dijalankan apabila event tersebut dipicu.

Untuk menjalankan event tersebut maka digunakan sintaks :

On NamaEvent = " kode JavaScript "

Contoh :

OnMouseOver = "fungsi_data (a,b)"

Memanggil Event Handler

Metode pemanggilan event handler dapat menggunakan dua metode, yaitu :

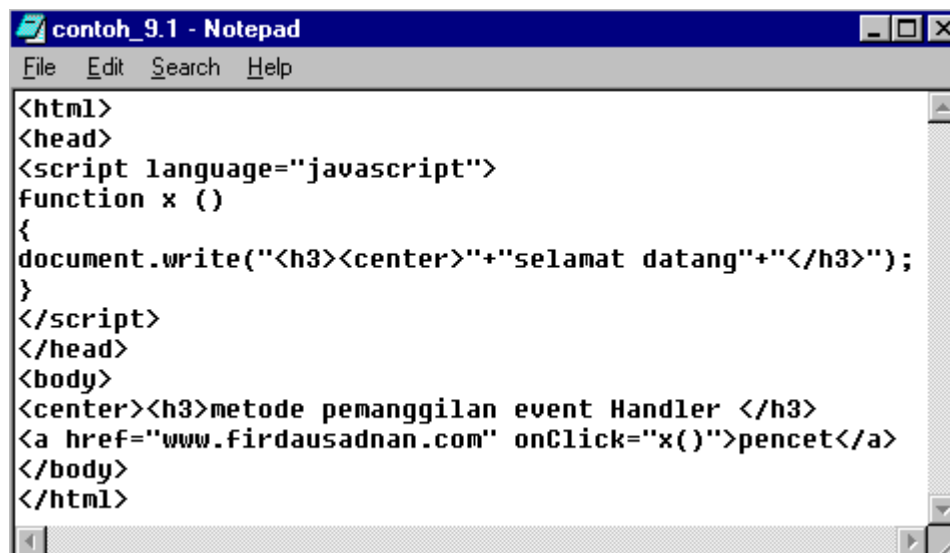
1. Memanggil melalui fungsi

Event Handler dapat dijalankan dengan cara memanggil fungsi.

Sintaknya adalah :

OnNamaEvent = "namafungsi (parameter)"

Untuk lebih jelasnya lagi, perhatikan contoh_9.1.html di bawah ini :



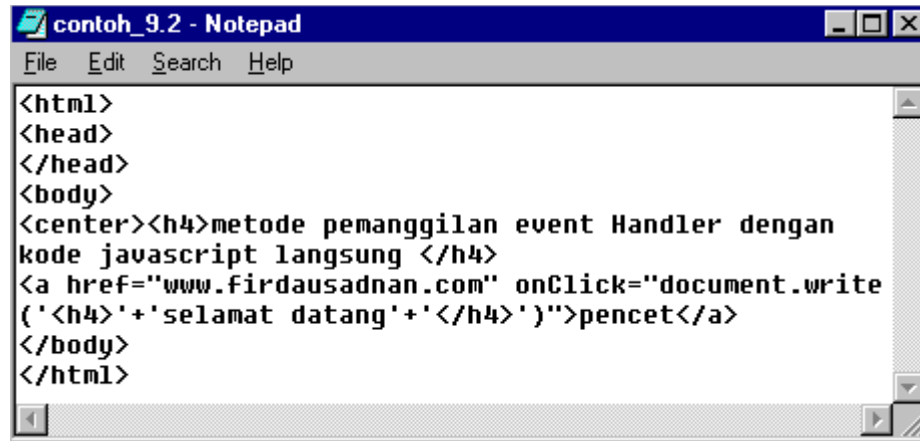
```
<html>
<head>
<script language="javascript">
function x ()
{
document.write("<h3><center>"+"selamat datang"+"</h3>");
}
</script>
</head>
<body>
<center><h3>metode pemanggilan event Handler </h3>
<a href="www.firdausadnan.com" onClick="x()">pencet</a>
</body>
</html>
```

2. Memanggil dengan menjalankan kode javascript secara langsung

Event Handler dapat dijalankan dengan cara membuat kode javascript secara langsung di text editor. Sintaknya adalah :

