

DASAR PEMROGRAMAN JAVASCRIPT

A. Mengenal Javascript

1. Apa itu javascript

Javascript merupakan bahasa pemrograman web yang bersifat *Client Side Programming*, artinya pemrosesan dilakukan di sisi klien. Berbeda dengan PHP yang diproses di sisi server dimana browser akan melakukan loading ketika program sedang diproses di server. Sedangkan pada pemrosesan javascript, browser tidak perlu melakukan refresh atau loading.

Karena sifatnya yang diproses di sisi klien maka respon dari jauh lebih cepat dibanding pemrograman yang diproses di server. Selain itu, javascript juga dapat digunakan untuk membuat animasi teks maupun gambar. Namun karena sifatnya yang diproses di sisi klien, skrip javascript dapat dilihat oleh pengunjung website dengan membuka *source code* pada browser. Selain itu, hampir semua browser memiliki fitur untuk mematikan javascript bahkan memanipulasi kode javascript. Oleh karena itu, kita tidak bisa sepenuhnya bergantung dengan javascript.

2. Kegunaan javascript

Pada awal perkembangannya, javascript digunakan untuk membuat interaksi antara user dan website menjadi lebih cepat seperti pada validasi form yang lebih cepat dibanding menggunakan PHP yang harus menunggu beberapa saat untuk melihat hasil validasi. Namun dalam perkembangannya, javascript digunakan untuk berbagai keperluan yang lebih modern seperti animasi elemen web, fitur chatting, game online, dan sebagainya.

3. Menjalankan Javascript

Javascript merupakan bahasa script yang berjalan pada web browser, sehingga untuk menuliskan skrip javascript cukup menggunakan text editor sedangkan untuk menjalankan hasilnya cukup dengan web browser. Untuk aplikasi text editor kita dapat menggunakan aplikasi bawaan Windows yaitu Notepad. Namun, akan lebih mudah jika menggunakan aplikasi khusus seperti **notepad++** yang cukup ringan untuk digunakan. Sedangkan untuk web browser, kita dapat menggunakan web browser apa saja, karena sebagian besar web browser dapat menjalankan skrip javascript.

B. Meletakkan Kode Javascript

Ada berbagai cara meletakkan kode javascript pada dokumen HTML. Anda dapat menggunakan cara yang paling sesuai dengan selera Anda. Namun, biasanya disesuaikan dengan kebutuhan atau keadaan. Berikut berbagai cara meletakkan kode Javascript.

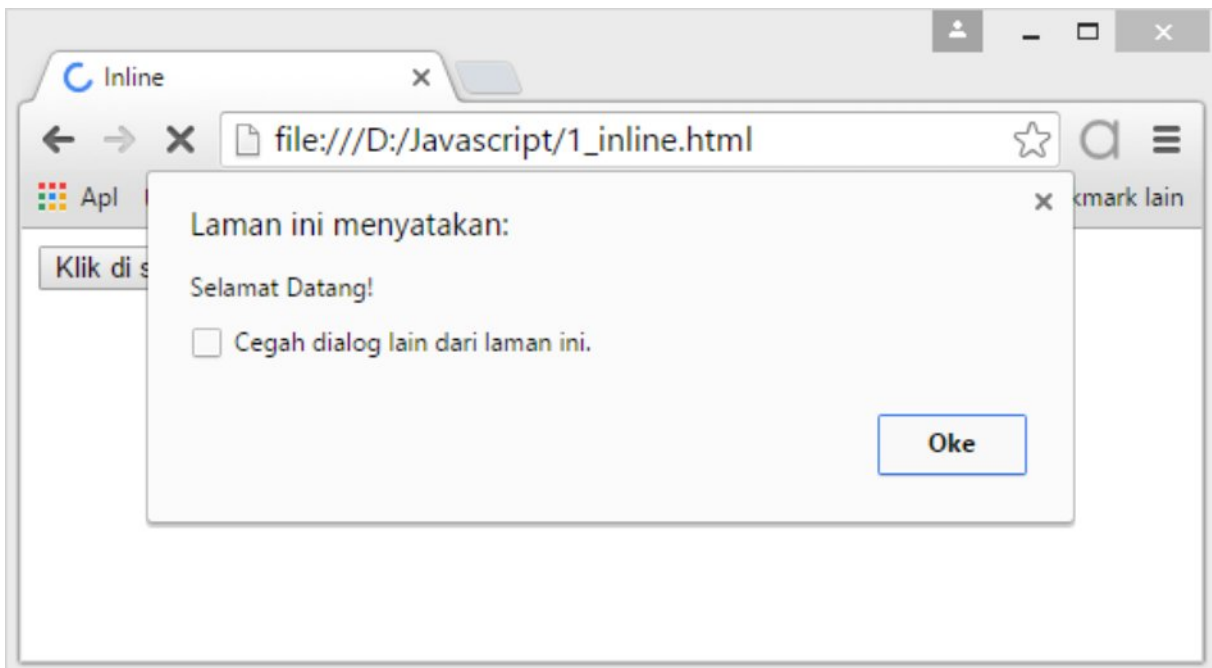
1. Menggunakan Event Handler (Inline)

Cara ini biasanya digunakan jika kode javascript hanya sedikit atau sekedar memanggil fungsi. Cara meletakkan kode javascript melalui cara ini yaitu dengan meletakkan kode setelah atribut event pada tag HTML. Pada contoh skrip berikut, kita akan menampilkan kotak pesan ketika dokumen baru dibuka dan ketika klik tombol.

```

1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Inline</title>
5   </head>
6   <body onload="alert('Selamat Datang!')">
7     <button onclick="alert('Anda menekan tombol')">
      Klik di sini</button>
8   </body>
9 </html>

```



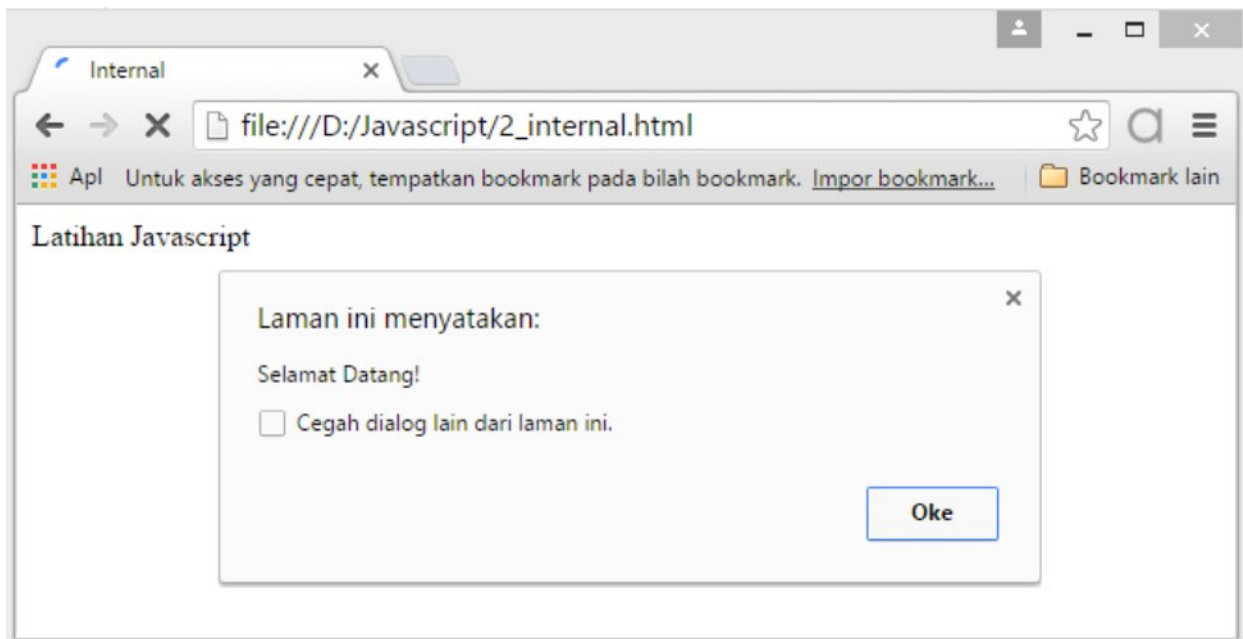
2. Menggunakan tag <script> (internal)

Caranya yaitu dengan meletakkan kode javascript di dalam tag <script>...</script>. Tag ini dapat diletakkan di dalam tag <head> maupun <body>. Cara ini biasanya digunakan jika kode javascript terdiri dari beberapa baris tetapi tidak terlalu panjang. Kita dapat menggunakan atribut **type="text/javascript"** untuk membedakan dengan skrip dari bahasa lain tetapi tidak wajib.

```

1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Internal</title>
5     <script type="text/javascript">
6       alert("Selamat Datang!");
7     </script>
8   </head>
9   <body>
10    <script>
11      document.write("Latihan Javascript");
12    </script>

```



3. Menggunakan tag `<script src="">` (eksternal)

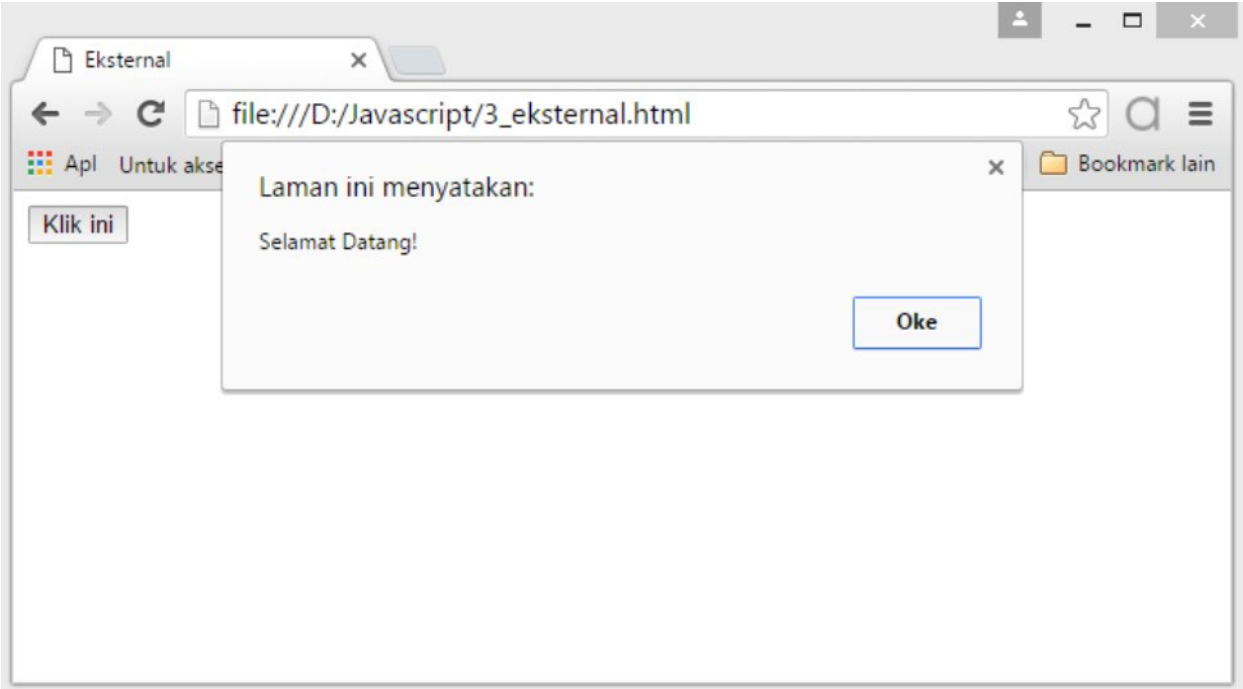
Caranya yaitu dengan memisahkan kode javascript dengan kode HTML. Kode javascript disimpan dengan file tersendiri dengan format `.js`. File tersebut kemudian dipanggil dengan tag `<script src=""></script>` pada bagian `<head>`. Cara ini biasanya digunakan jika kode javascript sangat panjang. Pada skrip berikut kita simpan kode javascript pada file **skrip.js** dan dipanggil melalui file **3_eksternal.html**.

