

MODULE 5

JAVA SCRIPT DAN JQUERY

5.1 RINGKASAN

JavaScript merupakan bahasa pemrograman berbasis client-side yang berfungsi untuk menjadikan halaman web menjadi interaktif. Client-side berarti bahasa ini akan berjalan pada client dan tidak membutuhkan server tersendiri layaknya PHP. JavaScript sebenarnya tidak ada kaitannya sama sekali dengan Java. JavaScript dibuat pada tahun 1995 oleh Brendan Eich di Netscape dengan nama asli “Livescript”. Saat itu Java sangat populer dan untuk kebutuhan pemasaran, maka Livescript menjadi Javascript atau yang biasa disingkat JS.

Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru-buru dan tidak ada pesan kesalahan yang di tampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara **Netscape** dan **Sun** (pengembang bahasa pemrograman “Java”) pada masa itu, maka Netscape memberikan nama “*JavaScript*” kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan **Microsoft** sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai “*Jscript*” di browser Internet Explorer 3.

JavaScript bergantung kepada browser(navigator) yang memanggil halaman web yang berisi skrip skrip dari Javascript dan tentu saja terselip di dalam dokumen HTML. Javascript juga tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya (pada kenyataannya kompilator Javascript sendiri sudah termasuk di dalam browser tersebut). Lain halnya dengan bahasa “Java” (dengan mana JavaScript selalu di banding bandingkan) yang memerlukan kompilator khusus untuk menterjemahkannya di sisi user/klien.

Di JavaScript kita tidak mungkin menyembunyikan kode skrip yang kita tulis, kode langsung di tulis di dalam dokumen HTML dan sangat mudah terlihat, sedangkan di Java, kode sudah berbentuk setengah terkompilasi (dalam bentuk applet) dan tidak mungkin terlihat dari dalam dokumen HTML, satu mesin virtual di sisi user yang bertanggung jawab untuk menterjemahkan program di dalam applet tersebut setiap kali halaman HTML yang memuat applet tersebut dipanggil oleh browser. Dibandingkan dengan applet java yang cukup lambat dibuka oleh browser, bisa kita katakan bahwa Javascript cukup cepat di panggil (di load) oleh navigator.

- **Menambahkan JavaScript**

Layaknya CSS, anda bisa langsung menambahkan **javascript** langsung pada dokumen html

anda atau anda bisa juga menyimpannya sebagai file yang terpisah dan kemudian memanggilnya melalui link. Kedua metode tersebut sama-sama membutuhkan elemen `<script>`

- **Menempelkan Javascript**

Untuk menempelkan langsung JavaScript pada dokumen web anda, cukup anda tuliskan elemen `<script>` seperti contoh di bawah ini:

```
<script>
    ... kode javascript ada disini
</script>
```

- **Eksternal File**

Metode yang lain adalah dengan menggunakan atribut `src` yang diarahkan ke file JavaScript yang disimpan dengan format ekstensi `.js`. Contohnya seperti di bawah ini:

```
<scriptsrc="script_baru.js"></script>
```

Keuntungan menggunakan file javascript yang terpisah, adalah anda bisa menggunakan file yang sama pada halaman-halaman web yang lain, sama halnya dengan file css yang terpisah. Kekurangannya adalah saat pemanggilan file JavaScript tersebut membutuhkan penambahan permintaan ke server berbasis HTTP yang tentunya juga memperlambat performanya.

Catatan:

Pada versi HTML 4.01, penulisan tag `<script>` harus dilengkapi dengan atribut `type` agar bias dikenali.

```
<script type="text/javascript"> .. </script>
```

Pada HTML 5, penulisannya di persingkat menjadi `<script>` dengan catatan pada tiap awal halaman dituliskan `<!doctype html>` untuk menunjukkan halaman tersebut adalah **HTML5**.

- **Variabel**

Variable adalah suatu obyek yang berisi data data, yang mana dapat di modifikasi selama pengekseskusion program. Di JavaScript kita bisa memberikan nama variabel sepanjang yang kita suka, akan tetapi harus memenuhi kriteria berikut ini .