OnNamaEvent = "kode javascript"

Perhatikan contoh_9.2.html :

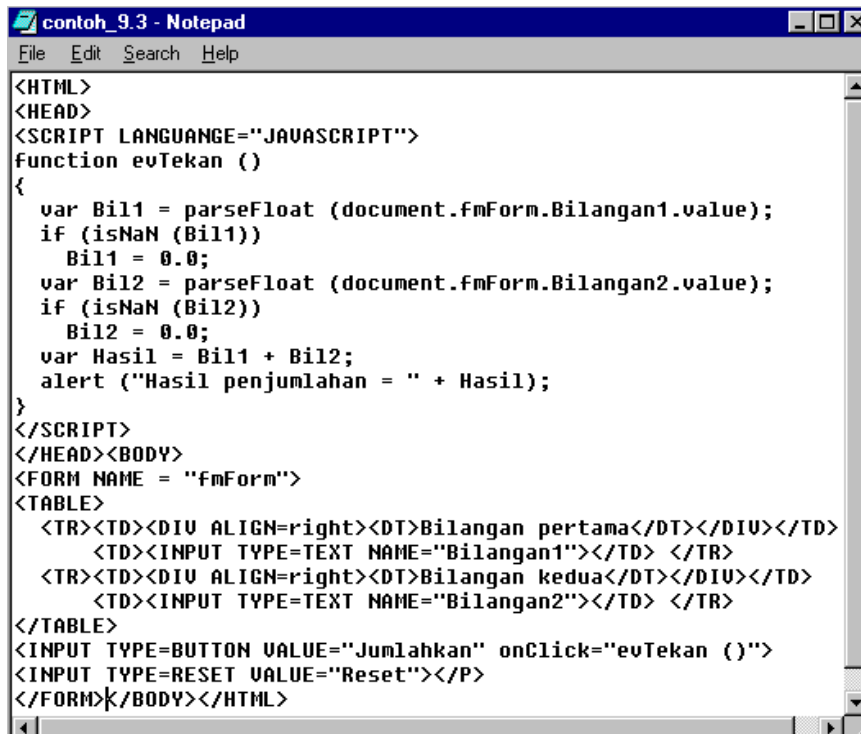


```
<html>
<head>
</head>
<body>
<center><h4>metode pemanggilan event Handler dengan
kode javascript langsung </h4>
<a href="www.firdausadnan.com" onClick="document.write
('<h4>'+ 'selamat datang'+ '</h4>')">pencet</a>
</body>
</html>
```

Dari contoh gambar di atas dapat di tarik beberapa kesimpulan, bahwa :
Jika link tersebut di klik yaitu kata *pencet*, maka akan dijalankan script javascript “document.write ('<h4>'+ 'selamat datang'+ '</h4>')” yang akan menampilkan teks selamat datang pada halaman browser tersebut.

Beberapa aplikasi Event

1. contoh_9.3.html mengenai event Click:



```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JAVASCRIPT">
function evTekan ()
{
  var Bil1 = parseFloat (document.fmForm.Bilangan1.value);
  if (isNaN (Bil1))
    Bil1 = 0.0;
  var Bil2 = parseFloat (document.fmForm.Bilangan2.value);
  if (isNaN (Bil2))
    Bil2 = 0.0;
  var Hasil = Bil1 + Bil2;
  alert ("Hasil penjumlahan = " + Hasil);
}
</SCRIPT>
</HEAD><BODY>
<FORM NAME = "fmForm">
<TABLE>
  <TR><TD><DIV ALIGN=right><DT>Bilangan pertama</DT></DIV></TD>
    <TD><INPUT TYPE=TEXT NAME="Bilangan1"></TD> </TR>
  <TR><TD><DIV ALIGN=right><DT>Bilangan kedua</DT></DIV></TD>
    <TD><INPUT TYPE=TEXT NAME="Bilangan2"></TD> </TR>
</TABLE>
<INPUT TYPE=BUTTON VALUE="Jumlahkan" onClick="evTekan ()">
<INPUT TYPE=RESET VALUE="Reset"></P>
</FORM></BODY></HTML>
```

Contoh di atas memberikan gambaran kerja sebuah perhitungan sederhana dengan menggunakan javascript yang menjadi gabungan antara fungsi dan event.

