

MATERI V - FUNGSI

TUJUAN:

Mahasiswa mengerti mengenai penggunaan fungsi dan cara pembuatan fungsi di Javascript.

MATERI:

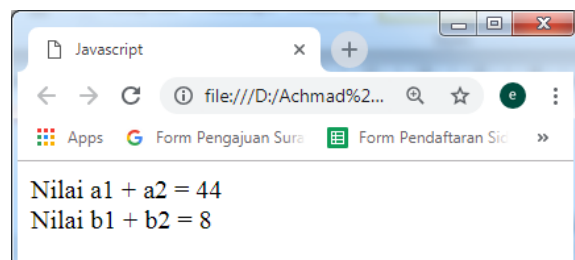
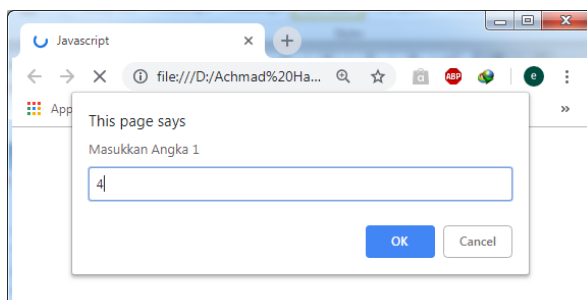
- Mengenal jenis-jenis fungsi Javascript
- Menggunakan fungsi built-in Javascript
- Membuat fungsi buatan sendiri
- Membuat fungsi dengan parameter
- Membuat fungsi yang menghasilkan nilai
- Variabel Global dan Variabel Lokal

LATIHAN PRAKTIKUM IV

1. Pada Javascript terdapat 2 jenis fungsi, yaitu fungsi built-in atau fungsi bawaan dari Javascript dimana tinggal dipergunakan saja dan fungsi buatan sendiri. Fungsi buatan sendiri dipergunakan untuk mempermudah pengerjaan perintah yang berulang-ulang, sehingga fungsi buatan sendiri harus didefinisikan terlebih dahulu agar dapat dikenali oleh Javascript.
2. Salah satu contoh fungsi built-in Javascript adalah fungsi *document.write* yang telah banyak dipergunakan sejak awal. Ada banyak fungsi-fungsi built-in Javascript lainnya, seperti fungsi untuk jam dan tanggal, fungsi matematika, fungsi string, dan sebagainya.
3. Selain fungsi *document.write*, fungsi built-in Javascript yang penting adalah *parseInt*, fungsi *parseInt* dipergunakan untuk melakukan konversi dari variabel string atau kata menjadi variabel dengan nilai integer atau angka, seperti contoh di bawah ini. Perbandingkan antara hasil penjumlahan tanpa menggunakan *parseInt* dengan yang menggunakan *parseInt*:

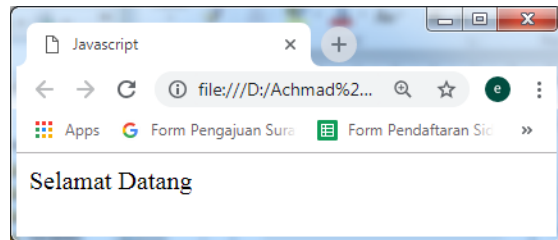
Nama File: **fungsi.htm**

```
7 <body>
8 <script type="text/javascript">
9     var a1,a2;
10    var b1,b2,c;
11
12    a1=prompt("Masukkan Angka 1","");
13    a2=prompt("Masukkan Angka 2","");
14    c=a1+a2;
15    document.write("Nilai a1 + a2 = "+c+"<br>");
16
17    b1=parseInt(a1);
18    b2=parseInt(a2);
19    c=b1+b2;
20    document.write("Nilai b1 + b2 = "+c+"<br>");
21 </script>
22 </body>
```



4. Untuk membuat fungsi sendiri dapat dipergunakan perintah function, seperti contoh di bawah ini, dimana dibuat fungsi dengan nama tulis daripada berulang kali menuliskan document.write.