File skrip.js

```
1 function hallo() {
2     alert("Selamat Datang!");
3 }
```

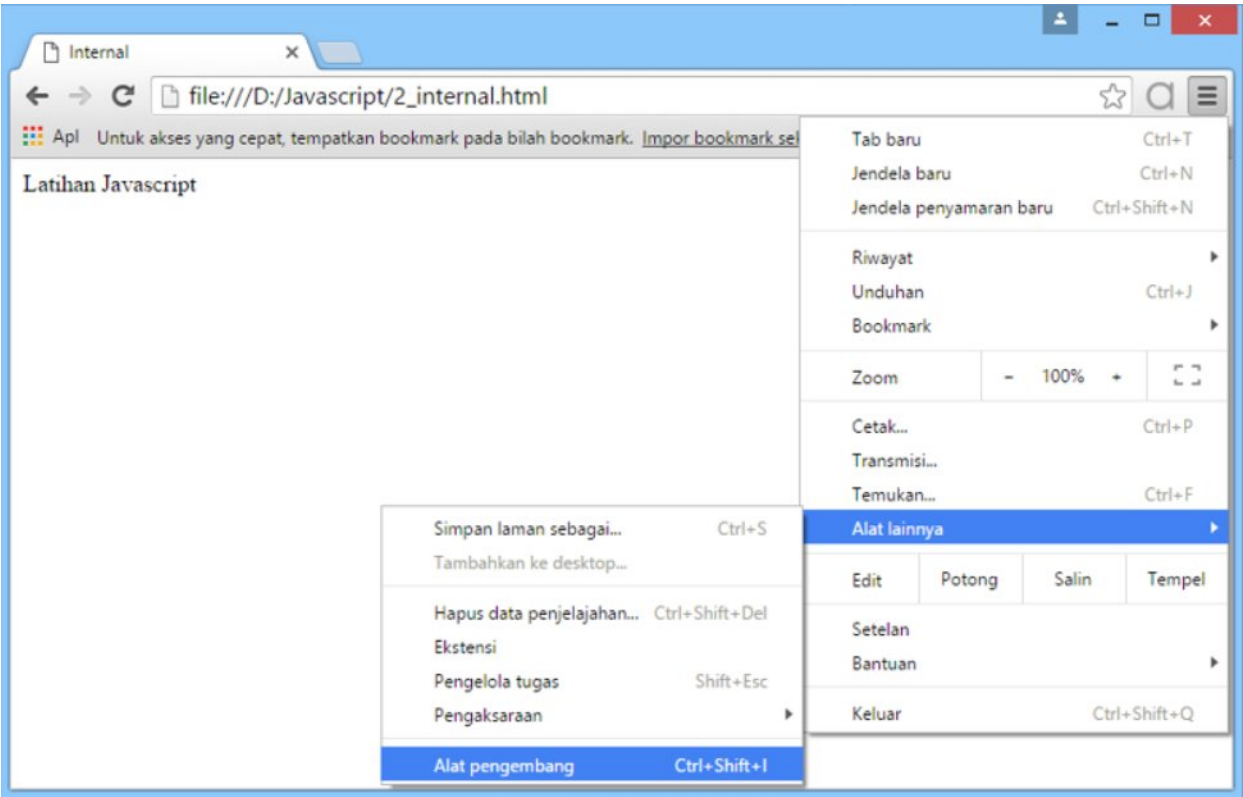
File 3_eksternal.html

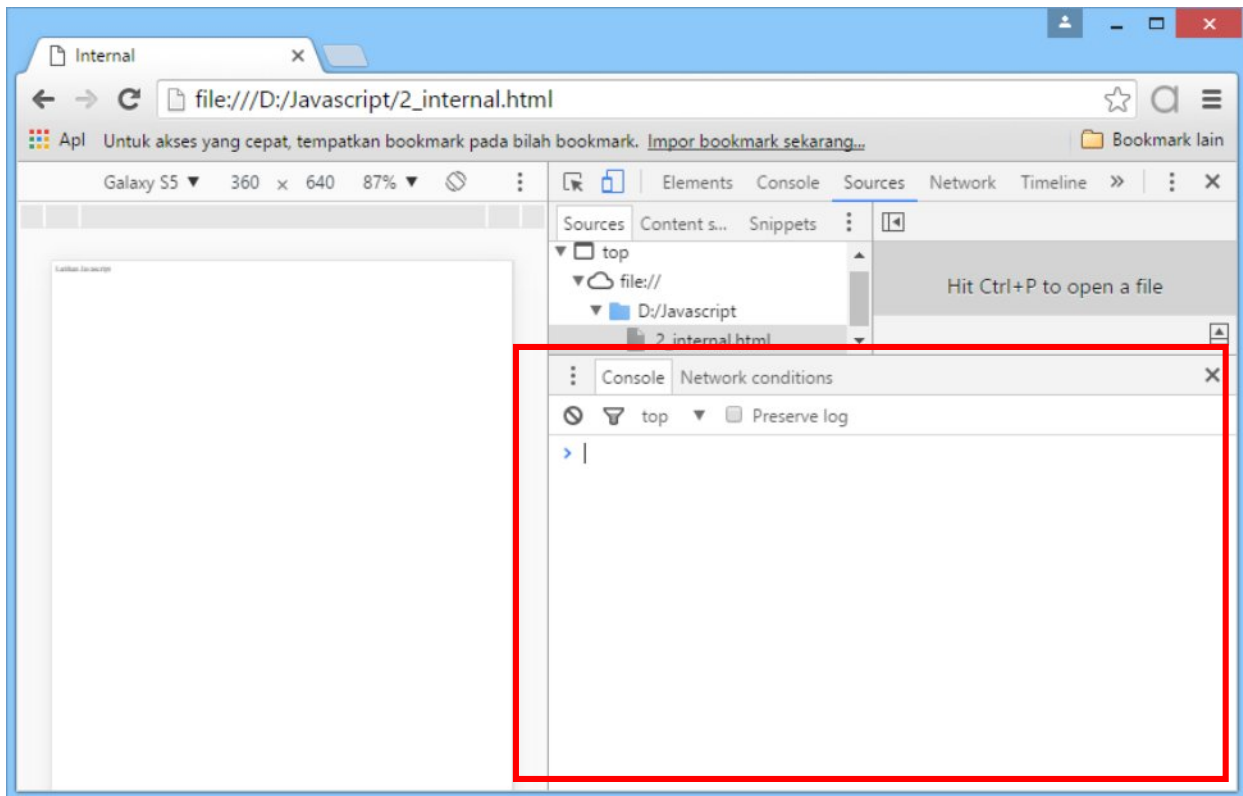
```
1 <!DOCTYPE HTML>
2 <html>
3     <head>
4         <title>Eksternal</title>
5         <script src="skrip.js"></script>
6     </head>
7     <body>
8         <button onclick="hallo()">Klik ini</button>
9     </body>
10 </html>
```



C. Mengatasi Error pada Javascript

Dalam menuliskan program pasti sering mengalami error pada program. Error pada javascript tidak akan menampilkan pesan apapun pada browser seperti PHP, jadi akan sedikit menyulitkan bagi programmer. Tapi, tidak perlu khawatir, karena beberapa browser menyediakan tool untuk mengatasi error pada javascript. Seperti pada browser Google Chrome disediakan menu Developer Tool. Biasanya pesan error pada program javascript akan tampil pada tab Console.





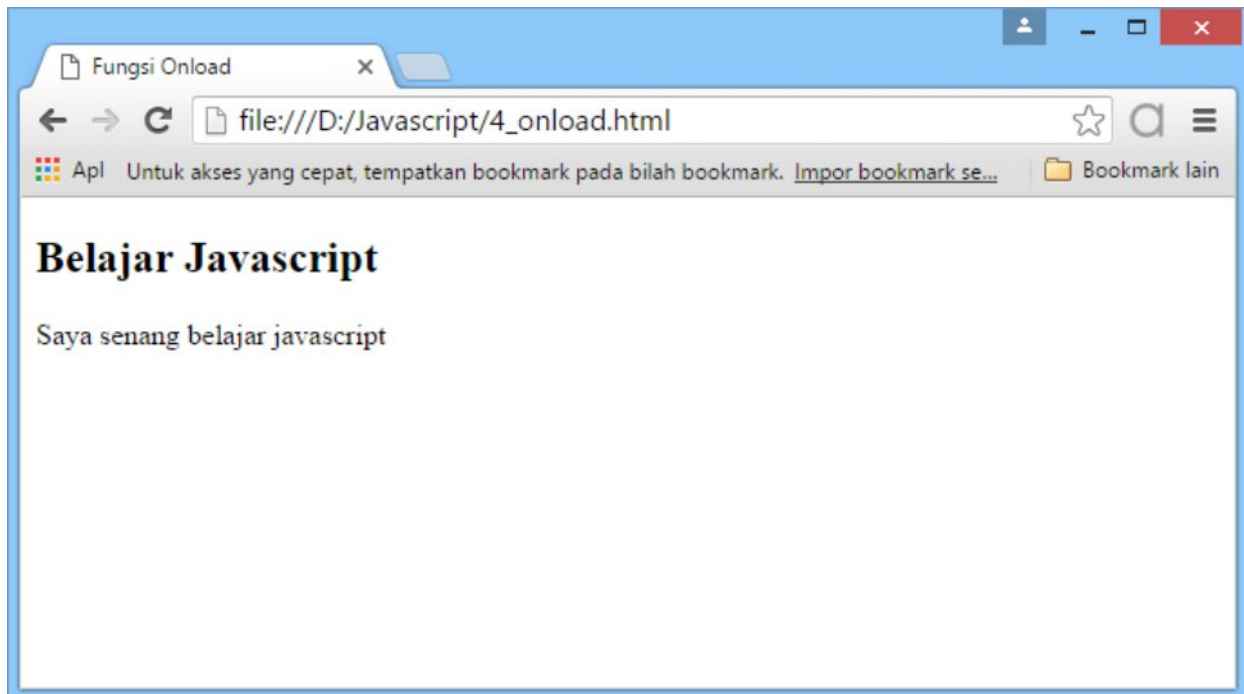
D. Fungsi Event Onload

Browser akan mengeksekusi skrip HTML dari atas ke bawah. Jadi, ketika kode javascript di letakkan di dalam tag `<head>`, kode javascript akan dieksekusi terlebih dahulu sebelum skrip HTML yang ada di dalam tag `<body>`. Padahal, penggunaan javascript kebanyakan untuk memanipulasi skrip HTML. Permasalahan terjadi ketika kode javascript dieksekusi, skrip HTML yang dimanipulasi belum tersedia (belum dieksekusi). Solusi untuk mengatasi hal ini dengan menggunakan fungsi event onload. Ada dua cara menggunakannya, yaitu dengan membuat skrip **window.onload** pada kode javascript dan dengan menambah atribut **onload** pada tag `<body>`.

```

1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <title>Fungsi Onload</title>
5      <script>
6        window.onload = function() {
7          var teks = document.getElementById('judul');
8          teks.innerHTML = "Belajar Javascript";
9        }
10
11       function kata() {
12         var kata = document.getElementById('kata');
13         kata.innerHTML = "Saya senang belajar javascript";
14       }
15     </script>
16   </head>
17   <body onload="kata()">
18     <h2 id="judul"></h2>
19     <p id="kata"></p>
20   </body>

```



E. Tipe Data Javascript

Setiap bahasa pemrograman pasti mengenal yang namanya tipe data. Tipe data pada javascript sedikit berbeda dengan bahasa pemrograman lain. Tipe data pada javascript bersifat seolah-olah sebagai objek. Sehingga setiap variabel pada javascript akan langsung memiliki method.

Javascript mengenal 3 tipe data dasar yaitu sebagai berikut:

- **Numbers** : contoh 125, 20.50, dsb.
- **Strings** : contoh "Hallo!", "200", dsb.
- **Boolean** : hanya ada dua nilai yaitu True dan False.

Selain tiga tipe di atas, javascript juga mengenal tipe data **null** dan **undefined**. **Null** merupakan nilai dari sebuah variabel yang tidak diberi nilai saat didefinisikan. Sedangkan **undefined** merupakan hasil yang didapat dari proses berikut:

- Nilai dari pemanggilan variabel yang belum didefinisikan
- Nilai dari pemanggilan elemen array yang tidak ada
- Nilai dari pemanggilan property objek yang tidak ada
- Nilai dari pemanggilan fungsi yang tidak mengembalikan nilai
- Nilai dari parameter fungsi yang tidak memiliki argumen

Dapat disimpulkan bahwa perbedaan antara null dan undefined adalah bahwa null biasanya diperoleh dari kondisi normal yang sudah direncanakan, sedangkan undefined biasanya diperoleh dari kesalahan program yang tidak direncanakan.

F. Variabel

1. Pengertian variabel dan cara mendeklarasikanya


```

    var panjang;
    var tinggi;
</script>

```

Selain dengan cara di atas, kita juga dapat mendeklarasikan beberapa variabel sekaligus dengan satu **var**.

ubah saat program dijalankan. Variabel pada javascript didefinisikan dengan **var**. Contohnya sebagai berikut:

```

<script>
    var panjang, tinggi;
</script>

```

Kita juga dapat langsung mendefinisikan suatu nilai awal pada saat pendeklarasian variabel.

```

<script>
    var panjang = 200;
    var tinggi = 150;
    var kata = "Javascript menyenangkan";
</script>

```

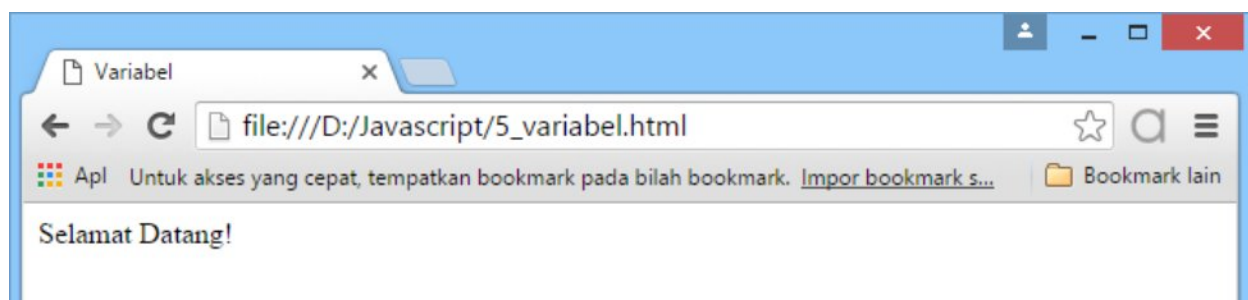
2. Scope Variabel

Yang dimaksud dengan scope variabel adalah batasan penggunaan variabel. Javascript mengenal variabel **global** dan variabel **local**. Variabel **global** merupakan variabel yang dapat dipakai dimana saja baik di luar fungsi maupun di dalam fungsi. Sedangkan variabel **local** yaitu variabel yang hanya dapat digunakan di dalam fungsi. Variabel yang dideklarasikan di luar fungsi akan menjadi variabel **global** dan variabel yang dideklarasikan di dalam fungsi akan menjadi variabel **local**.

```

1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <title>Variabel</title>
5  </head>
6  <body onload="kata()">
7      <script>
8          var a = "Selamat Datang!";
9          function selamat() {
10              var b = "Selamat pagi!";
11              document.write(a);
12          }
13          selamat();
14          document.write(b);
15      </script>
16  </body>
17 </html>

```



3. Aturan penamaan variabel

Pada gambar di atas, variabel **a** berhasil ditampilkan walaupun dideklarasikan di luar fungsi dan digunakan di dalam fungsi. Ini namanya variabel **global**. Sedangkan variabel **b** tidak tampil karena dideklarasikan di dalam fungsi dan digunakan di luar fungsi. Ini namanya variabel **local**.