Di bawah ini adalah contoh aplikasi form yang interaktif dengan user ketika memasukkan input ke dalamnya :

Perhatikan contoh_9.4.html, contoh ini merupakan contoh event focus :

```
contoh_9.4 - Notepad
File Edit Search Help
<HTML>
<HEAD>
<TITLE>LATIHAN event focus </TITLE>
</HEAD>
<BODY>
<H2>ruang tamu</H2>
<B> Tolong isi buku tamu di bawah ini </B>
<FORM METHOD="POST" ACTION=" kirim.html">
<PRE>
Nama           :<INPUT TYPE=TEXT NAME=
"Nama" onFocus="window.status = 'Ketikkan Nama Anda !'">
<BR>Alamat      :<INPUT TYPE=TEXT NAME=
"Alamat" SIZE=40 MAXLENGTH=60 onFocus="window.status =
'Ketikkan Alamat Anda ! ' "> <BR>
E-mail       :<INPUT TYPE=TEXT NAME=
"Email" SIZE=20 onFocus="window.staus =
'Ketikkan Email Anda ! ' "> <BR>
Telepon      :<INPUT TYPE=TEXT NAME="Telepon" SIZE=10
onFocus="window.status = 'Teleponnya dong ! ' "> <BR>
<INPUT TYPE=SUBMIT VALUE="OK">
<INPUT TYPE=RESET VALUE="BATAL">
</PRE>
</FORM>
</BODY>
</HTML>
```

Tampilan di browsernya adalah :



Perhatikan bagian yang dikenai oleh tanda panah di atas, bagian tersebut merupakan ciri khas dari event focus.

Contoh_9.5.html tentang event Submit adalah :

```
contoh_9.5 - Notepad
File Edit Search Help
<HTML>
<HEAD>
<TITLE>LATIHAN event submit</TITLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
function fungsi1 ( )
{
window.alert ("Form telah berhasil dikirimkan !");
}
</SCRIPT>
</HEAD>
<H1>Contoh onSubmit</H1>
<FORM onSubmit="fungsi1( )">
NAMA : <INPUT TYPE = TEXT NAME ="nama">
<BR><BR>
<INPUT TYPE = SUBMIT NAME = " send" VALUE= "kirim ">
</FORM>
</BODY>
</HTML>
```

Tampilan di browsernya adalah :