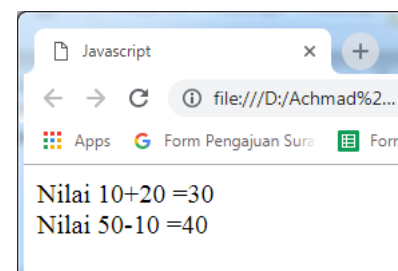
```
<body>
<script type="text/javascript">
    function tulis(kata){
        document.write(kata);
    }
    tulis("Selamat Datang");
</script>
</body>
```



5. Dari contoh di atas dapat dilihat bahwa fungsi dibuat dengan nama "tulis" sehingga untuk menulis ke layar tidak perlu menulis perintah document.write yang lebih panjang, kemudian diperlukan sebuah parameter (nilai yang dimasukkan ke dalam fungsi) melalui variabel dengan nama "kata", kemudian untuk menjalankan fungsi dapat dilakukan dengan menggunakan nama fungsi tersebut diikuti dengan parameter yang diperlukan seperti: tulis("Selamat Datang");
6. Selain dapat mengerjakan banyak perintah sekaligus, sebuah fungsi juga berguna untuk mengembalikan suatu nilai dari hasil perhitungan formula dalam fungsi tersebut. Fungsi built-in Javascript yang merupakan fungsi yang mengembalikan nilai adalah fungsi prompt dan fungsi parseInt, yang sudah sering dipergunakan. Sebagai contoh script buatan sendiri yang dapat mengembalikan nilai maka dibuat 2 buah fungsi baru yaitu fungsi "tambah" dan fungsi "kurang", yang akan melakukan perhitungan nilai parameter ke-1 ditambah atau dikurangi nilai parameter ke-2.

Nama File: **fungsi2.htm**

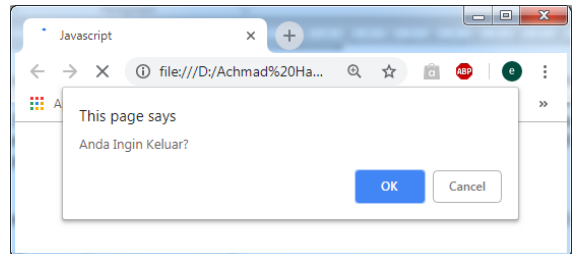
```
7 <body>
8 <script type="text/javascript">
9     var a,b,hasil;
10    function tambah(angka1,angka2) {
11        return (angka1+angka2);
12    }
13    function kurang(angka1,angka2) {
14        return (angka1-angka2);
15    }
16    hasil=tambah(10,20);
17    document.write("Nilai 10+20 =" +hasil+"<br>");
18    hasil=kurang(50,10);
19    document.write("Nilai 50-10 =" +hasil+"<br>");
20 </script>
21 </body>
```



7. Dari contoh di atas dapat dilihat bahwa untuk mengembalikan nilai dapat dipergunakan perintah return (nilai yang akan dikembalikan), dan parameter yang dimasukkan dapat lebih dari 1 parameter, sesuai dengan kebutuhan dari fungsi tersebut, sedangkan untuk melakukan pemanggilan terhadap fungsi yang mengembalikan nilai, diperlukan sebuah nilai variabel di depannya seperti contoh: hasil = tambah (10,20); dimana variabel "hasil" akan menampung nilai yang dikembalikan dari fungsi "tambah" yang merupakan hasil perhitungan angka 10 ditambah angka 20.

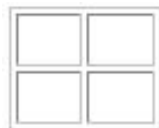
8. Dengan adanya penggunaan fungsi maka akan dikenal 2 jenis variabel, yaitu **variabel global** dan **variabel lokal**. Variabel global adalah variabel yang dikenal oleh seluruh fungsi-fungsi Javascript yang ada, sedangkan variabel lokal adalah variabel yang hanya dikenal oleh fungsi itu sendiri.
9. Pada contoh di atas, karena function "fungsi1" memiliki variabel lokal dengan nama "a" maka nilai "a" yang dipergunakan adalah nilai variabel lokal "a" yaitu 20, sedangkan pada function "fungsi2" karena tidak memiliki variabel lokal dengan nama "a" maka nilai variabel yang dipergunakan adalah nilai variabel global "a" yaitu 10. Sedangkan untuk perintah di luar fungsi sudah pasti akan dipergunakan nilai variabel global "a" yaitu 10.
10. Untuk menanyakan sesuatu dapat dipergunakan perintah confirm seperti tampak di bawah ini:

```
<script type="text/javascript">  
    jawab=confirm("Anda Ingin Keluar?");  
    document.write(jawab);  
</script>
```