Dalam penulisan variabel, ada aturan yang harus dipatuhi agar tidak menghasilkan error pada program. Dalam javascript, aturan penulisan variabel sebagai berikut:

- Jangan menggunakan kata-kata yang menjadi kode javascript atau dikenal *reserved keyword*. Contohnya: break, boolean, else, and, var, true, enum, switch, this, throw, long, new, dan sebagainya.
- Variabel tidak boleh diawali dengan angka atau simbol. Variabel harus diawali dengan huruf atau underscore. Contoh: panjang, _tinggi, dan sebagainya. Sedangkan yang tidak dibolehkan seperti: 7inggi, 4las, @las, dan sebagainya.
- Variabel dalam javacript bersifat **case-sensitive** artinya huruf kecil dan huruf besar dibedakan. Jadi, antara **tinggi** dan **Tinggi** adalah dua variabel yang berbeda.

G. Operator

Operator merupakan sebuah simbol atau kata yang digunakan untuk melakukan operasi terhadap satu atau lebih data atau variabel sehingga menghasilkan data baru. Variabel atau data yang dioperasikan disebut dengan **operand**. Javacript mengenal beberapa jenis operator yaitu sebagai berikut:

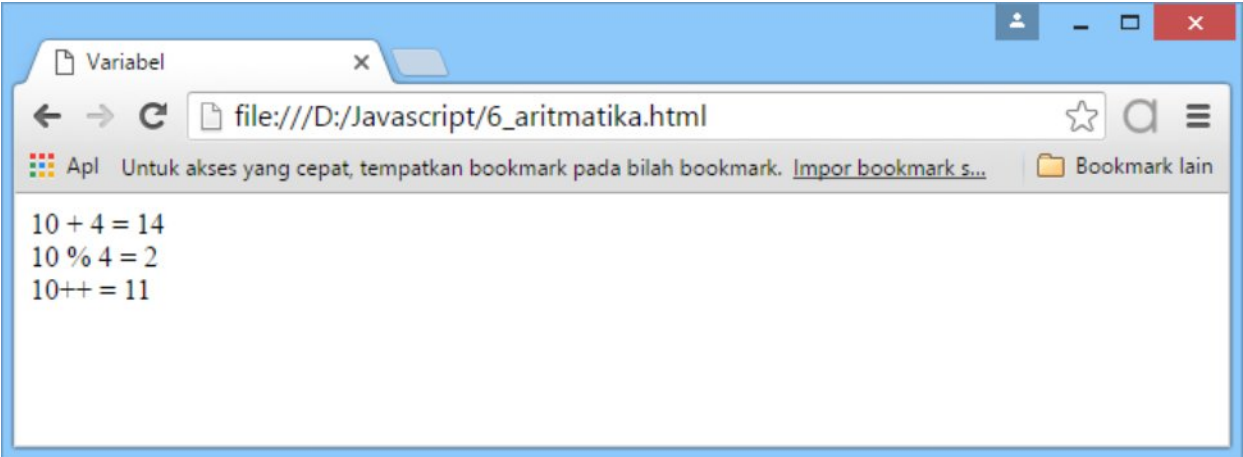
1. Operator Aritmatika

Operator aritmatika merupakan operator yang digunakan untuk melakukan perhitungan data numerik. Daftar operator aritmatika dan cara penggunaanya dapat dilihat pada tabel berikut:

Operator	Keterangan
+	Untuk penjumlahan, misal A = 10 dan B = 4, maka: A + B menghasilkan 14
-	Untuk pengurangan, misal A = 10 dan B = 4, maka: A - B menghasilkan 6
*	Untuk perkalian, misal A = 10 dan B = 4, maka: A * B menghasilkan 40
/	Untuk pembagian, misal A = 10 dan B = 4, maka: A / B menghasilkan 2.5

++	Berarti ditambah satu, misal A = 10, maka: A++ menghasilkan 11
--	Berarti dikurangi satu, misal A = 10, maka: A-- menghasilkan 9

```
1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <title>Operator Aritmatika</title>
5  </head>
6  <body onload="kata()">
7      <script>
8          var A = 10, B = 4;
9          document.write("10 % 4 = ");
10         C = A % B;
11         document.write(C);
12         document.write("<br/>");
13
14         document.write("10++ = ");
15         A++;
16         document.write(A);
17         document.write("<br/>");
18     </script>
19 </body>
20 </html>
```

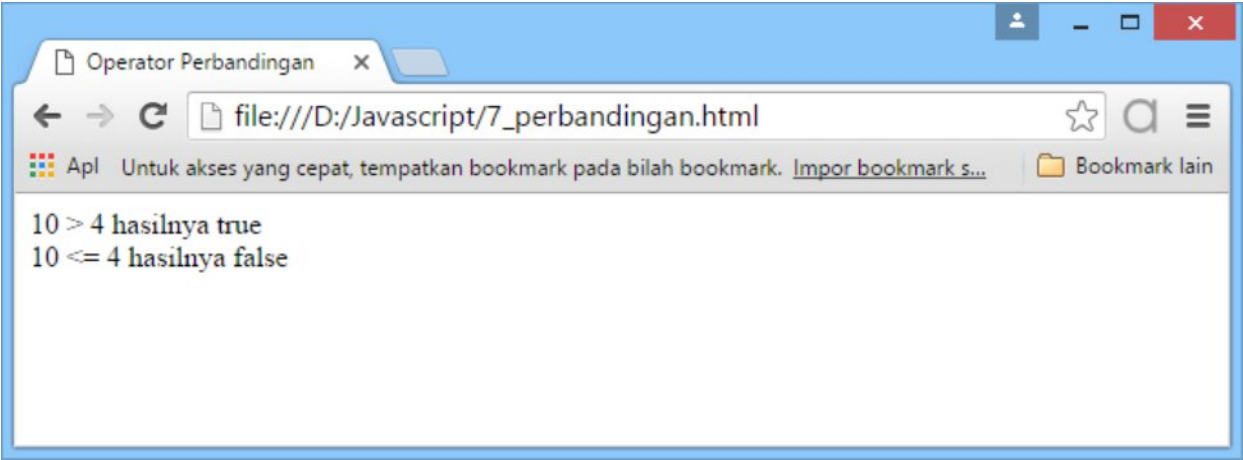


2. Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan dua variabel atau data. Hasil dari

==	Berarti sama dengan, misal A = 10 dan B = 4, maka: (A == B) menghasilkan false
!=	Berarti tidak sama dengan, misal A = 10 dan B = 4, maka: (A != B) menghasilkan true .
>	Berarti operand pertama lebih besar dari operand ke dua, misal A = 10 dan B = 4, maka: (A > B) menghasilkan true
<	Berarti operand pertama lebih kecil dari operand kedua, misal A = 10 dan B = 4, maka (A < B) menghasilkan false
>=	Berarti lebih besar atau sama dengan, misal A = 10 dan B = 4, maka: (A >= B) menghasilkan true dan (10 >= A) juga menghasilkan true
<=	Berarti lebih kecil atau sama dengan, misal A = 10 dan B = 4, maka: (A <= B) menghasilkan false , sedangkan (10 <= A) menghasilkan true

```
1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <title>Operator Perbandingan</title>
5    </head>
6    <body>
7      <script>
8        var A = 10, B = 4;
9        document.write("10 > 4 hasilnya ");
10       document.write(A > B);
11       document.write("<br/>");
12
13       document.write("10 <= 4 hasilnya ");
14       document.write(A <= B);
15       document.write("<br/>");
16     </script>
17   </body>
18 </html>
```

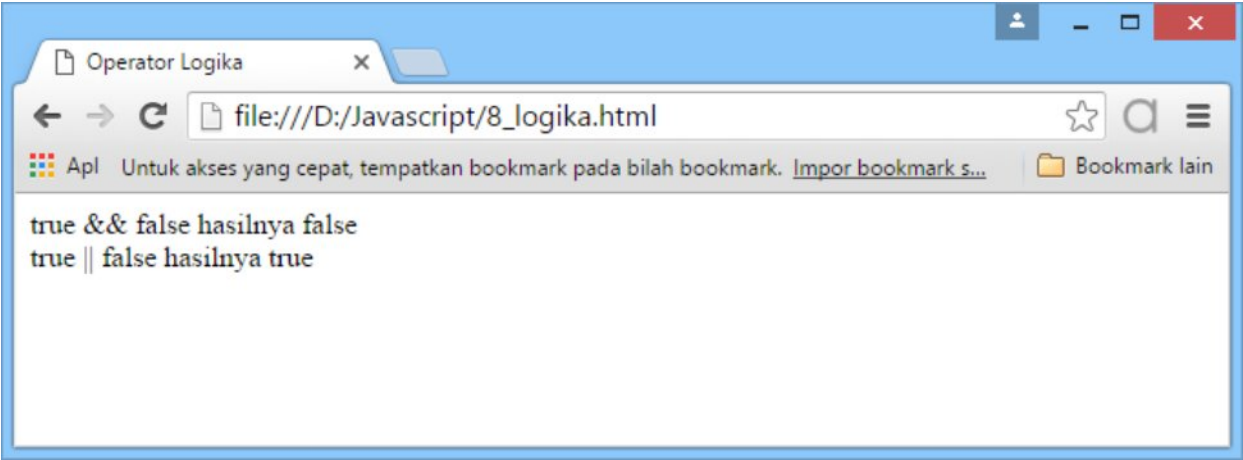


3. Operator Logika

Operator logika digunakan untuk mengoperasikan data boolean. Hasil dari perbandingan tersebut akan menghasilkan nilai **true** atau **false**. Daftar operator logika dan penjelasannya dapat dilihat pada table berikut:

Operator	Keterangan
&&	Menghasilkan nilai true jika kedua operand bernilai true. (logika And). Misal A = true dan B = false, maka: (A && B) menghasilkan false
	Menghasilkan nilai true jika salah satu atau kedua operand bernilai true. (logika Or). Misal A = true dan B = false, maka: (A B) menghasilkan true
!	Membalikan nilai sebuah variabel (logika Not). Misal A = true, maka: (!A) menghasilkan false

```
1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <title>Operator Logika</title>
5  </head>
6  <body>
7  <script>
8      var A = true, B = false;
9      document.write("true && false hasilnya ");
10     document.write(A && B);
11     document.write("<br/>");
12
13     document.write("true || false hasilnya ");
14     document.write(A || B);
```



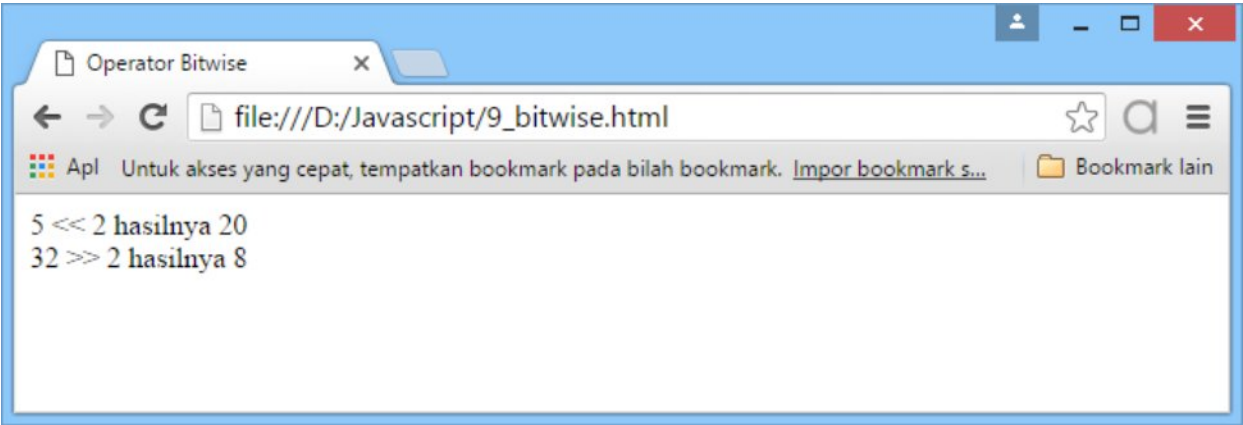
4. Operator Bitwise

Operator bitwise akan membandingkan nilai biner dua bilangan. Misalnya 2 akan dikonversi ke biner menjadi 10 dan 3 akan dikonversi ke biner menjadi 11. Jika 2 dan 3 dibandingkan dengan operator bitwise, maka yang dibandingkan adalah nilai biner-nya. Hasil dari operasi berupa biner yang akan dikonversi lagi dan ditampilkan dalam bentuk desimal. Daftar operator bitwise dan penjelasannya dapat dilihat pada table berikut:

Operator	Keterangan
&	<p>Menghasilkan angka 1 jika kedua angka yang dibandingkan 1 dan menghasilkan 0 jika salah satu atau kedua angka 0 (bitwise And). Misal, A = 2 (binernya 10) dan B = 3 (binernya 11), maka:</p> <p>(A & B) menghasilkan 2</p> <p>Maksudnya: digit pertama 1 & 1 menghasilkan 1, digit kedua 0 & 1 menghasilkan 0. Jika digabungkan hasilnya 10 jika didesimalkan menjadi 2.</p>
	<p>Menghasilkan angka 1 jika salah satu atau kedua angka yang dibandingkan 1 dan menghasilkan 0 jika kedua angka 0 (bitwise Or). Misal, A = 2 (binernya 10) dan B = 3 (binernya 11), maka:</p> <p>(A B) menghasilkan 3</p>
^	<p>Menghasilkan angka 1 jika salah satu angka yang dibandingkan 1 dan menghasilkan 0 jika kedua angka 0 atau kedua angka 1 (bitwise Xor). Misal, A = 2 (binernya 10) dan B = 3 (binernya 11), maka:</p> <p>(A ^ B) menghasilkan 1</p>
~	<p>Membalikkan suatu angka yang 1 menjadi 0 dan yang 0 menjadi 1 (bitwise Not). Biasanya hasilnya lebih besar satu angka tapi bernilai negatif. Misal, A = 3 (binernya 11) , maka:</p> <p>(~A) menghasilkan -4</p>

>>	Menggeser ke kanan bit operand pertama sejumlah angka operand kedua (Right Shift). Misal A = 9 (binernya 1001), maka: (A>>1) menghasilkan 4 (binernya 100)
>>>	Hampir sama dengan >> hanya di sebelah kiri selalu 0.

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Operator Bitwise</title>
5   </head>
6   <body>
7     <script>
8       var A = 5, B = 32;
9       document.write("5 << 2 hasilnya ");
10      document.write(A << 2);
11      document.write("<br/>");
12
13      document.write("32 >> 2 hasilnya ");
14      document.write(B >> 2);
15      document.write("<br/>");
16    </script>
17  </body>
18 </html>
```



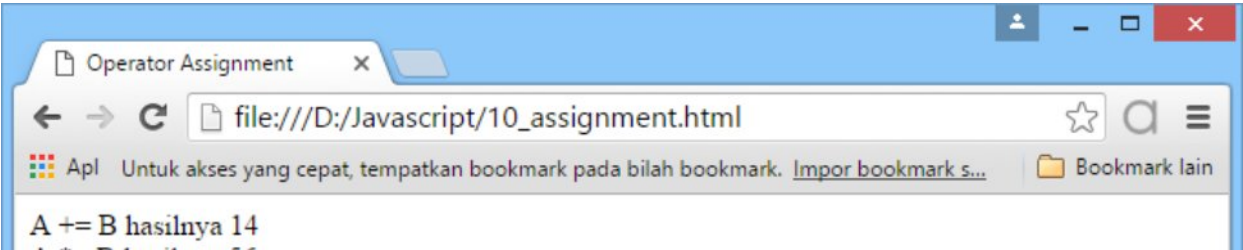
5. Operator Assignment

Operator ini digunakan untuk memberikan nilai kepada suatu variabel menggunakan tanda sama dengan. Daftar operator assignment dan penjelasanya dapat dilihat pada tabel berikut:

Operator	Keterangan
=	Memberi nilai pada suatu variabel dengan nilai variabel atau hasil perhitungan di sebelah kanan. Misal A = 3 dan B = 4, maka:

	A += B artinya nilai A ditambah nilai B, sehingga nilai A menjadi 7
-=	Mengurangi nilai suatu variabel dengan nilai atau perhitungan di sebelah kanan. Misal A = 3 dan B = 4, maka: B -= A artinya nilai B dikurangi nilai A, sehingga nilai B menjadi 1
*=	Mengalikan nilai suatu variabel dengan nilai atau perhitungan di sebelah kanan. Misal A = 3 dan B = 4, maka: B *= A artinya nilai B dikalikan nilai A, sehingga nilai B menjadi 12
/=	Membagikan nilai suatu variabel dengan nilai atau perhitungan di sebelah kanan. Misal A = 12 dan B = 4, maka: A /= B artinya nilai A dibagi nilai B, sehingga nilai A menjadi 3
%=	Mengambil sisa bagi suatu variabel dengan nilai atau perhitungan di sebelah kanan. Misal A = 12 dan B = 4, maka: A %= B artinya nilai sisa bagi A dengan B, sehingga nilai A menjadi 0

```
1  <!DOCTYPE HTML>
2  <html>
3      <head>
4          <title>Operator Assignment</title>
5      </head>
6      <body>
7          <script>
8              var A = 10, B = 4;
9              document.write(" A += B hasilnya ");
10             A += B;
11             document.write(A);
12             document.write("<br/>");
13
14             document.write("A *= B hasilnya ");
15             A *= B;
16             document.write(A);
17             document.write("<br/>");
18         </script>
19     </body>
20 </html>
```

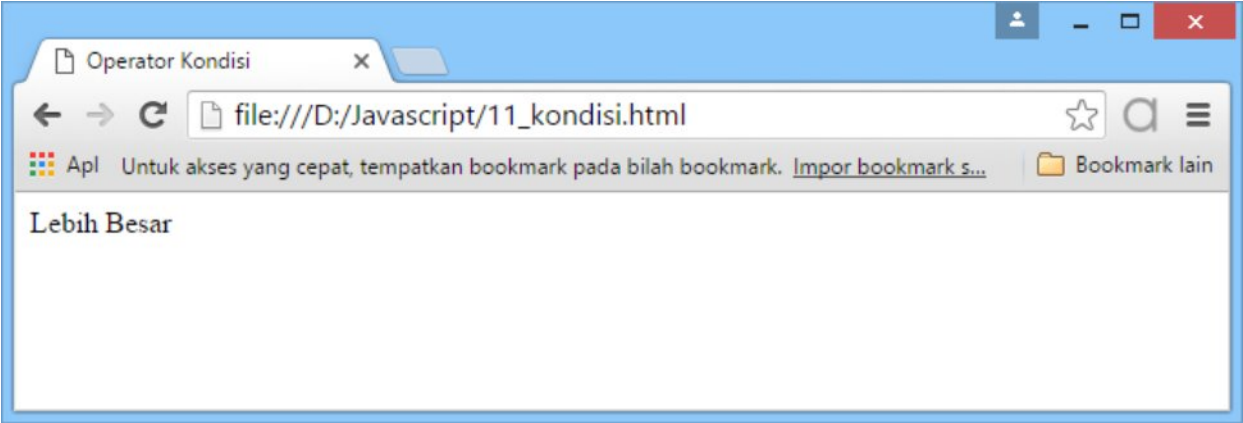


6. Operator Kondisi

Operator ini digunakan untuk memberikan nilai pada suatu variabel sesuai kondisi yang ditentukan. Jika kondisi **true** maka akan diberi nilai dengan nilai di sebelah kiri tanda titik dua, sedangkan jika kondisi **false** maka akan diberi nilai dengan nilai di sebelah kanan tanda titik dua. Kondisi diletakkan di dalam kurung diikuti dengan tanda **?** yang memisahkan dengan nilai.

Operator	Keterangan
? ... : ...	Tanda tanya memisahkan kondisi dengan nilai, sedangkan tanda titik dua memisahkan nilai pertama dan nilai kedua. Nilai pertama dipakai jika kondisi true dan nilai kedua dipakai jika kondisi false .

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Operator Kondisi</title>
5   </head>
6   <body>
7     <script>
8       var A = 10, B = 4;
9       var kata = (A>4) ? "Lebih Besar" : "Lebih Kecil";
10      document.write(kata);
11    </script>
12  </body>
13 </html>
```



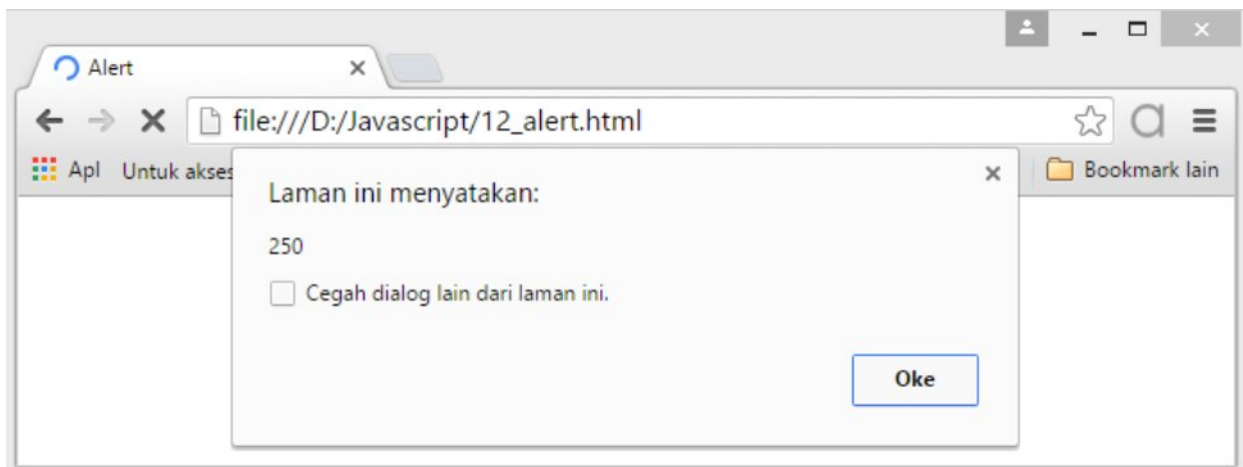
H. Menampilkan Hasil Program

Ada berbagai cara menampilkan hasil program yang dibuat dengan javascript baik untuk tujuan utama dibuatnya program maupun untuk menampilkan pesan kesalahan ketika terjadi kesalahan pada program.

1. Menampilkan hasil program dengan alert()

Cara ini merupakan yang paling sering digunakan yaitu dengan menampilkan hasil program melalui

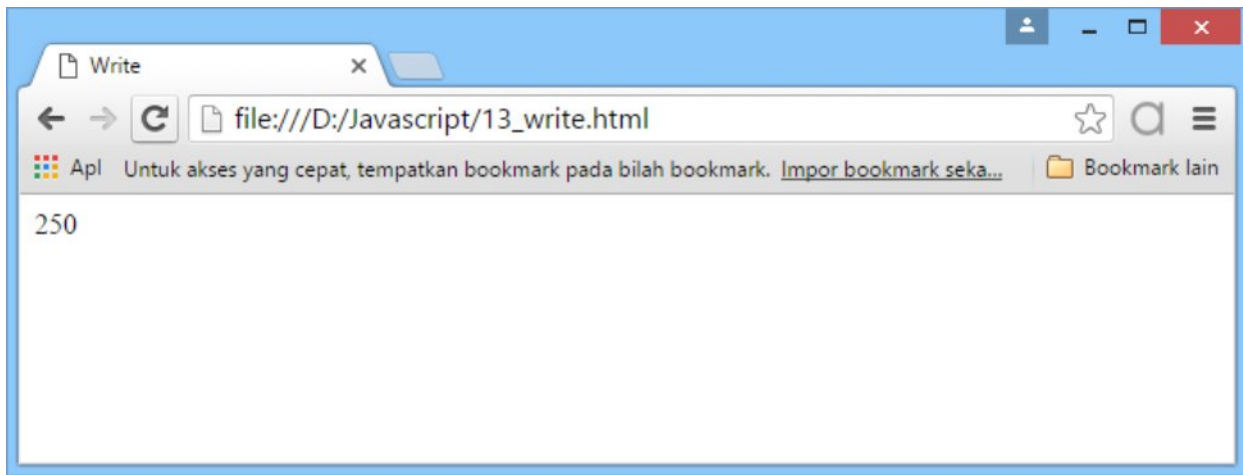
```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Alert</title>
5     <script>
6       var hasil = 100 + 150;
7       alert(hasil);
8     </script>
9   </head>
10  <body>
11  </body>
12 </html>
```



2. Menampilkan hasil program dengan write()

Caranya yaitu dengan menuliskan teks atau variabel di dalam tanda kurung fungsi **document.write()**. Hasil program yang ditampilkan dengan cara ini akan langsung tampil pada halaman web.