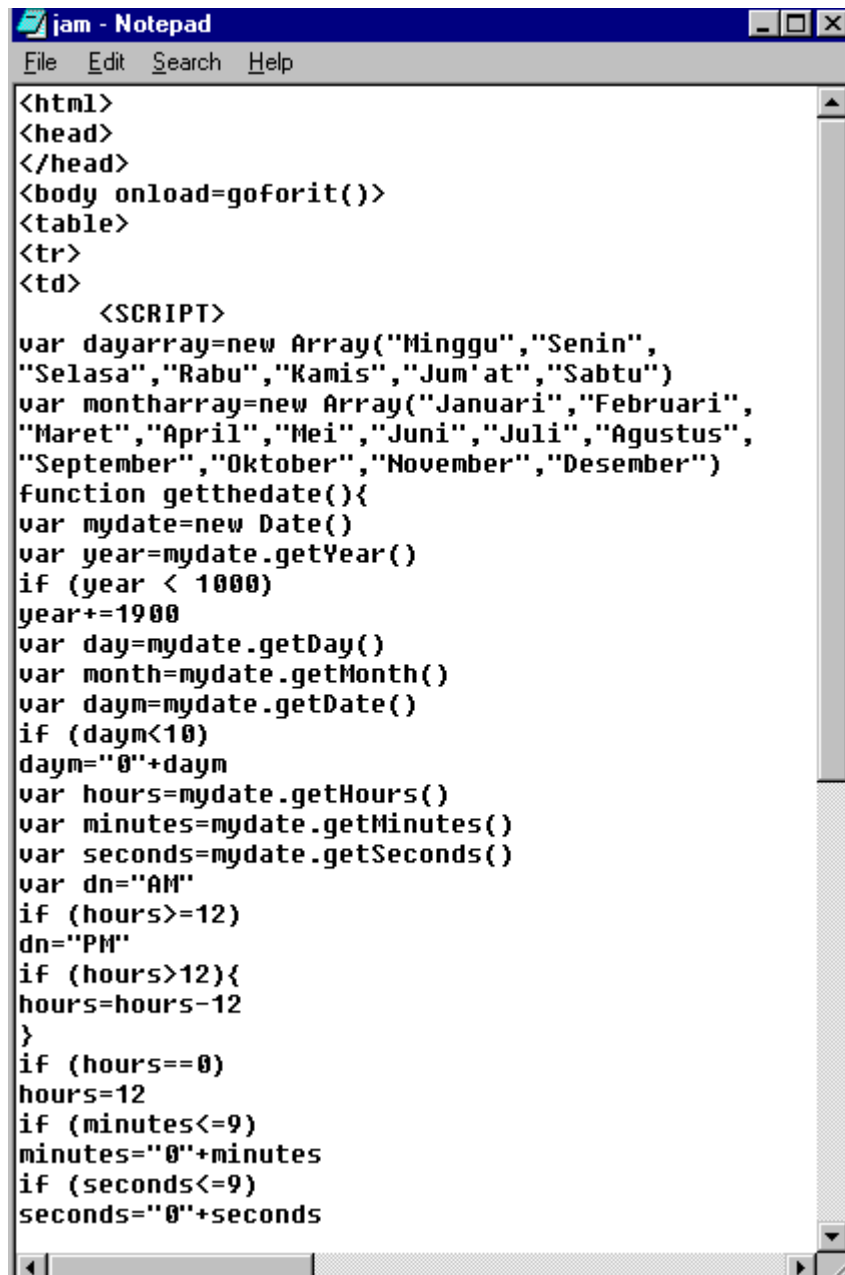


Tampilan yang dikenai oleh tanda panah, merupakan cirri khas dari event onSubmit.

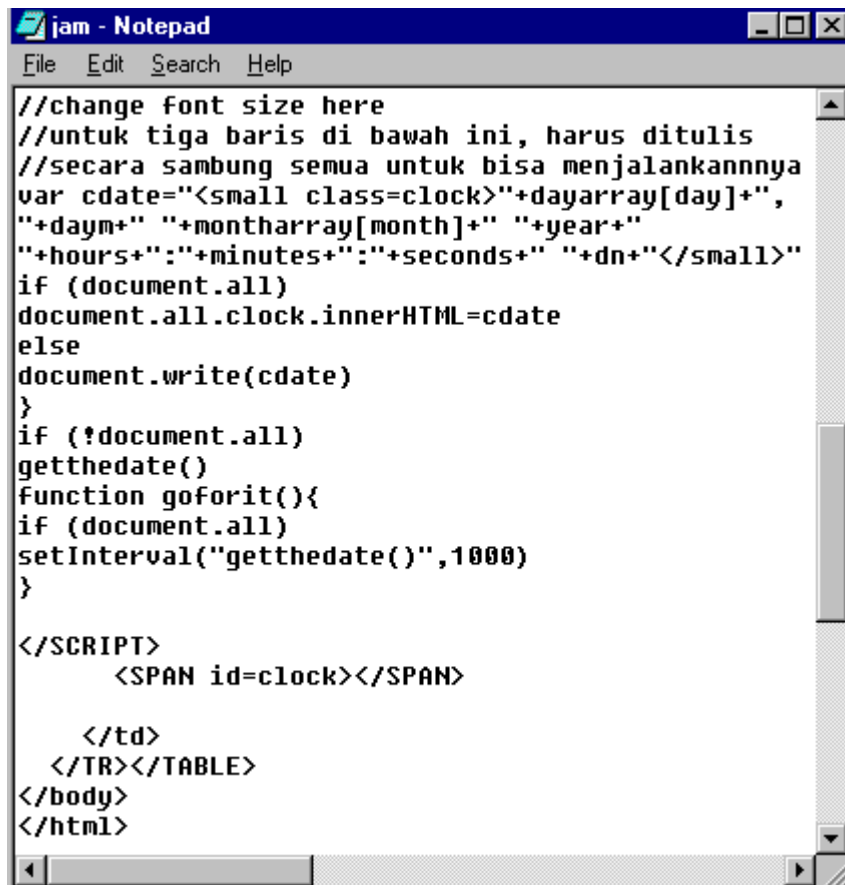
ANEKA APLIKASI JAVASCRIPT

Di bawah ini akan di sajikan beberapa aplikasi javascript yang dinamis sehingga dapat membuat tampilan web site kita menjadi lebih indah dan interaktif.