TUGAS PRAKTIKUM

1. Buatlah sebuah fungsi yang akan menggambar tabel sesuai dengan parameter baris x kolom.
contoh: tabel(2,2) maka akan menggambar tabel sebagai berikut:



-- selamat mengerjakan--

PRAKTIKUM VI - DOM

TUJUAN:

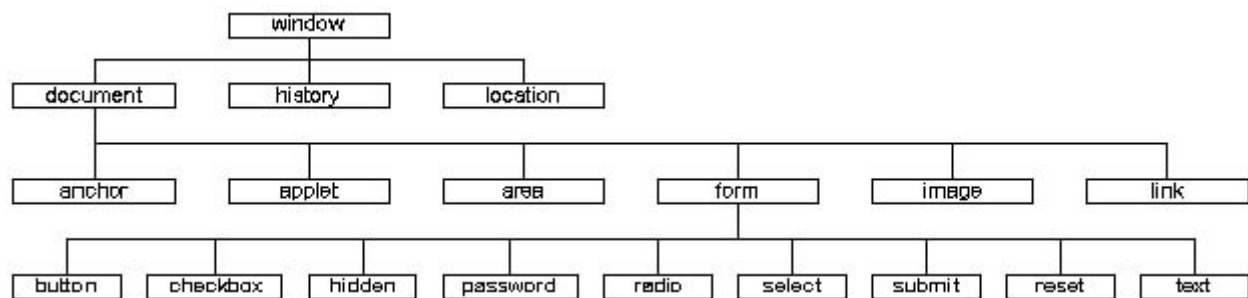
Mahasiswa mengerti tentang Document Object Model (DOM), hubungannya dengan Javascript dan cara penggunaan DOM dengan Javascript.

MATERI:

- Mengenal Document Object Model.
- Mengenal Objek
- Mengenal Event
- Mengenal Properti
- Memprogram Document Object Model.

TUGAS PRAKTIKUM V:

1. Javascript adalah merupakan bahasa pemrograman yang berorientasi objek, yaitu dimana segala hal yang dapat diprogram oleh Javascript didefinisikan sebagai sebuah objek. Ada beberapa objek yang dikenal oleh Javascript yaitu objek built-in yaitu objek bawaan dari Javascript sendiri misalnya objek date untuk tanggal dan waktu, objek math untuk operasi matematis, objek string untuk menangani operasi karakter pada sebuah kalimat, dan lain sebagainya.
2. Selain objek built-in ada juga objek buatan dari user dan objek browser. Objek browser inilah yang dikenal dengan istilah Document Object Model, yaitu sebuah browser seperti Internet Explorer atau Netscape itu sebenarnya terdiri dari beberapa komponen objek yang tersusun dalam sebuah struktur hirarki yang dapat digambarkan sebagai berikut:



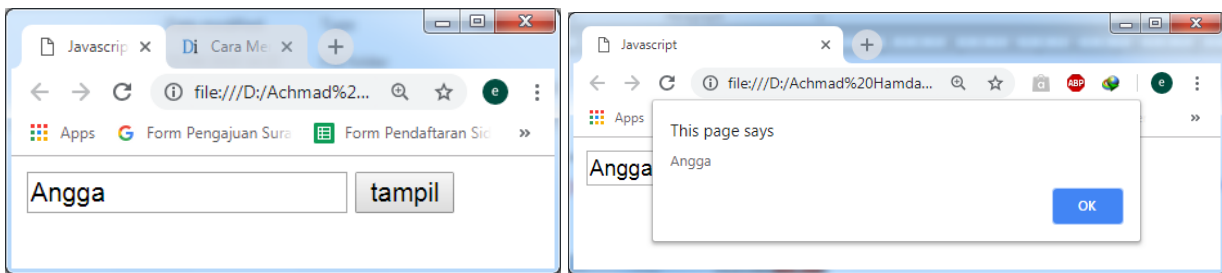
3. Dari diagram di atas dapat dilihat bahwa objek yang dimaksud adalah window, document, history, location dan navigator. Sedangkan objek-objek di bawah objek document disebut objek HTML, seperti anchor, applet, area, form, image dan link. Sedangkan objek-objek di bawah Form disebut objek Form seperti button, checkbox, hidden, password, radio, dan sebagainya.
4. Masing-masing objek dapat memiliki identitas atau nama, untuk membedakannya dari objek yang lain, serta masing-masing objek juga memiliki event atau kejadian yang dikenalnya, agar apabila ada suatu event yang terjadi maka objek tersebut dapat merespon dengan melakukan suatu perintah tertentu, seperti contoh di bawah ini:

Nama File: **latihanDOM.htm**

```

7 <body>
8 <script type="text/javascript">
9     function tampil() {
10         hasil=document.getElementById("teks").value;
11         alert(hasil);
12     }
13 </script>
14
15 <input id="teks" type="text" />
16 <input type="button" value="tampil" onclick="tampil()" />
17 </body>

```



5. Perhatikan pada contoh script di atas, pada tag `<SCRIPT>...</SCRIPT>` hanya didefinisikan sebuah fungsi buatan dengan nama "tampil", fungsi tersebut akan menampilkan kotak pesan dengan isi pesan sesuai dengan isi dari komponen objek dengan id "teks" dan properti "value". Kemudian dibuat sebuah objek Form berupa textbox dengan nama "teks", setelah itu buat lagi sebuah objek tombol atau button dengan tulisan "Tampil" dan tombol tersebut memiliki sebuah event yaitu "OnClick" dengan nilai "OnClick = tampil()", jadi pada saat tombol tersebut diklik maka akan dipanggil fungsi "tampil" yang telah didefinisikan sebelumnya.

6. Sebuah contoh yang lain adalah seperti tampak di bawah ini:

Nama File: **latihanDOM2.htm**

```

<body>
<script type="text/javascript">
    function tambah(a,b) {
        a=parseInt(a);
        b=parseInt(b);

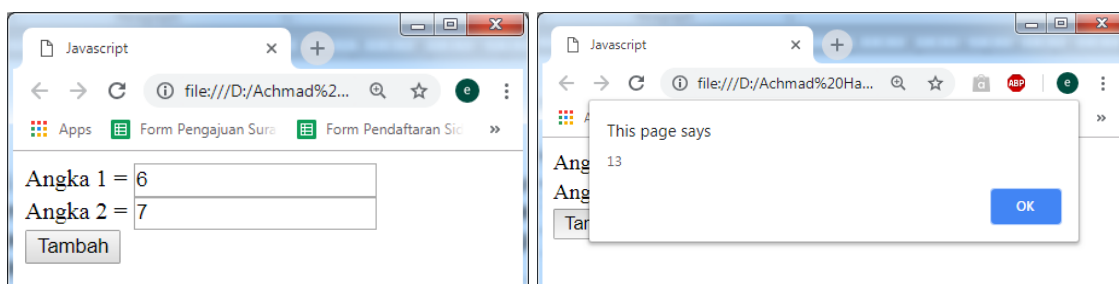
        alert(a+b);
    }
</script>

```

```

Angka 1 = <input type="text" id="angka1" /><br />
Angka 2 = <input type="text" id="angka2" /><br />
<input type="button" value="Tambah" onclick="tambah(angka1.value, angka2.value)" />
</body>

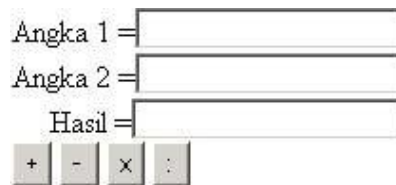
```



7. Dari contoh di atas dapat dilihat bahwa dibuat 2 buah textbox untuk memasukkan angka 1 dan angka 2, kemudian ada sebuah tombol dengan tulisan "Tambah" yang akan melakukan perhitungan penambahan dari nilai yang ada pada textbox dengan nama "angka1" dengan nilai yang ada pada textbox dengan nama "angka2".

TUGAS PRAKTIKUM

Lengkapilah program kalkulator sederhana di atas sehingga memiliki tampilan seperti di bawah ini:



The image shows a simple calculator interface. It consists of three input fields stacked vertically. The first field is labeled "Angka 1 =", the second "Angka 2 =", and the third "Hasil =". Below the input fields, there are four buttons arranged horizontally: a plus sign (+), a minus sign (-), a multiplication sign (x), and a division sign (:).