- Nama variabel harus dimulai oleh satu huruf (huruf besar maupun huruf kecil) atau satu karakter `"_"`.
- Nama variabel bisa terdiri dari huruf huruf, angka angka atau karakter `_` dan `&` (spasi kosong tidak diperbolehkan).

- Nama variabel tidak boleh memakai nama-nama berikut ini (reserved oleh program)
 - abstract
 - boolean break byte
 - case catch char class const continue
 - debugger default delete do double
 - else export extends
 - false final finally float for function
 - goto
 - if, implements, import, in, infinity, instanceof, int, interface
 - label, long
 - native, new, null
 - package, private, protected, public
 - return
 - short, static, super, switch, synchronized
 - this, throw, throws, transient, true, try, typeof

JavaScript menggunakan sistem case sensitive yang artinya membedakan nama variabel dengan huruf besar dan huruf kecil, oleh karena itu biasakanlah memberikan nama variabel dengan aturan yang sama, huruf besar semua atau huruf kecil semua (disarankan menggunakan selalu huruf kecil).

• Mendeklarasikan Variabel

Penulisan variabel JavaScript sangatlah fleksibel, dan tidaklah terlalu rumit dan ketat, sehingga kita tidaklah terlalu sering menerima pesan error pada saat menjalankan program. Sebagai contoh deklarasi variabel di JavaScript dapat kita lakukan dengan dua cara :

- eksplisit : dengan menuliskan kata kunci var kemudian diikuti dengan nama variabel dan nilai dari variabel :
`var test = "halo"`
- implisit : dengan menuliskan secara langsung nama dari variabel dan diikuti nilai dari variabel :
`test = "halo"`

Navigator secara otomatis akan memperlakukan pernyataan itu sebagai deklarasi dari sebuah variabel. Pada navigator versi lama mungkin terjadi kasus di mana navigator tidak mengenali

pendeklarasian variabel secara implisit, maka disarankan untuk menggunakan cara eksplisit dalam menulis program JavaScript.

JQUERY

jQuery adalah sebuah library JavaScript. Dalam dunia pemrograman, library adalah kumpulan dari berbagai fungsi ‘siap pakai’ untuk memudahkan pembuatan sebuah aplikasi, dengan demikian, jQuery adalah kumpulan fungsi-fungsi JavaScript yang memudahkan penulisan kode JavaScript. jQuery dikembangkan pertama kali oleh John Resig di tahun 2006. Sejak saat itu, jQuery berkembang menjadi proyek opensource dan menjadi library JavaScript paling populer di dunia. Situs resmi jQuery beralamat di jquery.com.

JavaScript sangat powerfull untuk memanipulasi element HTML, tapi penggunaannya juga tidak mudah dan ‘sedikit ribet’. Sebagai contoh, untuk mencari sebuah element HTML, dengan JavaScript kita menggunakan fungsi `getElementById()`:

```
var x = document.getElementById("tombol");
```

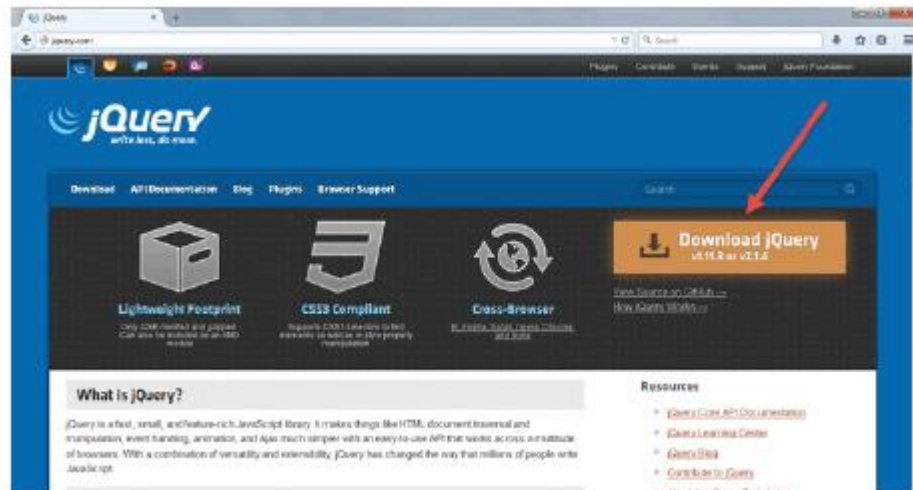
Dengan jQuery, penulisannya jauh lebih singkat:

```
var x = $("#tombol");
```