1. Aplikasi Jam



```
<html>
<head>
</head>
<body onload=goforit()>
<table>
<tr>
<td>
    <SCRIPT>
var dayarray=new Array("Minggu","Senin",
"Selasa","Rabu","Kamis","Jum'at","Sabtu")
var montharray=new Array("Januari","Februari",
"Maret","April","Mei","Juni","Juli","Agustus",
"September","Oktober","November","Desember")
function getthedata(){
var mydate=new Date()
var year=mydate.getYear()
if (year < 1000)
year+=1900
var day=mydate.getDay()
var month=mydate.getMonth()
var daym=mydate.getDate()
if (daym<10)
daym="0"+daym
var hours=mydate.getHours()
var minutes=mydate.getMinutes()
var seconds=mydate.getSeconds()
var dn="AM"
if (hours>=12)
dn="PM"
if (hours>12){
hours=hours-12
}
if (hours==0)
hours=12
if (minutes<=9)
minutes="0"+minutes
if (seconds<=9)
seconds="0"+seconds
```



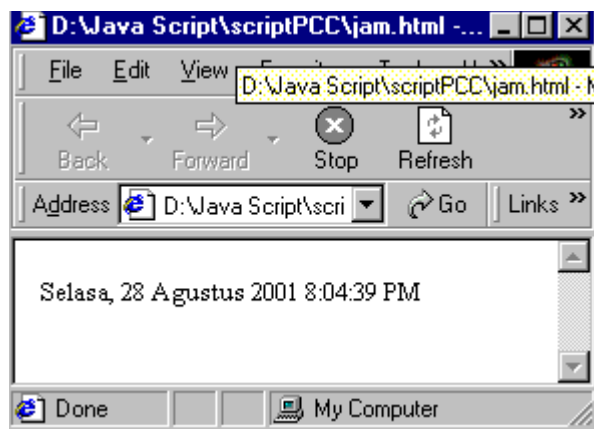
```
//change font size here
//untuk tiga baris di bawah ini, harus ditulis
//secara sambung semua untuk bisa menjalankannya
var cdate=""<small class=clock>"+dayarray[day]+",
"+daym+" "+montharray[month]+" "+year+"
"+hours+": "+minutes+": "+seconds+" "+dn+"</small>"
if (document.all)
document.all.clock.innerHTML=cdate
else
document.write(cdate)
}
if (!document.all)
getthedata()
function goforit(){
if (document.all)
setInterval("getthedata()",1000)
}

</SCRIPT>
    <SPAN id=clock></SPAN>

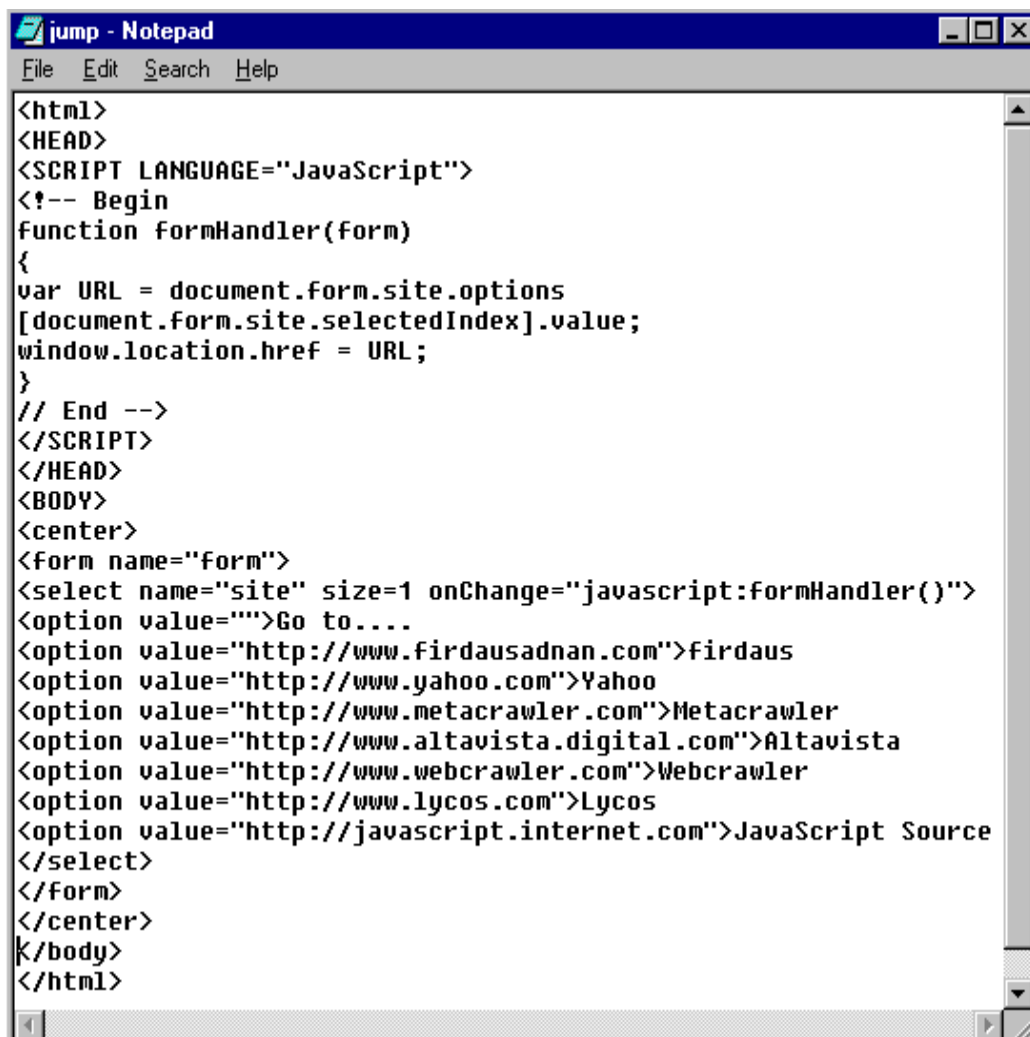
    </td>
  </TR></TABLE>
</body>
</html>
```