-- selamat mengerjakan--

PRAKTIKUM VII – OBJEK BUILT-IN

TUJUAN:

Mahasiswa mengenal objek-objek bawaan/built-in yang didukung oleh Javascript serta cara menggunakan metode-metode yang dimiliki oleh objek-objek tersebut.

MATERI:

- Pengenalan Metode
- Objek String
- Objek Math
- Objek Date

LATIHAN PRAKTIKUM

1. Objek di Javascript ada 3 jenis, yaitu: objek bawaan/built-in, objek buatan user, dan objek HTML. Objek bawaan/built-in adalah objek milik Javascript sendiri yang langsung dapat dipergunakan tanpa harus dibuat atau diprogram terlebih dahulu.
2. Ada 8 jenis objek bawaan yang dikenal oleh Javascript, yaitu: objek String, Number, Math, Date, Array, Function, Boolean, dan Screen. Tetapi objek yang paling banyak dipergunakan adalah objek String, Math dan Date.
3. Setiap dari objek akan memiliki properti dan metode, serta mengalami event / kejadian.
 - Properti : atribut yang dimiliki oleh setiap objek, misal: nama objek atau status objek.
 - Metode : perintah yang dikenal oleh suatu objek / kemampuan mengerjakan suatu tugas.
 - Event : kejadian yang dialami oleh suatu objek, misal: objek diklik dengan mouse.
4. Javascript merupakan bahasa pemrograman berbasis objek, yaitu dimana setiap perintah yang diberikan akan secara langsung berkaitan dengan salah satu jenis objek bawaan Javascript, seperti contoh di bawah ini:

```
var salam = "Halo Javascript";
```

perintah di atas dikenal sebagai perintah untuk membuat sebuah variabel String dengan nama "salam" dan diisi dengan kalimat "Halo Javascript", tetapi sebenarnya yang dikerjakan oleh Javascript adalah membuat sebuah objek bawaan String dengan nama "salam" dan kemudian diisi dengan kalimat "Halo Javascript", seperti contoh kode di bawah ini:

```
var salam = new String ("Halo Javascript");
```

perintah di atas merupakan perintah Javascript untuk membuat suatu objek baru.

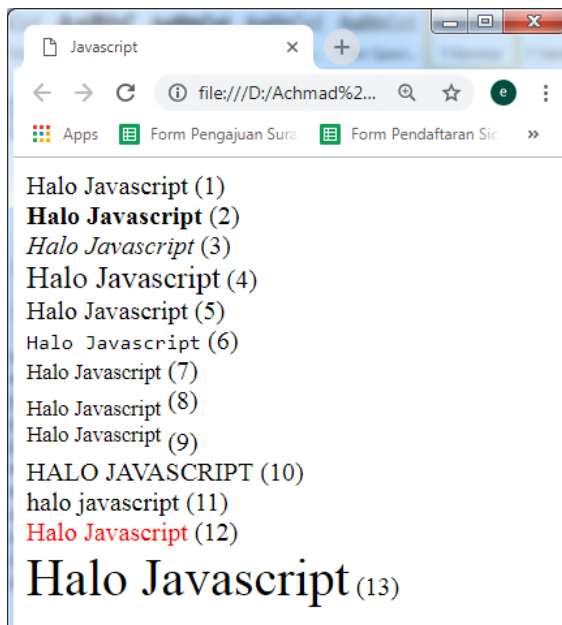
5. Objek String memiliki banyak metode-metode yang dapat dipergunakan untuk mengubah tampilan string tersebut di browser, seperti contoh di bawah ini:

Nama File: **objek.htm**

```

7  <body>
8  <script type="text/javascript">
9      var salam = new String("Halo Javascript");
10
11     document.write(salam + " (1)</br>");
12     document.write(salam.bold() + " (2)</br>");
13     document.write(salam.italics() + " (3)</br>");
14     document.write(salam.big() + " (4)</br>");
15     document.write(salam.blink() + " (5)</br>");
16     document.write(salam.fixed() + " (6)</br>");
17     document.write(salam.small() + " (7)</br>");
18     document.write(salam.sub() + " (8)</br>");
19     document.write(salam.sup() + " (9)</br>");
20     document.write(salam.toUpperCase() + " (10)</br>");
21     document.write(salam.toLowerCase() + " (11)</br>");
22     document.write(salam.fontcolor("red") + " (12)</br>");
23     document.write(salam.fontSize("6") + " (13)</br>");
24 </script>
25 </body>