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Write</title>
5     <script>
6       var hasil = 100 + 150;
7       document.write(hasil);
8     </script>
9   </head>
10  <body>
11  </body>
12 </html>
```

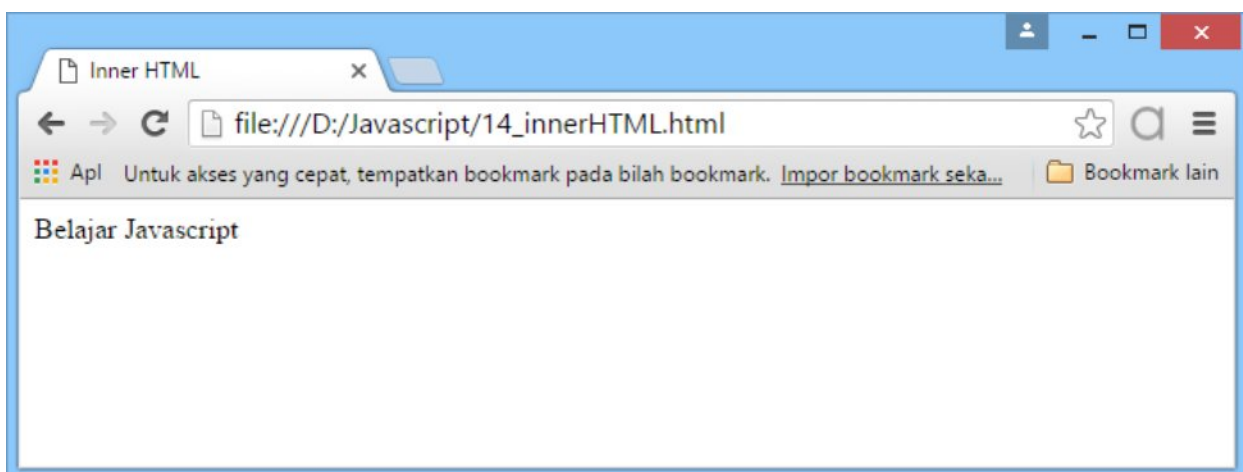


3. Menampilkan hasil program di dalam tag HTML

Caranya yaitu dengan membuat tag kontainer terlebih dahulu yang diberi atribut id tertentu. Kemudian mendeteksi tag tersebut dengan fungsi **getElementById()**. Untuk menampilkan hasil program, tuliskan atau teks atau variabel setelah sama dengan setelah properti **innerHTML**. Penggunaan cara ini harus disertai dengan penggunaan event onload.

```

1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <title>Inner HTML</title>
5      <script>
6        window.onload = function() {
7          var pesan = document.getElementById("pesan");
8          pesan.innerHTML = "Belajar Javascript";
9        }
10     </script>
11   </head>
12   <body>
13     <div id="pesan"></div>
14   </body>
15 </html>
  
```

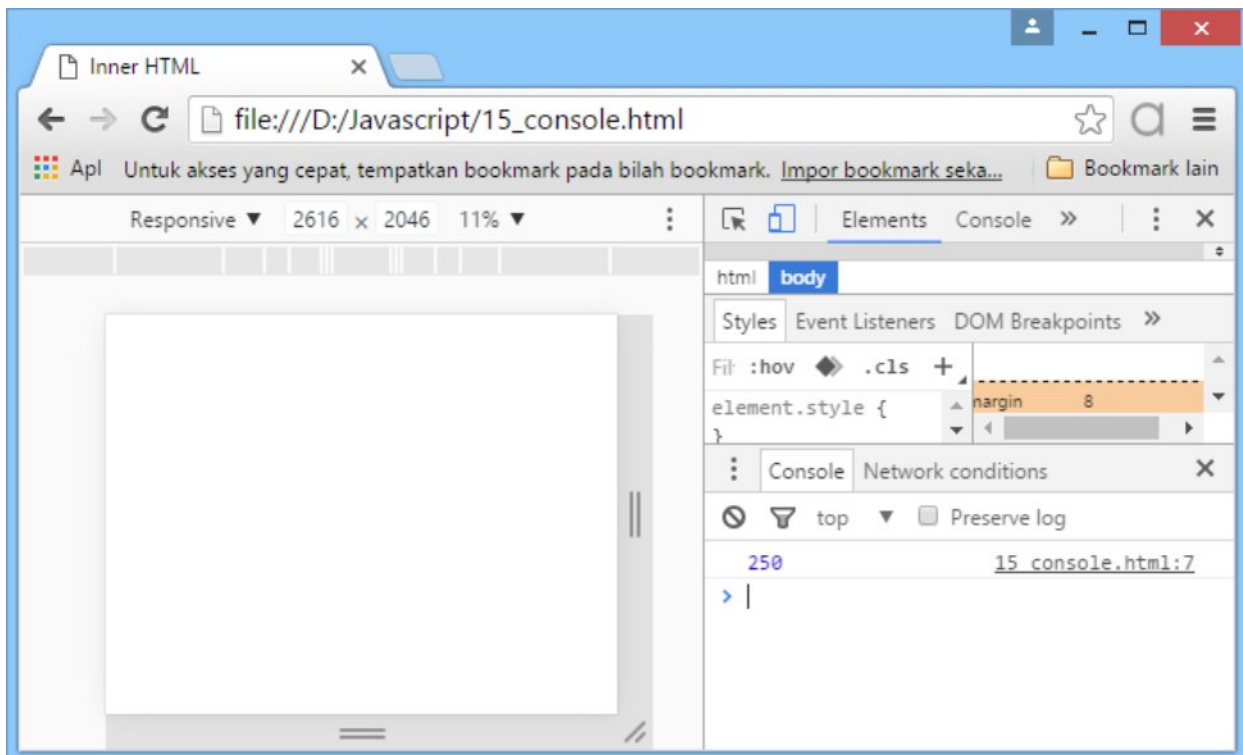


4. Menampilkan hasil program dengan console log

```

1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Console Log</title>
5     <script>
6       var hasil = 100+150;
7       console.log(hasil);
8     </script>
9   </head>
10  <body>
11  </body>
12 </html>

```



I. Komentar

Komentar pada pemrograman merupakan teks yang tidak akan dieksekusi sebagai program. Komentar biasanya untuk memberi keterangan pada skrip program. Cara menuliskan komentar pada javascript yaitu dengan meletakkan tanda `//` sebelum komentar jika komentar hanya satu baris. Jika komentar lebih dari satu baris, maka komentar ditulis di antara tanda `/*` dan `*/`.

```

1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Komentar</title>
5     <script>
6       //Ini komentar satu baris
7
8       /*Ini komentar lebih dari satu baris
9        komentar ini tidak akan dieksekusi sebagai program */
10    </script>
11  </head>

```

J. Percabangan

Percabangan pada bahasa pemrograman digunakan untuk menjalankan suatu program jika suatu kondisi terpenuhi. Ada berbagai jenis percabangan pada javascript yang dapat kita gunakan sesuai kebutuhan.

1. Percabangan dengan if

Percabangan ini digunakan untuk menjalankan suatu program jika kondisi terpenuhi (**true**) dan jika kondisi tidak terpenuhi tidak ada alternatif lain untuk dijalankan. Penulisan kondisi biasanya menggunakan variabel dengan tipe **boolean** atau menggunakan operator **logika** yang akan menghasilkan tipe data boolean.

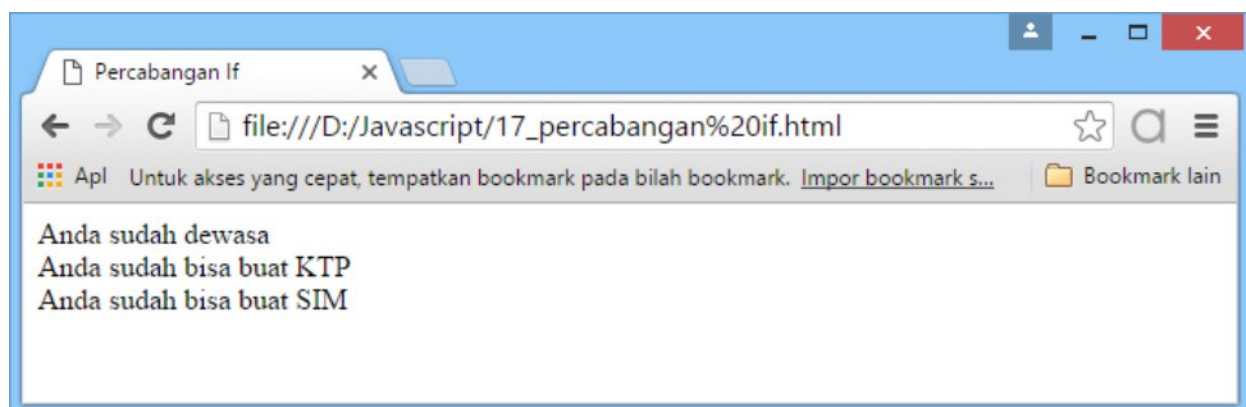
Penggunaan **if** dapat ditulis dengan format sebagai berikut:

```
if(kondisi) pernyataan
```

Jika pernyataan lebih dari satu baris, cara di atas tidak dapat digunakan, tapi menggunakan format berikut:

```
if(kondisi){
    pernyataan baris pertama;
    pernyataan baris kedua;
}
```

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Percabangan If</title>
5     <script>
6       var umur = 20;
7       if(umur > 17) document.write("Anda sudah dewasa");
8       if(umur > 20) document.write("Anda sudah boleh menikah");
9
10      var dewasa = true;
11      if(dewasa){
12        document.write("<br>Anda sudah bisa buat KTP");
13        document.write("<br>Anda sudah bisa buat SIM");
14      }
15    </script>
16  </head>
17  <body>
18  </body>
19 </html>
```



Percabangan ini digunakan untuk menjalankan program jika kondisi terpenuhi, dan jika kondisi tidak terpenuhi ada alternatif untuk dijalankan. Cara menggunakannya yaitu dengan format sebagai berikut:

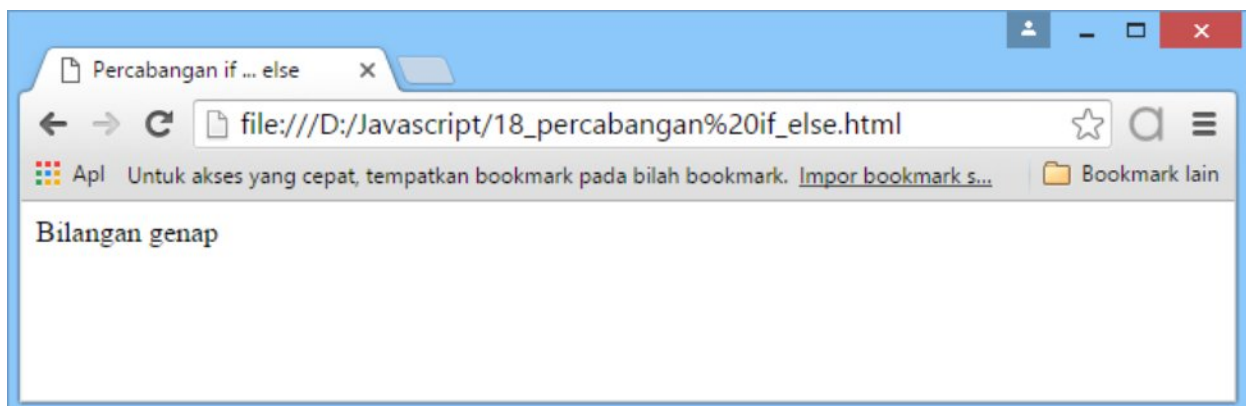
```
if(kondisi) pernyataan jika kondisi terpenuhi;
else pernyataan jika kondisi tidak terpenuhi;
```

Jika pernyataan lebih dari satu baris, maka formatnya menjadi seperti berikut:

```
if(kondisi){
    pernyataan jika kondisi terpenuhi;
}else{
    pernyataan jika kondisi tidak terpenuhi;
}
```

```

1  <!DOCTYPE HTML>
2  <html>
3      <head>
4          <title>Percabangan if ... else</title>
5          <script>
6              var angka = 20;
7              if(angka%2 == 0) document.write("Bilangan genap");
8              else document.write("Bilangan ganjil");
9          </script>
10     </head>
11     <body>
12     </body>
13 </html>
```



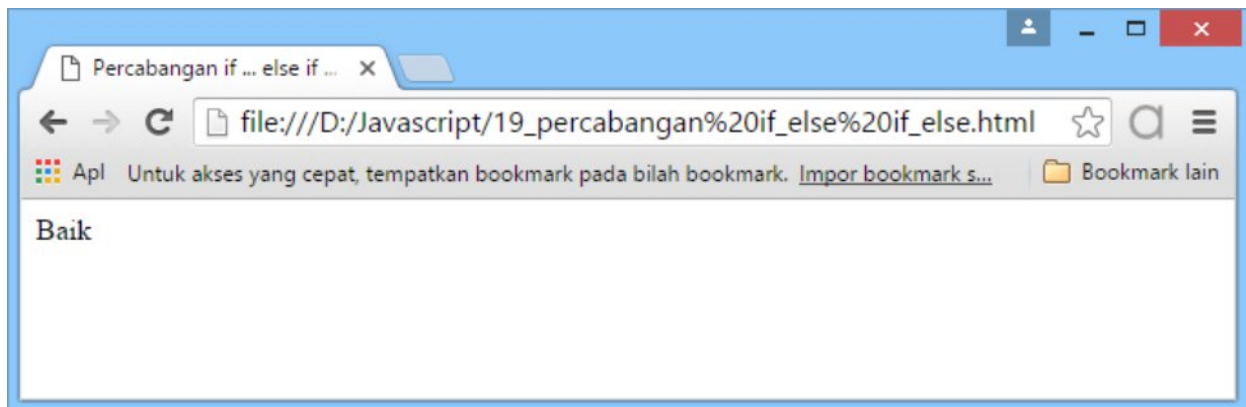
3. Percabangan dengan if ... else if... else