Dari script di atas, akan membuat suatu tampilan jam yang dinamis sesuai dengan pergerakan jam yang ada di komputer kita sendiri.

Tampilan di brosernya adalah :

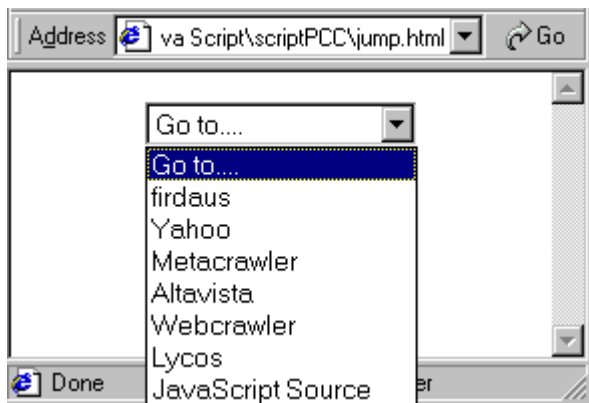


2 Di bawah ini merupakan contoh link ke beberapa web site terkenal di dunia. Dengan memilih option-nya, kita sudah langsung membuka website tersebut tanpa harus merubah bentuk mouse menjadi seperti tangan. Perhatikan contoh di bawah ini



```
<html>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- Begin
function formHandler(form)
{
var URL = document.form.site.options
[document.form.site.selectedIndex].value;
window.location.href = URL;
}
// End -->
</SCRIPT>
</HEAD>
<BODY>
<center>
<form name="form">
<select name="site" size=1 onChange="javascript:formHandler()">
<option value="">Go to....
<option value="http://www.firdausadnan.com">firdaus
<option value="http://www.yahoo.com">Yahoo
<option value="http://www.metacrawler.com">Metacrawler
<option value="http://www.altavista.digital.com">Altavista
<option value="http://www.webcrawler.com">Webcrawler
<option value="http://www.lycos.com">Lycos
<option value="http://javascript.internet.com">JavaScript Source
</select>
</form>
</center>
</body>
</html>
```