```

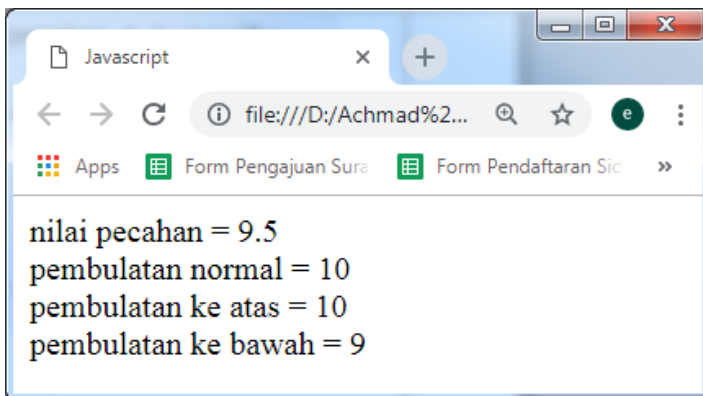


Contoh script di atas memperkenalkan metode `.bold()` dan metode `.italics()` yang berfungsi untuk mengubah tampilan dari objek string "salam" menjadi cetak tebal atau cetak miring di browser dengan cara penulisan, nama objek diikuti dengan metode yang hendak dijalankan seperti: `salam.bold()` atau `salam.italics()`

6. Objek Math dipergunakan untuk melakukan perhitungan matematis, seperti mendapatkan nilai pembulatan dan sebagainya, objek Math tidak dapat dibuat dengan fungsi `new` seperti objek String tetapi objek Math tersedia untuk dapat dipergunakan bagi objek-objek yang lain, seperti contoh di bawah ini:

Nama File: **ObjekMath.htm**

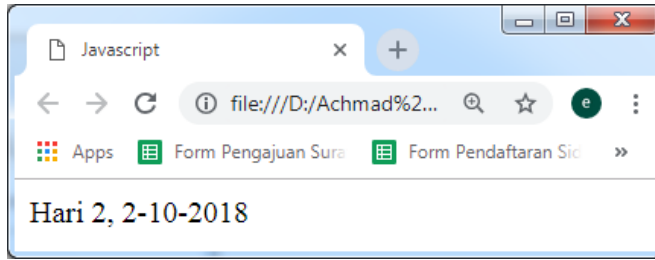
```
7 <body>
8 <script type="text/javascript">
9     var pecahan = "9.5";
10    var bulat;
11
12    document.write("nilai pecahan = " + pecahan + "<br>");
13
14    bulat=Math.round(pecahan);
15    document.write("pembulatan normal = " + bulat + "<br>");
16
17    bulat=Math.ceil(pecahan);
18    document.write("pembulatan ke atas = " + bulat + "<br>");
19
20    bulat=Math.floor(pecahan);
21    document.write("pembulatan ke bawah = " + bulat + "<br>");
22
23 </script>
24 </body>
```



7. Objek Date dipergunakan untuk menangani fungsi-fungsi tanggal dan waktu, sebagai contoh untuk mengambil tanggal sekarang dapat dipergunakan fasilitas `getDay` (hari), `getDate` (tanggal), `getMonth` (bulan), dan `getFullYear` (tahun), seperti contoh script di bawah ini:

Nama File: **ObjekDate.htm**

```
7 <body>
8 <script type="text/javascript">
9     var tgl = new Date();
10    var tanggal = "";
11
12    tanggal=tanggal + "Hari " + tgl.getDay() + ", ";
13    tanggal=tanggal + tgl.getDate() + "-";
14    tanggal=tanggal + (tgl.getMonth()+1) + "-";
15    tanggal=tanggal + tgl.getFullYear();
16
17    document.write(tanggal);
18 </script>
19 </body>
```



8. Perhatikan bahwa hasil yang didapat untuk `getDay` dan `getMonth` masih dalam bentuk angka jadi untuk mengubahnya menjadi nama hari atau nama bulan harus dilakukan sendiri, sedangkan pada `getMonth` ditambah dengan 1, hal ini dikarenakan bulan dimulai dari angka 0 untuk Januari. Hal ini menyebabkan kesalahan angka pada bulan yang ditampilkan sehingga harus ditambahkan 1 agar bulan Januari = angka 1, dan terus sampai Desember = angka 12.

