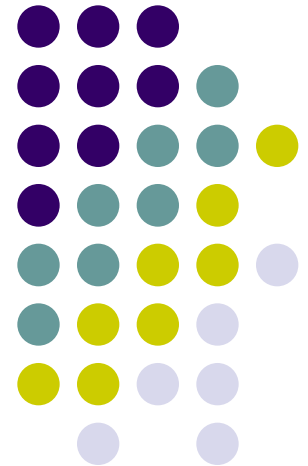


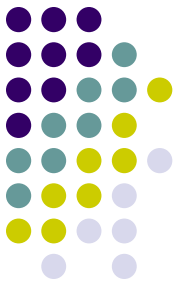
JavaScript

Pemrograman Web 1

Genap 2011 - 2012

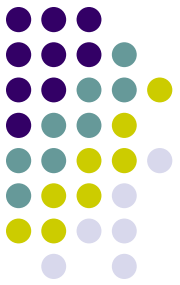


Popup Box



- JavaScript memiliki tiga macam Popup Box, yaitu :
 - Alert box
 - Biasanya digunakan untuk memberikan informasi ke pengguna.
 - Confirm box
 - Biasanya digunakan apabila aplikasi membutuhkan verifikasi atau persetujuan dari pengguna.
 - Prompt box.
 - Biasanya digunakan apabila aplikasi ingin pengguna memasukkan dahulu suatu nilai sebelum memasuki suatu halaman.

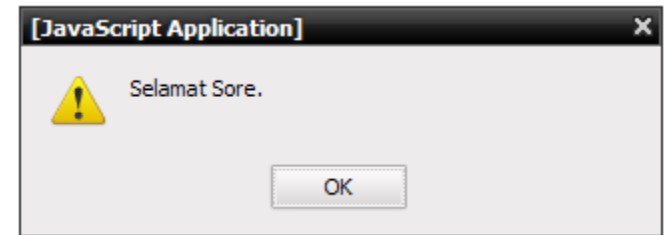
Popup Box - Alert

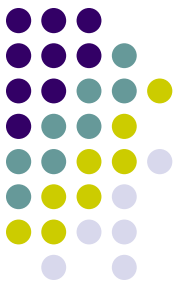


- **Contoh**

```
<script type="text/javascript">
    var jam = 17;
    if (jam <= 10)
    {
        alert("Selamat Pagi.");
    }
    else if (jam > 10 && jam <= 15)
    {
        alert("Selamat Siang.");
    }
    else
    {
        alert("Selamat Sore.");
    }
</script>
```

Hasil



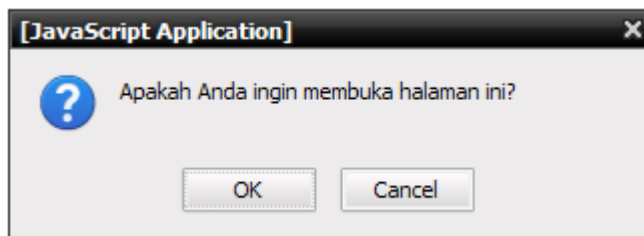


Popup Box - Confirm

- Contoh

```
<script type="text/javascript">
    var konfirmasi = confirm("Apakah Anda ingin membuka halaman ini?");
    if (konfirmasi == true)
        document.location.href = "halaman1.html";
    else
        document.location.href = "home.html";
</script>
```

- Hasil



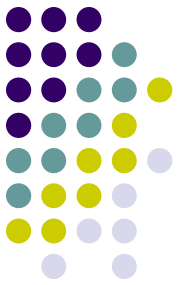
`document.location.href`

Sintaks JavaScript untuk mengalihkan browser ke halaman yang dituju (*redirect*).

`var konfirmasi = confirm();`

Jika tombol “OK” yang ditekan, maka confirm akan mengisi nilai “**true**” ke variabel konfirmasi, sebaliknya “**false**” jika tombol “Cancel” yang ditekan.