Tampilannya di browser adalah :



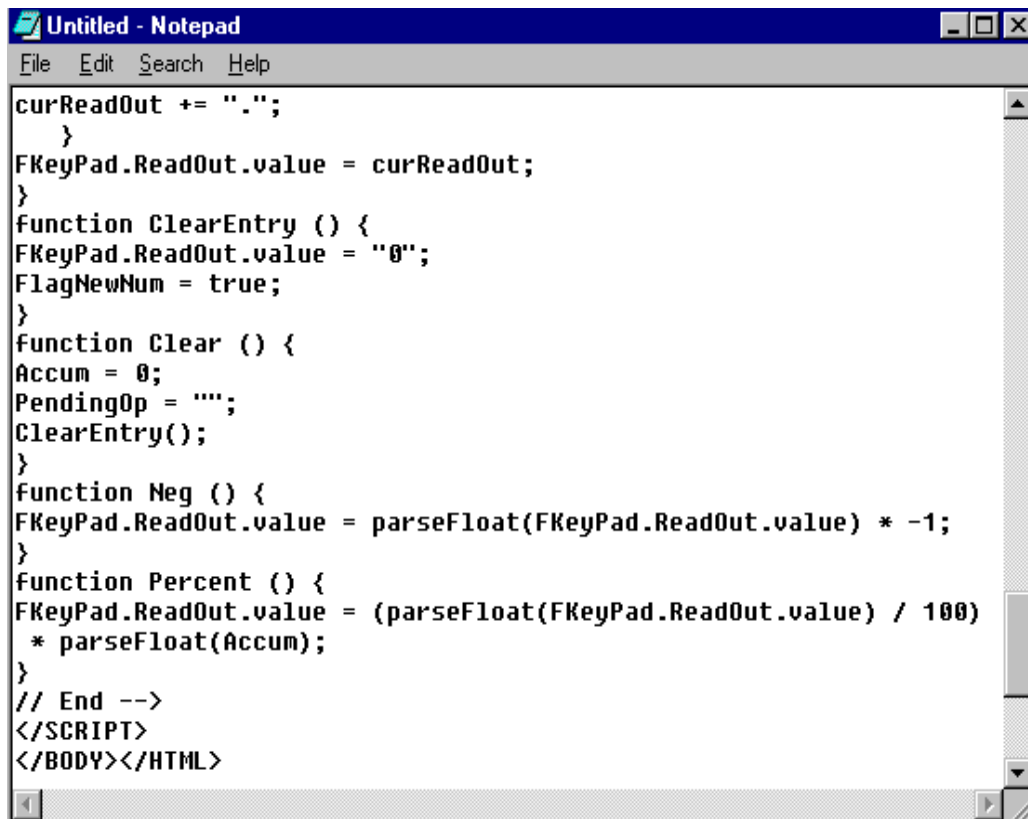
3. Manipulasi kalkulator

```
Untitled - Notepad
File Edit Search Help
<HTML><HEAD><TITLE>kalkulator</TITLE>
</HEAD>
<BODY bgColor=#ffffff>
<CENTER>
<FORM action="" name=Keypad>
<!-- -->
<TABLE>
  <TBODY></TBODY></TABLE>
<TABLE border=2 cellPadding=1 cellSpacing=5 height=60 width=50>
  <TBODY> <TR> <TD align=middle colSpan=3>
<INPUT name=ReadOut size=24 value=0 width="100%"> </TD>
<TD><INPUT name=btnClear onclick=Clear() type=button value="C">
</TD><TD><INPUT name=btnClearEntry onclick=ClearEntry()
type=button value=" CE ">
</TD></TR> <TR>
  <TD><INPUT name=btnSeven onclick=NumPressed(7)
type=button value=" 7 ">
</TD> <TD><INPUT name=btnEight onclick=NumPressed(8)
type=button value=" 8 ">
</TD> <TD><INPUT name=btnNine onclick=NumPressed(9)
type=button value=" 9 ">
</TD> <TD></TD>
<TD><INPUT name=btnNeg onclick=Neg()
type=button value=" +/- "> </TD>
<TD><INPUT name=btnPercent onclick=Percent()
type=button value=" % ">
</TD></TR> <TR> <TR>
  <TD><INPUT name=btnFour onclick=NumPressed(4)
type=button value=" 4 ">
```

```
Untitled - Notepad
File Edit Search Help

</TD> <TD><INPUT name=btnFive onclick=NumPressed(5)
  type=button value=" 5 ">
</TD> <TD><INPUT name=btnSix onclick=NumPressed(6)
  type=button value=" 6 ">
</TD> <TD></TD>
  <TD align=middle><INPUT name=btnPlus
    onclick="Operation('+')" type=button value=" + ">
  </TD> <TD align=middle><INPUT name=btnMinus
    onclick="Operation('-')" type=button value=" - ">
</TD></TR> <TR>
  <TD><INPUT name=btnOne onclick=NumPressed(1)
    type=button value=" 1 ">
</TD> <TD><INPUT name=btnTwo onclick=NumPressed(2)
    type=button value=" 2 ">
</TD> <TD><INPUT name=btnThree onclick=NumPressed(3)
    type=button value=" 3 ">
  </TD> <TD></TD>
  <TD align=middle><INPUT name=btnMultiply