Percabangan ini digunakan untuk menjalankan program jika kondisi terpenuhi, dan jika kondisi tidak terpenuhi ada pengecekan kondisi berikutnya hingga tidak ada lagi kondisi untuk dicek baru dijalankan alternatif terakhir. Cara menggunakannya yaitu dengan format sebagai berikut:

```
if(kondisi){
    pernyataan jika kondisi terpenuhi;
}else if(kondisi kedua){
    pernyataan jika kondisi kedua terpenuhi;
}else if(kondisi ketiga){
    pernyataan jika kondisi ketiga terpenuhi;
}else{
    pernyataan jika tidak ada kondisi terpenuhi;
}
```



```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Percabangan if ... else if ... else</title>
5     <script>
6       var nilai = 80;
7       if(nilai > 90) document.write("Sangat Baik");
8       else if(nilai > 75) document.write("Baik");
9       else if(nilai > 60) document.write("Cukup");
10      else document.write("Kurang");
11    </script>
12  </head>
13  <body>
14  </body>
15 </html>
```

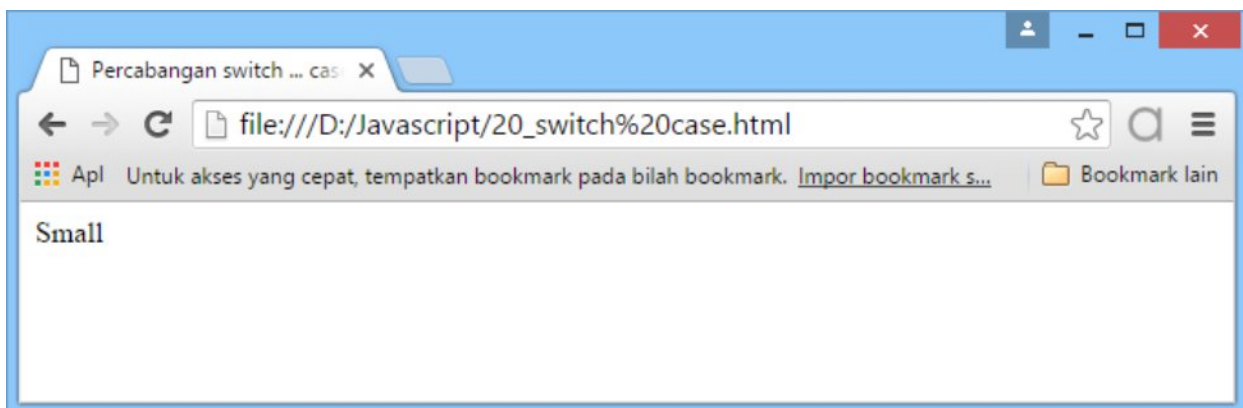


4. Percabangan dengan switch ... case

Percabangan ini merupakan alternatif untuk menggantikan percabangan dengan if ... else if ... else. Namun, jika variabel yang dicek sama, cara ini lebih disarankan daripada menggunakan if ... else if ... else. Cara penggunaannya yaitu sebagai berikut:

```
switch (variabel){
  case kondisi 1: pernyataan;
    break;
  case kondisi 2: pernyataan;
    break;
  case kondisi 3: pernyataan;
    break;
  default: pernyataan default;
}
```

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Percabangan switch ... case</title>
5     <script>
6       var ukuran = "S";
7       switch(ukuran) {
8         case "L": document.write("Large");
9           break;
10        case "M": document.write("Medium");
11          break;
12        default: document.write("Small");
13      }
14    </script>
15  </head>
16  <body>
17  </body>
18 </html>
```



K. Perulangan

Perulangan pada bahasa pemrograman digunakan untuk menjalankan suatu program secara berulang-ulang sampai batas yang telah ditentukan. Ada berbagai jenis perulangan pada javascript yang dapat kita gunakan sesuai kebutuhan.

1. Perulangan dengan while

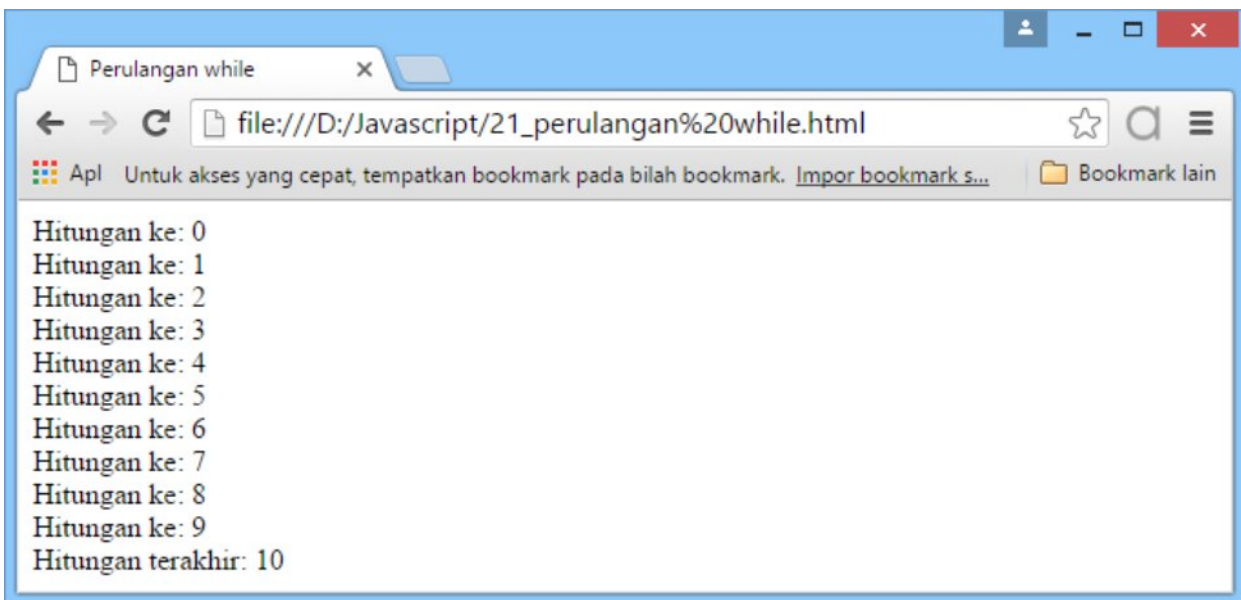
Perulangan ini digunakan untuk menjalankan program berulang-ulang selama kondisi yang ditentukan terpenuhi. Cara penggunaanya yaitu sebagai berikut:

```
while(kondisi){
  pernyataan;
}
```

```

1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Perulangan while</title>
5     <script>
6       var hitung = 0;
7       while(hitung < 10){
8         document.write("Hitungan ke: " + hitung + "<br>");
9         hitung++;
10      }
11      document.write("Hitungan mencapai " + hitung + "<br>");
12    </script>
13  </head>
14  <body>
15  </body>
16 </html>

```



2. Perulangan dengan do ... while

Perulangan ini digunakan untuk menjalankan program berulang-ulang hingga kondisi yang ditentukan terpenuhi. Cara penggunaannya yaitu sebagai berikut:

```

do{
  Pernyataan;
} while(kondisi);

```

```

1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Perulangan do ... while</title>
5     <script>
6       var hitung = 0;
7       do{
8         document.write("Hitungan ke: " + hitung + "<br>");
9         hitung++;
10      } while (hitung<10);
11      document.write("Hitungan mencapai " + hitung + "<br>");
12    </script>

```

Hasilnya sama dengan perulangan menggunakan **while**.

3. Perulangan dengan for

Perulangan ini digunakan untuk menjalankan program berulang-ulang dengan menentukan nilai awal, nilai akhir, dan penambahan/pengurangan nilai. Cara penggunaanya sebagai berikut:

```
for(inisialisasi; cek kondisi; penambahan/pengurangan nilai){
    pernyataan;
}
```

```

1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <title>Perulangan for</title>
5      <script>
6          var hitung;
7          for(hitung=0; hitung<10; hitung++){
8              document.write("Hitungan ke: " + hitung + "<br>");
9          }
10         document.write("Hitungan mencapai " + hitung + "<br>");
11     </script>
12 </head>
13 <body>
14 </body>
15 </html>
```

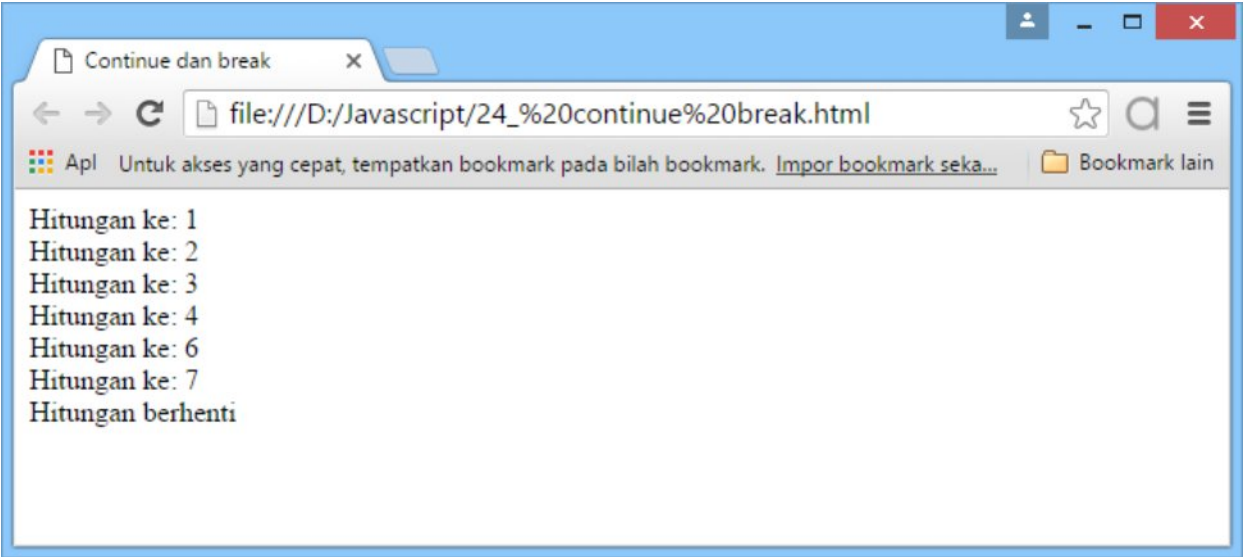
Hasilnya sama dengan perulangan menggunakan **while**.

L. Continue dan Break

Continue digunakan untuk melewati suatu perulangan karena kondisi tertentu. Sedangkan **break** digunakan untuk menghentikan paksa suatu perulangan sebelum kondisi perulangan berakhir karena kondisi tertentu.

```

1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <title>Continue dan break</title>
5      <script>
6          var hitung = 0;
7          while(hitung<10){
8              hitung++;
9              if(hitung == 5) continue; //dilewati
10             if(hitung == 8) break; //berhenti
11             document.write("Hitungan ke: " + hitung + "<br>");
12         }
13         document.write("Hitungan berhenti");
14     </script>
15 </head>
16 <body>
17 </body>
18 </html>
```

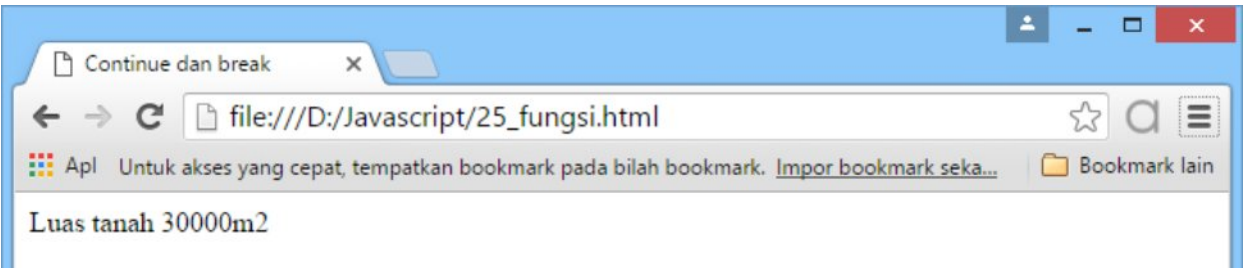


M. Fungsi

Fungsi merupakan blok program yang dapat digunakan secara berulang-ulang dengan memanggil nama fungsi. Fungsi dapat mengembalikan sebuah nilai dengan perintah **return** atau tanpa mengembalikan nilai. Pemanggilan fungsi yang mengembalikan nilai dapat digunakan sebagai nilai suatu variabel. Cara menulis fungsi yaitu sebagai berikut:

```
function nama_fungsi(parameter1, parameter 2){
  pernyataan;
  return variabel;
}
```

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Continue dan break</title>
5     <script>
6       function hitung_luas(p, l){
7         var L = p * l;
8         return L;
9       }
10      function tampilkan_hasil(){
11        var teks = "Luas tanah " + hitung_luas(200, 150) + "m2";
12        document.write(teks);
13      }
14      tampilkan_hasil();
15    </script>
16  </head>
17  <body>
18  </body>
19 </html>
```



N. Event

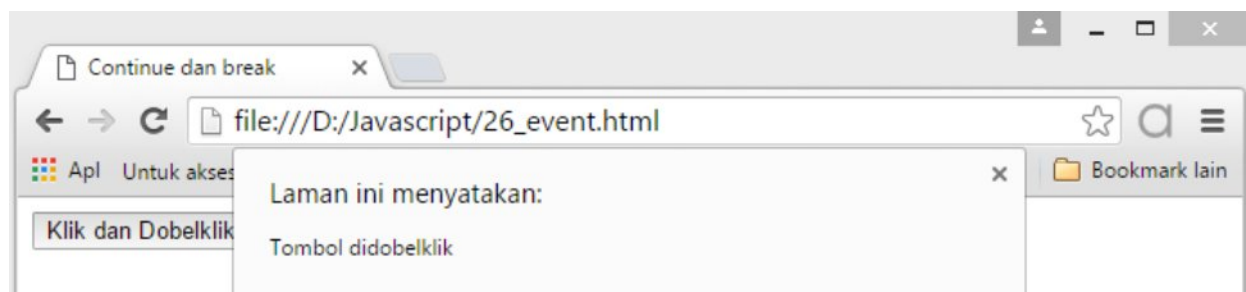
Event merupakan suatu perlakuan yang dilakukan terhadap suatu elemen HTML. Ada berbagai macam event yang dapat digunakan, yaitu sebagai berikut:

- onblur, ketika suatu elemen tidak focus
- onchange, ketika nilai suatu elemen diubah
- onclick, ketika suatu elemen diklik
- ondblclick, ketika suatu elemen didobelklik
- onerror, ketika error
- onfocus, ketika suatu elemen sedang fokus
- onkeydown, ketika tombol huruf pada keyboard ditekan
- onkeypress, ketika tombol keyboard ditekan
- onkeyup, ketika tombol keyboard di lepas
- onload, ketika dokumen di-load
- onmousedown, ketika tombol mouse ditekan
- onmousemove, ketika mouse digerakkan
- onmouseout, ketika mouse keluar dari area suatu elemen
- onmouseover, ketika mouse melewati area suatu elemen
- onmouseup, ketika tombol mouse dilepas setelah diklik
- onreset, ketika form direset
- onresize, ketika elemen diubah ukurannya
- onselect, ketika suatu elemen dipilih
- onsubmit, ketika data form dikirim
- onunload, ketika keluar dari suatu dokumen

```

1  <!DOCTYPE HTML>
2  <html>
3      <head>
4          <title>Continue dan break</title>
5          <script>
6              function dobelklik() {
7                  alert("Tombol didobelklik");
8              }
9          </script>
10     </head>
11     <body>
12         <button ondblclick="dobelklik()">
13             Klik dan Dobelklik</button>
14     </body>
15 </html>

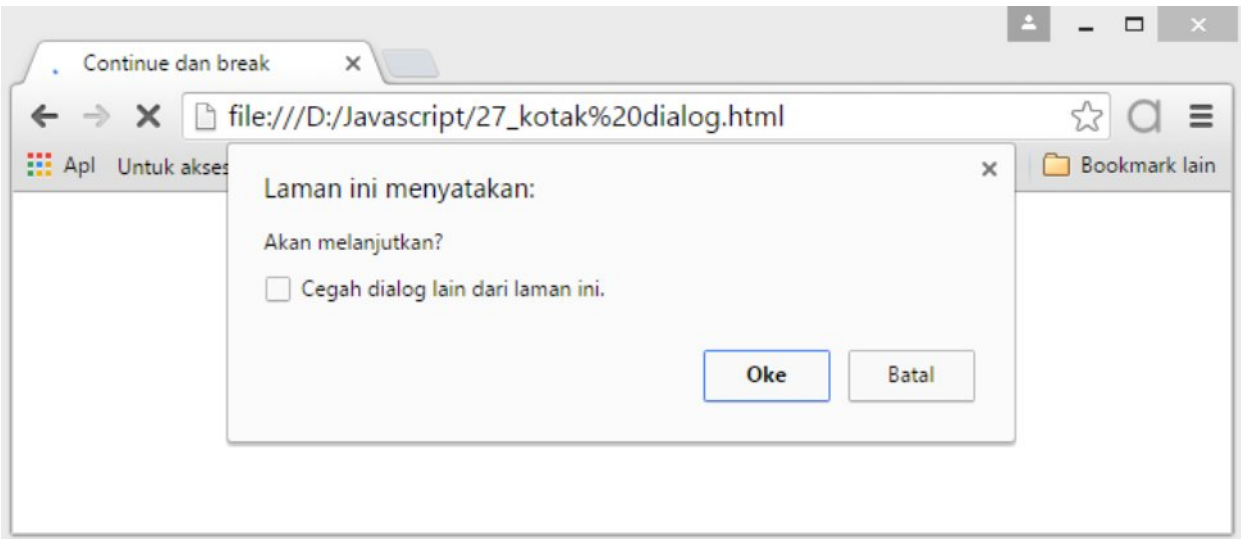
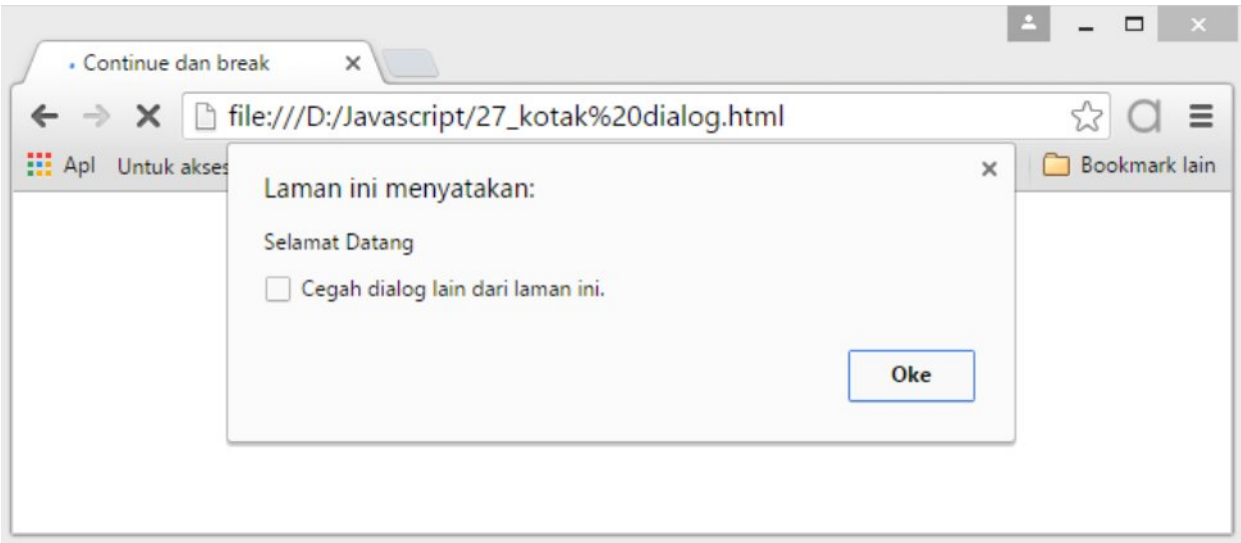
```

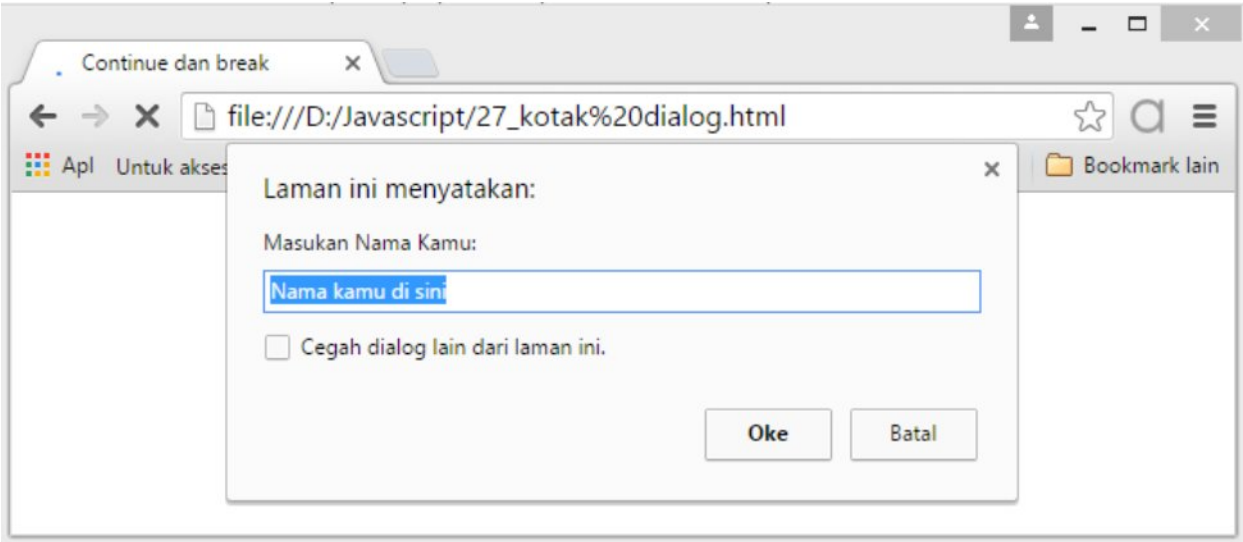


O. Kotak Dialog

Ada 3 macam kotak dialog pada javascript yaitu dengan fungsi **alert()**, **confirm()**, dan **prompt()**.

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Continue dan break</title>
5     <script>
6       alert("Selamat Datang");
7       confirm("Akan melanjutkan?");
8       prompt("Masukan Nama Kamu:", "Nama kamu di sini");
9     </script>
10  </head>
11  <body>
12  </body>
13 </html>
```





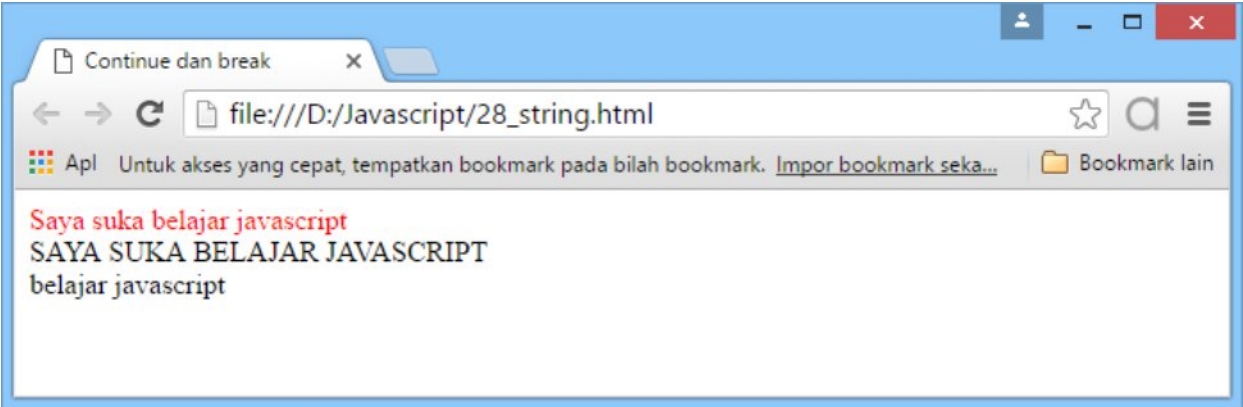
P. Objek String

Seperti telah dijelaskan sebelumnya bahwa setiap variabel akan menjadi objek sehingga akan otomatis memiliki properti dan method. Untuk variabel dengan tipe data string memiliki properti dan method sebagai berikut:

Properti / Method	Keterangan
length	Menghasilkan jumlah karakter dari suatu teks
anchor(nama)	Menghasilkan link sama seperti menggunakan tag
big()	Menjadikan teks menjadi lebih besar karena diapit tag <big>
blink()	Menjadikan teks berkedip karena diapit tag <blink>
bold()	Menjadikan teks bercetak tebal karena diapit tag
fontcolor(warna)	Menjadikan teks berwarna karena diapit tag
fontsize(ukuran)	Mengubah ukuran teks karena diapit tag
italics()	Menjadikan teks miring karena diapit tag <i>
link(url)	Merubah teks menjadi link karena diapit tag
small()	Menjadikan teks kecil karena diapit tag <small>
strike()	Menjadikan teks tercoret karena diapit tag <strike>
sub()	Menjadikan teks subscript karena diapit tag <sub>
sup()	Menjadikan teks superscript karena diapit tag <sup>
toLowerCase()	Mengubah teks menjadi huruf kecil
toUpperCase()	Mengubah teks menjadi huruf besar
charAt(x)	Menghasilkan karakter pada posisi x pada string
charCodeAt(x)	Menghasilkan nilai Unicode dari karakter pada posisi x pada suatu string
concat(teks1, teks2, ...)	Menggabungkan dua atau lebih string menjadi satu.
fromCharCode(c1, c2,)	Menghasilkan string yang dibuat menggunakan urutan dari unicode
indexOf(substr, mulai)	Mencari dan menghasilkan nomor index
charCodeAt(x)	Menghasilkan nilai unicode value dari karakter pada posisi x pada suatu string

indexOf(substr, [start])	Menghasilkan nomor index dari karakter atau substring yang ada di dalam kurung.
match(regex)	Mencocokkan suatu string dengan pola regular expression
replace(regex, replacetext)	Menukar suatu string berdasarkan pola regular expression
search(regex)	Mengetes apakah string cocok dengan pola regular expression, jika cocok akan menghasilkan index dari yang cocok.
slice(start, end)	Menghasilkan substring dari suatu string dimulai dari parameter pertama dan diakhiri parameter kedua
split(delimiter, limit)	Memecah string menjadi array berdasarkan tanda pemisah yang ditentukan.
substr(start, length)	Menghasilkan potongan dari suatu string yang dimulai dari parameter pertama sebanyak parameter kedua.