Selain itu (sebagaimana CSS) terdapat beberapa perbedaan implementasi JavaScript dari 1 web browser dengan web browser lain. Kode JavaScript yang berjalan mulus di Mozilla Firefox, belum tentu bisa berjalan di Internet Explorer, begitu juga sebaliknya. Hal ini diperparah dengan perbedaan versi dari masing-masing web browser. Untuk mengatasi hal ini, jQuery memiliki fungsi bawaan yang mendeteksi web browser dan menyediakan fungsi pengganti. Dengan demikian, kita tidak perlu pusing memikirkan perbedaan implementasi JavaScript dari setiap web browser. Fungsi JavaScript lanjutan seperti AJAX (Asynchronous JavaScript and XML), juga disederhanakan oleh jQuery. Untuk menggunakan AJAX, kita hanya butuh beberapa baris perintah. Fitur lain dari jQuery (dan mungkin menjadi alasan paling penting), jQuery menyediakan berbagai efek animasi menarik. Beberapa diantaranya sangat mudah digunakan dan hanya membutuhkan 1 atau 2 baris kode program. Khusus untuk membuat efek animasi ini, jQuery juga memiliki jQuery UI yang fokus ke animasi dan interaksi user. jQuery juga memiliki berbagai plugin atau kode tambahan untuk fungsi-fungsi lanjutan, seperti membuat efek slideshow, image carousel, hingga jQuery mobile. Efek-efek seperti ini dibuat berdasarkan kode dasar jQuery. Dengan berbagai fitur dalam jQuery, tidak heran jika jQuery sangat populer. Tidak jarang pula programmer pemula lebih mengenal jQuery daripada JavaScript.

Pada dasarnya, jQuery hanyalah sebuah file javascript external. Agar bisa menggunakan jQuery, kita harus download file jQuery, kemudian menghubungkannya

menggunakan tag `<script>`. Terdapat 2 cara menginput file jQuery: mendownload dan mengakses jQuery secara lokal, atau menggunakan CDN (*Content Delivery Network*). Untuk menggunakan jQuery secara lokal (offline), kita harus mendownload file jQuery, kemudian mengaksesnya melalui tag `<script>`. jquery.com. Pada bagian kanan tengah, terdapat tombol “Download jQuery”. Klik tombol ini. Kita akan dibawa ke halaman jquery.com/download.



Pada bagian ini anda bisa melihat 2 versi dan 2 jenis file jQuery. memilih antara versi *compressed* and *uncompressed*. Kedua file ini sebenarnya memiliki kode jQuery yang sama. Perbedaannya, versi compressed berukuran lebih kecil karena seluruh karakter yang tidak perlu, telah di hapus (seperti spasi dan baris komentar). Hasilnya, ukuran file jQuery versi compressed menjadi lebih kecil, tapi tidak bisa dibaca. Jika anda ingin mempelajari kode program yang ada di dalam jQuery (atau ingin mengembangkannya), pilih versi uncompressed. Anda bisa mendownload sendiri kedua versi ini untuk melihat perbedaannya.