    onclick="Operation('*')" type=button value=" * ">
  </TD> <TD align=middle><INPUT name=btnDivide
    onclick="Operation('/')" type=button value=" / ">
</TD></TR> <TR>
  <TD><INPUT name=btnZero onclick=NumPressed(0)
    type=button value=" 0 ">
  </TD> <TD><INPUT name=btnDecimal onclick=Decimal()
    type=button value=" . ">
</TD> <TD colspan=3></TD>
  <TD><INPUT name=btnEquals
    onclick="Operation('=')" type=button value=" = ">
  </TD></TR></TBODY></TABLE></TABLE></FORM></CENTER>
<SCRIPT language=JavaScript>
<!-- Begin
var FKeyPad = document.Keypad;
var Accum = 0;
var FlagNewNum = false;
var PendingOp = "";
```

```
function NumPressed (Num) {
if (FlagNewNum) {
FKeyPad.ReadOut.value = Num;
FlagNewNum = false;
} else {
if (FKeyPad.ReadOut.value == "0")
FKeyPad.ReadOut.value = Num;
else
FKeyPad.ReadOut.value += Num;//batas
}
}function Operation (Op) {
var Readout = FKeyPad.ReadOut.value;
if (FlagNewNum && PendingOp != "=");
else
{FlagNewNum = true;
if ( '+' == PendingOp )
Accum += parseFloat(Readout);
else if ( '-' == PendingOp )
Accum -= parseFloat(Readout);
else if ( '/' == PendingOp )
Accum /= parseFloat(Readout);
else if ( '*' == PendingOp )
Accum *= parseFloat(Readout);
else
Accum = parseFloat(Readout);
FKeyPad.ReadOut.value = Accum;
PendingOp = Op;
}
}
function Decimal () {
var curReadOut = FKeyPad.ReadOut.value;
if (FlagNewNum) {
curReadOut = "0.";
FlagNewNum = false;
} else {
if (curReadOut.indexOf(".") == -1)
```

```
curReadOut += ".";
}
FKeyPad.ReadOut.value = curReadOut;
}
function ClearEntry () {
FKeyPad.ReadOut.value = "0";
FlagNewNum = true;
}
function Clear () {
Accum = 0;
PendingOp = "";
ClearEntry();
}
function Neg () {
FKeyPad.ReadOut.value = parseFloat(FKeyPad.ReadOut.value) * -1;
}
function Percent () {
FKeyPad.ReadOut.value = (parseFloat(FKeyPad.ReadOut.value) / 100)
* parseFloat(Accum);
}
// End -->
</SCRIPT>
</BODY></HTML>
```