```
1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <title>Manipulasi string</title>
5      <script>
6        var teks = "Saya suka belajar javascript";
7        document.write(teks.fontcolor('red') + '<br>');
8        document.write(teks.toUpperCase() + '<br>');
9        document.write(teks.substr(10, 18));
10     </script>
11   </head>
12   <body>
13   </body>
14 </html>
```



Q. Objek Window

Ada beberapa method yang digunakan untuk manipulasi window atau dokumen, yaitu sebagai berikut

Properti / Method	Keterangan
reload()	Digunakan untuk mereload window, contoh: window.location.reload()

print()	Digunakan untuk membuka jendela print, contoh: window.close()
location	Digunakan untuk pindah ke halaman lain. window.location = "url"

R. Method Tanggal dan Waktu

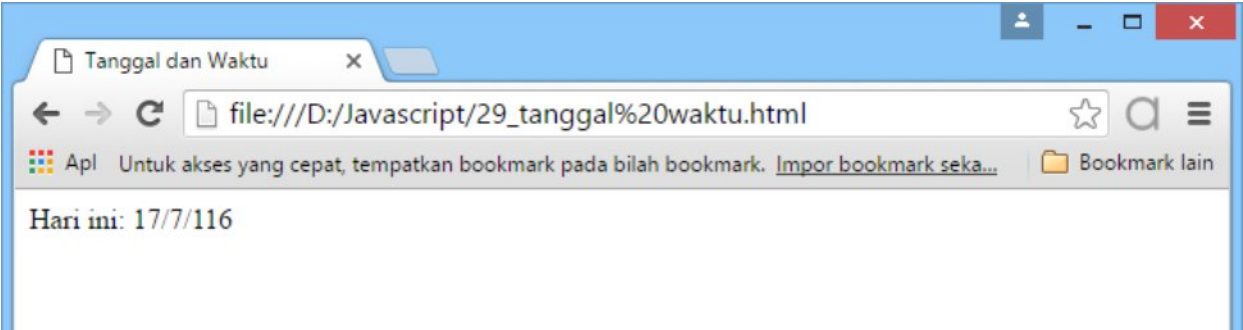
Ada berbagai method untuk memanipulasi tanggal atau waktu. Untuk menggunakan method tersebut, harus memanggil objek Date() terlebih dahulu seperti berikut:

```
var tanggal = new Date();
```

Selanjutnya method tersebut digunakan pada variabel tanggal. Adapun daftar method yang bisa dipakai sebagai berikut:

Properti / Method	Keterangan
getDate()	Menghasilkan tanggal dalam satu bulan.
getDay()	Menghasilkan nama hari.
getHours()	Menghasilkan jam dimulai dari 0 hingga 23
getMinutes()	Menghasilkan menit.
getSeconds()	Menghasilkan detik.
getMonth()	Menghasilkan bulan dimulai dari 0 hingga 11
getFullYear()	Menghasilkan tahun
getTime()	Menghasilkan waktu yang lengkap.

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Tanggal dan Waktu</title>
5     <script>
6       var waktu = new Date();
7       document.write('Hari ini: ' + waktu.getDate() + '/'
8         + waktu.getMonth() + '/' + waktu.getFullYear() );
9     </script>
10  </head>
11  <body>
12  </body>
13 </html>
```



Array merupakan sebuah variabel yang dapat menampung beberapa data sekaligus yang terindeks secara teratur. Untuk mendeklarasikan array ada beberapa cara yaitu sebagai berikut:

Cara 1

```
var buah = new Array();
buah[0] = "Mangga";
buah[1] = "Apel";
buah[2] = "Jeruk";
```

Cara 2

```
Var buah = new Array("Mangga", "Apel", "Jeruk");
```

Cara 3

```
Var buah = ["Mangga", "Apel", "Jeruk"];
```

Cara menggunakan array yaitu dengan menyebutkan nama variabel dan indeks yang diletakkan di dalam kurung siku.

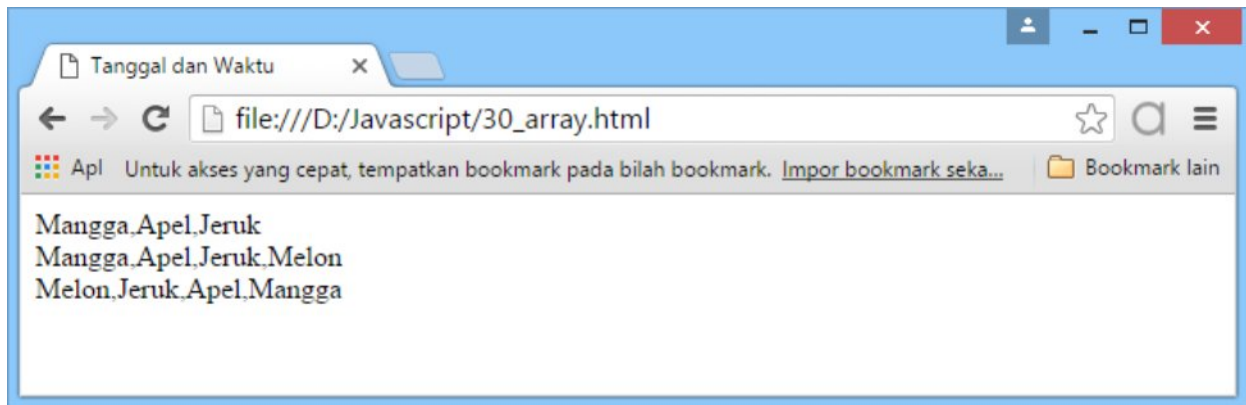
```
document.write(buah[1]);
```

Skip di atas akan menghasilkan tulisan **"Mangga"**, karena indeks dimulai dari 0.

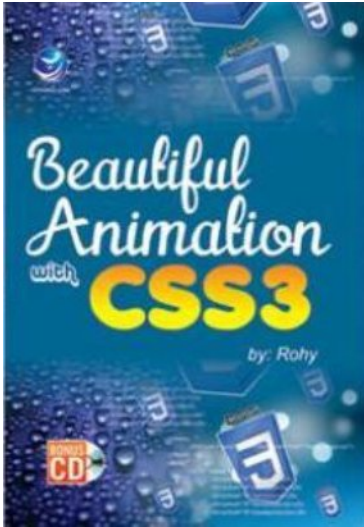
Beberapa properti dan method yang bisa digunakan untuk memanipulasi array yaitu sebagai berikut:

Properti / Method	Keterangan
length	Menghasilkan jumlah elemen dalam satu array
join()	Digunakan untuk menggabungkan nilai array menjadi string yang dipisahkan dengan tanda yang ditentukan.
concat()	Menggabungkan dua array atau lebih menjadi array baru.
pop()	Digunakan untuk menghapus elemen terakhir pada array
push()	Digunakan untuk menambahkan satu atau lebih elemen baru ke bagian akhir array.
reverse()	Digunakan untuk membalik urutan elemen array.
shift()	Digunakan untuk menghapus elemen pertama dari suatu array
unshift()	Digunakan untuk menambahkan satu atau lebih elemen pada awal array
sort()	Digunakan untuk mengurutkan elemen array

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Tanggal dan Waktu</title>
5     <script>
6       var buah = ["Mangga", "Apel", "Jeruk"];
7       document.write(buah + '<br>');
8       buah.push("Melon");
9       document.write(buah + '<br>');
10      buah.reverse();
11      document.write(buah + '<br>');
12    </script>
```



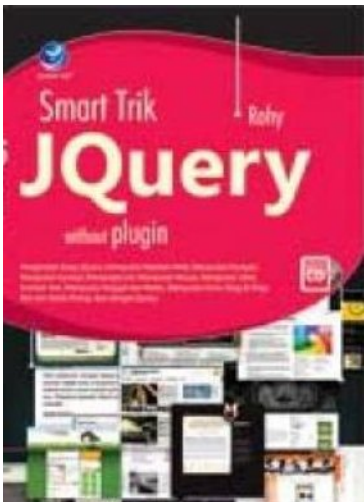
BUKU-BUKU KARYA PENULIS



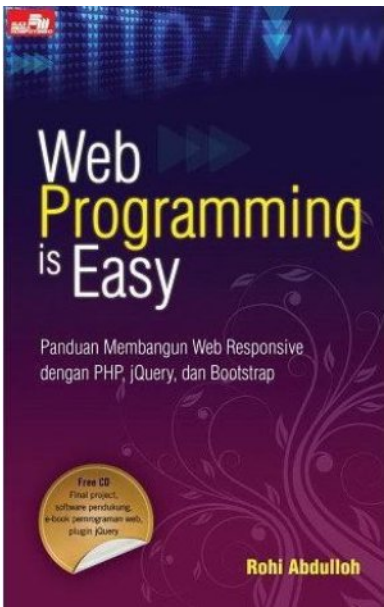
Judul : **Beatiful Animation with CSS3**
Ukuran/Halaman : 16 x 23 cm / 122 halaman
Penerbit : Andi Publisher
Tahun Terbit : 2014
Keterangan Singkat : Membahas 40+ tutorial animasi dengan CSS3 yang sering digunakan pada desain website ,diantaranya: manipulasi background, menu, tab, accordion, image viewer, image slider, social button, dsb.



Judul : **Be Creative with HTML5 Canvas**
Ukuran/Halaman : 14 x 21 cm / 204 halaman
Penerbit : Elex Media Komputindo
Tahun Terbit : 2014
Keterangan Singkat : Membahas penggunaan HTML5 Canvas mulai dari bentuk dasar, animasi, video, audio, menangani event, penerapan sifat-sifat fisika, hingga tutorial membuat jam analog, diagram (chart), aplikasi painting, graphic editor dan photo editor.



Judul : **Smart Trik JQuery without Plugin**
Ukuran/Halaman : 14 x 21 cm / 248 halaman
Penerbit : Andi Publisher
Tahun Terbit : 2015
Keterangan Singkat : Membahas 60+ tutorial jQuery tanpa menggunakan plugin, diantaranya : manipulasi gambar, tabel, tanggal & waktu, teks,



Judul : **Web Programming Is Easy**

Ukuran/Halaman : 14 x 21 cm / 224 halaman

Penerbit : Elex Media Komputindo

Tahun Terbit : 2015

Keterangan Singkat : Membahas panduan mudah membuat website bagi pemula mulai dari membuat halaman admin, halaman front-end, trik pengembangan website, menggunakan jQuery, desain responsive dengan Bootstrap, konversi ke mysql, hingga upload ke hosting.



Judul : **Super Dahsyat Responsive Web Design dengan Foundation 5**

Halaman : 192 halaman

Penerbit : Asfa Solution

Tahun Terbit : 2015

Keterangan Singkat : Membahas panduan membuat website responsive dengan foundation 5 mulai dari dasar-dasar foundation hingga penerapannya pada desain website baik halaman administrator maupun halaman pengunjung.



Judul : **Trik Rahasia Menjadi Master Wordpress Handal**

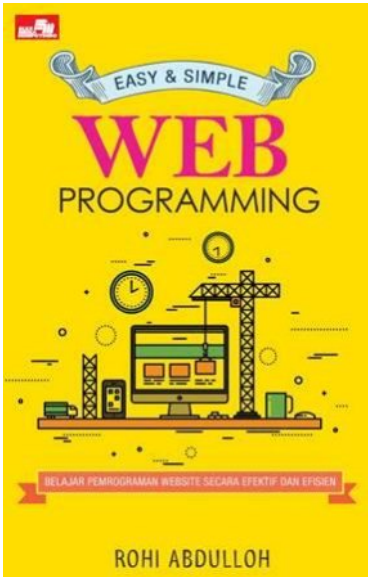
Halaman : 180 halaman

Penerbit : Asfa Solution

Tahun Terbit : 2015

Keterangan Singkat : Membahas pemrograman pada wordpress mulai dari dasar

widget, hingga membuat plugin sendiri.



Judul	: Easy & Simple Web Programming
Halaman	: 228 halaman
Penerbit	: Elex Media Komputindo
Tahun Terbit	: 2016
Keterangan Singkat	: Edisi kedua dari buku "Web Programming Is Easy" dengan penambahan 2 bab yaitu mengatasi error pada PHP dan menggunakan widget dari internet seperti google map, komentar facebook, tweet pada twitter, statistik pengunjung, polling, dan sebagainya.