Popup Box - Prompt



- Sintaks

```
prompt("text","nilaiDefault");
```

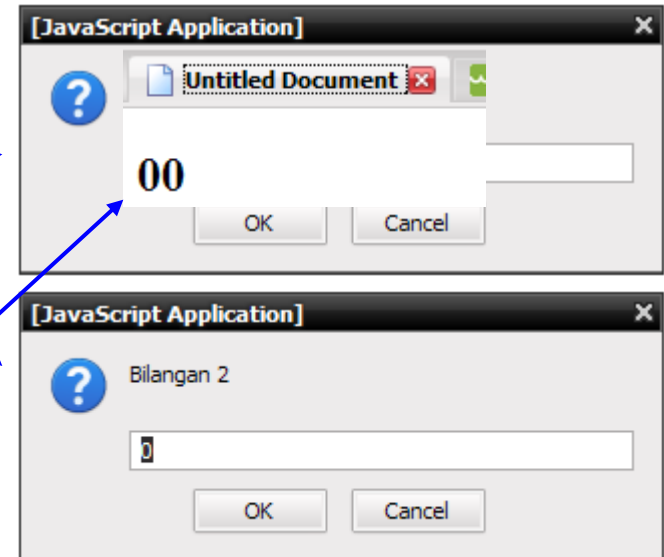
- Contoh

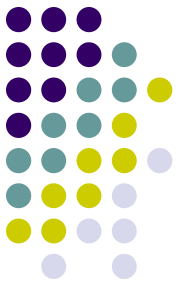
```
<script type="text/javascript">
    var bil1, bil2, jml;

    bil1 = prompt("Bilangan 1","0");
    bil2 = prompt("Bilangan 2","0");
    jml = bil1 + bil2;

    document.write("<h2>"+jml+"</h2>");
</script>
```

Hasil





Popup Box - Prompt

- **Penjelasan**

```
bil1 = prompt("Bilangan 1","0");
```

Fungsi **prompt** akan mengisikan nilai yang diinput oleh pengguna ke variabel bil1. Tipe data yang dikeluarkan oleh **prompt** bertipe **string**.

```
jml = bil1 + bil2;
```

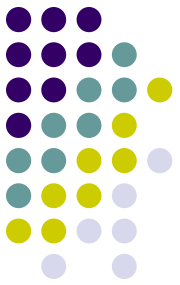
jml berisi nilai **"00"** karena **bil1** dan **bil2** bertipe string. Tanda plus "+" jika digunakan pada tipe string, maka akan menempelkan isi dari dua variabel tersebut (*concat*).

- **Konversi tipe data**

```
jml = parseInt(bil1) + parseInt(bil2);
```

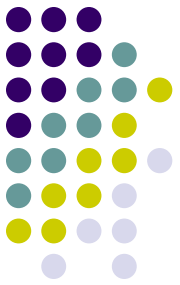
jml berisi nilai **"0"** karena tipe data **bil1** dan **bil2** dikonversi menjadi integer.

Fungsi



- Sebuah fungsi berisi *source code* yang akan dieksekusi ketika dipanggil.
- Sebuah fungsi akan dieksekusi jika dipanggil nama fungsinya atau dieksekusi oleh suatu *event*.
- Sisipkan *source code* ke dalam fungsi untuk mencegah suatu *source code* langsung tereksekusi saat halaman terbuka.
- Fungsi dapat dipanggil dari mana saja di suatu halaman bahkan dari file JavaScript yang berbeda (external javascript), jika pada halaman tersebut file-file javascript sudah disisipkan.
- Fungsi bisa mengembalikan nilai ke pemanggilnya.

Fungsi



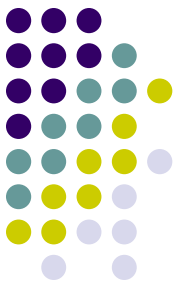
- **Sintaks**

```
function namaFungsi(parameter1, parameter2, ... parameterN)
{
    .....
}
```

- **Contoh**

```
<html>
<head>
    <script type="text/javascript">
        function displaymessage()
        {
            alert("Hello World!");
        }
    </script>
</head>

<body>
<form>
    <input type="button" value="Click me!" onclick="displaymessage()" />
</form>
</body>
</html>
```

Fungsi – Return value

- Contoh fungsi dengan return value

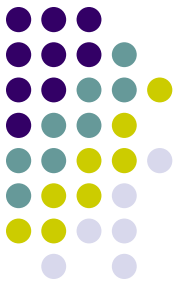
```
<html>
<head>
    <script type="text/javascript">
        function kali(a,b)
        {
            return a*b;
        }
    </script>
</head>

<body>
    <script type="text/javascript">
        var hasil = kali(4,3);
        document.write(hasil);
    </script>
</body>
</html>
```

Hasil

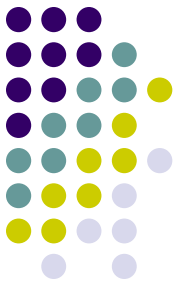


12



Event Handler

- *Event* adalah aksi yang dilakukan *user* terhadap elemen HTML yang dapat dideteksi oleh JavaScript.
- Setiap elemen di halaman web memiliki *event* tertentu yang dapat memicu JavaScript.
- *Event* didefinisikan dalam atribut tag HTML.
- *Event* digunakan dalam kombinasi dengan fungsi, dan fungsi tersebut tidak akan tereksekusi sebelum *event* tersebut terjadi.