Untuk menghubungkan file jQuery dengan HTML, kita cukup menambahkan tag `<script>` dengan alamat file `jquery-2.1.4.js`, seperti berikut ini:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Belajar jQuery </title>
6    <script src="jquery-2.1.4.js"></script>
7  </head>
8  <body>
9  </body>
10 </html>

```

Kode HTML diatas harus ditempatkan di dalam 1 folder dengan file. Apabila anda sudah mempelajari HTML dan CSS, tentu tidak asing dengan maksud penempatan file ini.

- **Ready Event**

Event `ready()` jQuery men-simulasi efek dari event `Onload` JavaScript, tapi lebih efisien. Dengan menggunakan fungsi `ready()` jQuery, kode program JavaScript baru akan dijalankan setelah web browser selesai memproses tag HTML, tapi sebelum file external

seperti gambar. Dengan demikian, kode JavaScript diproses dengan lebih awal daripada event Onload bawaan JavaScript. Penulisan fungsi ready() jQuery adalah sebagai berikut :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery</title>
<script src="jquery-2.1.4.js"></script>
<script>
$(document).ready(function() {
    $("#tombol").click(function() {
        $(this).after('<p>Sedang Belajar jQuery</p>');
    });
});
</script>
</head>
<body>
<h1>Belajar jQuery</h1>
<button id="tombol">Click Me!</button>
</body>
</html>
```

- **Object, Selector, Action**

Untuk bisa melakukan ‘sesuatu’ terhadap element HTML, kita tinggal menyambung penulisan jQuery Object dengan jQuery Action, atau dikenal juga dengan jQuery Event.

Format dasarnya adalah sebagai berikut:

`$(selector).action()`

Selector digunakan untuk mencari element mana dari HTML yang akan dimanipulasi.

Bisa berupa sebuah paragraf <p>, sebuah gambar , atau sesuatu yang lebih kompleks, seperti element <tr> pertama dari tabel kedua. Action adalah sesuatu yang bisa kita lakukan dengan element HTML tersebut, apakah menyembunyikannya, menampilkannya, mengubah warna, menambahkan element baru, dll. Action mirip seperti Event dalam JavaScript, tetapi dengan penambahan berbagai fitur lain. Di dalam jQuery, sebuah action bisa disambung dengan action lain, atau dikenal dengan istilah chaining. Berikut contoh penulisan jQuery Object, selector dan action :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery</title>
<script src="jquery-3.1.1.js"></script>
<script>
$( document ).ready(function() {
    $( "#tombol_hide" ).click(function() {
        $("h1").hide();
    });

    $( "#tombol_show" ).click(function() {
        $("h1").show();
    });

    $( "#tombol_small" ).click(function() {
        $("h1").css("font-size","20px");
    });

    $( "#tombol_red" ).click(function() {
        $("h1").css( "color", "red" );
    });
});
</script>
</head>
<body>
<h1>Belajar jQuery</h1>
<button id="tombol_hide">Sembunyikan</button>
<button id="tombol_show">Tampilkan</button>
<button id="tombol_small">Perkecil</button>
<button id="tombol_red">Merahkan</button>
</body>
</html>
```

- **Selector**

jQuery memudahkan proses pencarian element HTML dengan menyediakan jQuery Selector. jQuery Selector adalah sebuah cara yang disediakan jQuery untuk mencari element HTML. Tidak hanya dengan atribut id saja, kita juga bisa menggunakan class, atribut lain, dan kombinasi diantaranya. Jika anda sudah pernah menggunakan CSS, jQuery meminjam cara penulisan CSS Selector sebagai jQuery Selector. Dengan jQuery, untuk mencari sebuah element HTML yang memiliki id="tombol", penulisannya jauh lebih singkat: Karena sama seperti selector CSS, jQuery selector mendukung hampir semua penulisan selector dari CSS, berikut contohnya:

- Mencari seluruh element dengan tag tertentu. Misalkan kita ingin mencari seluruh tag <p>, penulisan jQuery Selectornya adalah: \$("p").
- Mencari element dengan id tertentu. Misalkan ingin mencari sebuah tag HTML yang memiliki id="belajar", maka kita bisa menulis: \$("#belajar").
- Mencari element dengan class tertentu. Misalkan kita ingin mencari seluruh tag HTML yang menggunakan atribut class="warna", maka bisa ditulis dengan: \$(".warna").

Sama seperti CSS, kita juga bisa mengkombinasikan selector untuk mencari element yang lebih spesifik, misalnya untuk mencari tag <p> yang memiliki class="warna" dan berada di dalam tag <div>, selectornya adalah: \$("div p.warna").

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Belajar jQuery</title>
<script src="jquery-3.1.1.js"></script>

<script>
$( document ).ready(function() {
    $("#tombol").click(function() {
        $("p").css( "color", "red" );
        $("#belajar").css( "color", "green" );
        $(".warna").css( "color", "blue" );
        $("#test.saja").css( "color", "yellow" );
        $("div p.warna").css( "color", "pink" );
    });
});
</script>

</head>
<body>
```

```
<p>
    Sebuah paragraf tanpa atribut apapun
</p>
<p id="belajar">
    Sebuah paragraf dengan id = "belajar"
</p>
<p class="warna">
    Sebuah paragraf dengan class = "warna"
</p>
<p id="test" class="saja">
    Sebuah paragraf dengan id="test" class="saja"
</p>
<div>
    <p class="warna">
        Sebuah paragraf tanpa atribut apapun
    </p>
</div>
<button id="tombol">Action</button>
</body>
</html>
```

- **Mengambil Nilai HTML**

- Metode `text()`: Metode pertama untuk mengambil nilai sebuah elemen HTML dengan jQuery adalah menggunakan method `text()`. Caranya cukup menambahkan method/fungsi `text()` ke jQuery Selector.
- Metode `html()`: Cara berikutnya untuk mengambil nilai HTML dengan jQuery adalah menggunakan method `html()`. Penggunaannya sama persis dengan method `text()`. Bedanya, hasil dari method `html()` akan tetap mempertahankan tag HTML yang ada di dalam element HTML tersebut.
- Metode `val()`: Baik method `text()` maupun `html()` sudah mencukupi untuk mengambil nilai yang ada disetiap element HTML. Tapi khusus untuk form, jQuery menyediakan method `val()`.

5.2 LATIHAN

Bagian ini mencakup penggunaan dasar tentang JavaScript dan jQuery seperti menampilkan text, menampilkan form HTML, komentar dan sebagainya. Simpanlah semua file-file dalam folder tersendiri, cobalah dan pahami serta eksplorasi latihan-latihan yang diberikan pada modul ini.

LEMBAR KERJA PRAKTIKUM PEMROGRAMAN WEB I

Dosen Pengampu:

1. Nurul Fathanah Mustamin, S.Pd., M.T.
2. Arief Trisno Eko Suryo, S.T.,M.T

Modul 5. JavaScript dan jQuery

Tujuan Praktikum:

1. Mahasiswa mampu memahami konsep dasar JavaScript dan jQuery
2. Mahasiswa mampu memahami struktur dasar syntax JavaScript dan jQuery

Soal Praktikum:

Ketikkan setiap kelompok tag dalam sebuah file, simpan dalam folder Anda sendiri yang telah ditentukan, kemudian jalankan di web browser yang telah terinstall.

1. Menulis ke halaman web melalui JavaScript

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4      <p>
5          Javascript bisa langsung menuliskan ke halaman web:
6      </p>
7
8      <script>
9          document.write("<h1>Ini contoh header</h1>");
10         document.write("<p>ini bagian paragraf</p>");
11     </script>
12
13     <p>
14         Anda bisa menggunakan <strong>document.write</strong>
15         untuk menulis pada halaman web melalui javascript
16     </p>
17
18 </body>
19 </html>

```

2. Merespon Event

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4      <h1>Latihan JavaScript</h1>
5
6      <p id="demo">
7          Javascript bisa merubah isi elemen HTML
8      </p>
9
10     <script>
11         function myFunction()
12         {
13
14             x=document.getElementById("demo"); // temukan elemen yg mana
15             x.innerHTML="Ini isi terbaru!"; // rubah isinya
16         }
17     </script>
18
19     <button type="button" onclick="myFunction()" ">Click Me!</button>
20
21 </body>
22 </html>