Event – onLoad onUnload

- **onLoad**

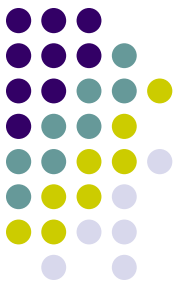
- *Event* yang terjadi saat memasuki suatu halaman.
- Biasanya dipakai untuk mendeteksi browser yang dipakai.

```
<body onLoad="alert('Selamat datang di website Knowbots Devp.')">  
    Ini halaman web.  
</body>
```

- **onUnload**

- *Event* yang terjadi saat keluar dari suatu halaman.
- Biasanya dipakai untuk menghapus cookie atau session.

```
<body onUnload="alert('Terima Kasih telah berkunjung.')">  
</body>
```

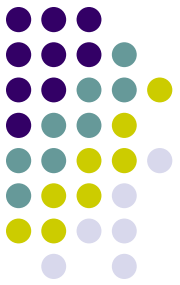


Event - onChange

- Event yang terjadi bila nilai suatu elemen berubah.
- **Contoh**

```
<html>
<head>
  <script type="text/javascript">
    function tampilKota(propinsi)
    {
      //Source code untuk tampil kota.
      alert("Tampil kota pada propinsi "+propinsi);
    }
  </script>
</head>

<body>
  <form>
    <select name="propinsi" onChange="tampilKota(this.value);">
      <option value="Jabar">Jawa Barat</option>
      <option value="Jateng">Jawa Tengah</option>
    </select>
  </form>
</body>
</html>
```



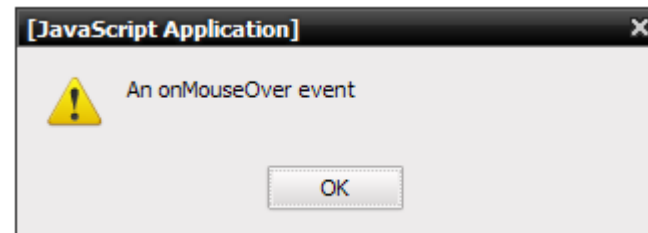
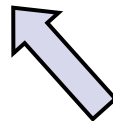
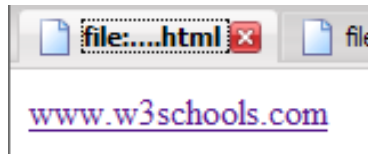
Event - onmouseover

- Event yang terjadi ketika kursor berada di atas suatu elemen.

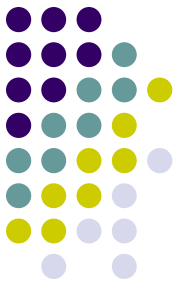
- **Contoh**

```
<a href="http://www.w3schools.com" onmouseover="alert('An onmouseover event');">  
  www.w3schools.com  
</a>
```

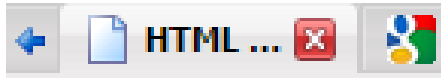
- **Hasil**



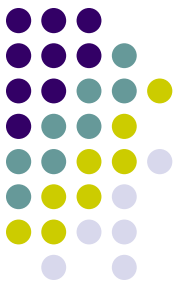
Timing



- Contoh berikut ini akan menampilkan jam digital yang perpindahan detik, menit dan jam-nya *realtime*.



20:22:42



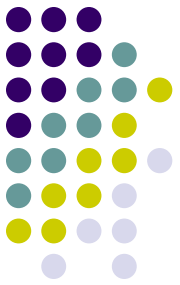
Timing

- Contoh (jam.html)

```
<head>
  <script type="text/javascript" src="jam.js"></script>
  <style type="text/css">
    .jam{
      font-size:24px;
      color:#0000FF;
    }
  </style>
</head>

<body onload="startTime()">
  <div id="jam" class="jam"></div>
</body>
```

Timing



- **Contoh (jam.js)**

```
function startTime()  
{  
    var today=new Date();  
    var h=today.getHours();  
    var m=today.getMinutes();  
    var s=today.getSeconds();  
    // Menambahkan angka 0 jika nilai <10  
    h=checkTime(h);  
    m=checkTime(m);  
    s=checkTime(s);  
    document.getElementById('jam').innerHTML=h+":"+m+":"+s;  
    t=setTimeout('startTime()',500);  
}
```

```
function checkTime(i)  
{  
    if (i<10)  
    {  
        i="0" + i;  
    }  
    return i;  
}
```

```
setTimeout("namaFungsi()", delay);
```

Fungsi setTimeout akan memanggil namaFungsi() bila waktu delay sudah terpenuhi.

Satuan waktu delay : milisecond.

1000ms = 1s

TERIMA KASIH

Pertemuan berikutnya membahas
HTML DOM