```

3. Merubah style Elemen HTML melalui JavaScript

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5      <h1>Latihan JavaScript</h1>
6
7      <p id="demo">
8          Javascript bisa merubah isi elemen HTML
9      </p>
10
11     <script>
12         function myFunction()
13         {
14
15             x=document.getElementById("demo"); // temukan elemen yg mana
16             x.style.color="#f8dfca"; // rubah isinya
17         }
18     </script>
19
20     <button type="button" onclick="myFunction()">Click Me!</button>
21
22 </body>
23 </html>

```

4. Pengenalan variabel dan aritmatika sederhana

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5     <script>
6         var x=5;
7         var y=6;
8         var z=5+6;
9
10        document.write("Variabel X: "+x + "<br>");
11        document.write("Variabel Y: "+y + "<br>");
12        document.write("Hasil X + Y: "+z + "<br>");
13        document.write("Hasil X * Y: "+ x*y + "<br>");
14        document.write("Hasil X / Y: "+ x/y + "<br>");
15    </script>
16
17 </body>
18 </html>

```

5. Tipe Data

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5      <script>
6          var pi=3.14;
7          var nama="John Doe";
8          var jawaban='Yes I am!';
9
10         document.write(pi + "<br>") ;
11         document.write(nama + "<br>") ;
12         document.write(jawaban + "<br>") ;
13     </script>
14
15 </body>
16 </html>

```

6. Merespon Event dan memberikan informasi berupa alert

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4
5      <h1>Latihan JavaScript</h1>
6
7      <p>
8          JavaScript merespon event yang terjadi, misalnya klik button:
9      </p>
10
11     <button type="button" onclick="alert('Selamat Datang!\n untuk anda')">
12         Click Me!</button>
13
14 </body>
15 </html>

```

7. Memanggil Fungsi

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <script type="text/javascript">
5      function fungsihello(txt)
6      {
7          alert(txt)
8      }
9  </script>
10 </head>
11 <body>
12 <form>
13     <input type="button"
14         onclick="fungsihello ('Hello')"
15         value="panggil fungsi Hello">
16 </form>
17 <p>Tekan tombol di atas, dan sebuah fungsi akan dipanggil</p>
18 </body>
19 </html>

```

8. Menampilkan kotak konfirmasi

```

1  <html>
2  <body>
3
4  <p>Klik tombol berikut untuk menampilkan konfirmasi</p>
5
6  <button onclick="myFunction()">Klik Disini</button>
7
8  <p id="demo"></p>
9
10 <script>
11     function myFunction()
12     {
13         var x;
14         var r=confirm("Apakah anda ingin melanjutkan?");
15         if (r==true)
16         {
17             x="Anda menekan OK!";
18         }
19         elsa
20         {
21             x="Anda menekan cancel!";
22         }
23         document.getElementById("demo").innerHTML=x;
24     }
25 </script>
26
27 </body>
28 </html>

```

9. Menampilkan prompt box

```

1 <html>
2 <body>
3
4 <p>Klik tombol berikut untuk menampilkan konfirmasi</p>
5
6 <button onclick="myFunction()">Klik Disini</button>
7
8 <p id="demo"></p>
9
10 <script>
11     function myFunction()
12     {
13         var x;
14
15         var name=prompt ("Silahkan masukkan nama anda","Galuh Meidaluna");
16
17         if (name!=null)
18         {
19             x="Selamat Datang ! " + name + " bagaimana kabar anda hari ini?";
20             document.getElementById("demo").innerHTML=x;
21         }
22     }
23 </script>
24
25 </body>
26 </html>

```

10. Mengatur Interval

```

1 <html>
2 <body>
3
4 <p>Klik tombol berikut, dan sbuah alert akan muncul 3 detik</p>
5 <p>Setelh anda klik, maka tiap 3 detik alert akan muncul</p>
6 <button onclick="myFunction()">Tampilkan alert</button>
7
8
9 <script>
10     function myFunction()
11     {
12         setInterval(function() {alert("Selamat Datang")},3000);
13     }
14 </script>
15
16 </body>
17 </html>

```

11. Validasi Input

```

1 <html>
2 <body>
3
4 <h1>Latihan Javascript</h1>
5
6 <p>Silahkan masukkan angka</p>
7
8 <input id="demo" type="text">
9
10 <script>
11     function myFunction()
12     {
13         var x=document.getElementById("demo").value;
14         if(x=="" || isNaN(x))
15         {
16             alert("Bukan angka");
17         }
18     }
19 </script>
20
21 <button type="button" onclick="myFunction()">Cek Angka</button>
22
23 </body>
24 </html>

```