TUGAS PRAKTIKUM

1. Perbaiki fungsi untuk menampilkan tanggal di atas sehingga dapat menampilkan tanggal dalam format bahasa Indonesia, seperti: Jumat, 4 Maret 2005.
2. Buatlah sebuah tampilan script seperti di bawah ini, dimana pada waktu tombol "Cetak" ditekan maka script akan menuliskan teks yang terdapat pada textbox dengan warna dan ukuran yang sesuai dengan pilihan pada combobox Color dan Size.

Text: Color: Size:

-- selamat mengerjakan--

PRAKTIKUM VIII – VALIDASI FORM

TUJUAN:

Mahasiswa mengerti tentang validasi form dan dapat membuat Javascript untuk melakukan validasi form.

MATERI:

- Pengertian Validasi Form
- Validasi Jenis Data
- Validasi String
- Validasi Checkbox
- Validasi Radiobutton
- Validasi Combobox
- Validasi Textarea

LATIHAN PRAKTIKUM

1. Javascript sebagai bahasa pemrograman yang berjalan pada sisi klien berfungsi untuk mengurangi beban server, salah satu komponen pemrograman sisi server yang dapat diambil alih fungsinya oleh sisi klien adalah validasi form. Yang dimaksud dengan validasi form adalah pemeriksaan yang dilakukan terhadap input yang dilakukan user pada sebuah form HTML. Daripada pengecekan nilai tersebut dilakukan oleh PHP atau ASP di server, lebih efisien bagi server apabila pemeriksaan tersebut apabila dilakukan di sisi klien yaitu dengan menggunakan Javascript, sehingga program PHP atau ASP yang ada di server tinggal memproses lebih lanjut nilai yang sudah pasti benar.
2. Validasi input form yang paling dasar adalah penentuan input user tersebut berupa kata atau angka (dalam istilah pemrograman, data tersebut bertipe String atau Numerik). Tetapi dalam Javascript tidak ada fasilitas atau fungsi built-in yang dapat dipergunakan untuk melakukan validasi jenis data dari input tersebut, oleh karena itu harus dibuat sebuah fungsi buatan agar dapat melakukan fungsi pendeteksian jenis data tersebut.

```
function iniAngka(nilai){  
    var test;  
    test = parseInt(nilai);  
    if(test != nilai){  
        return(false);  
    }else{  
        return(true);  
    }  
}
```

dengan menggunakan fungsi di atas dapat dilakukan pemeriksaan terhadap nilai suatu variabel, apabila nilai variabel tersebut berupa Angka (Numerik) maka akan dikembalikan nilai true (benar) dan apabila nilai variabel tersebut berupa Kata (String) maka akan dikembalikan nilai false (salah).

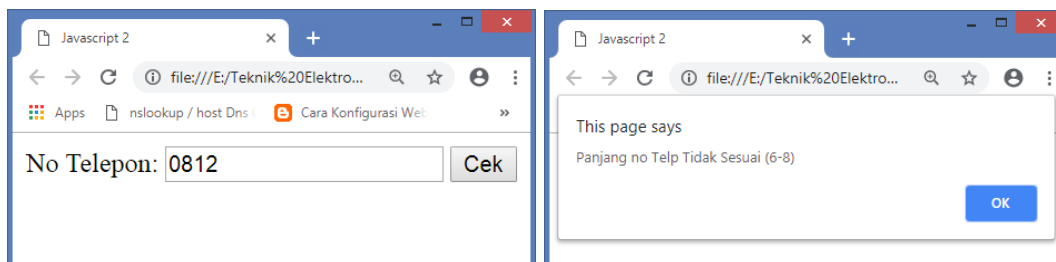
3. Apabila telah diketahui bahwa input user adalah berjenis String maka dapat diperiksa lagi apakah panjang String atau kata yang dimasukkan sesuai dengan yang diharapkan, misal untuk input nomer telepon, pasti diharapkan panjang 6-8 karakter, apabila kurang atau lebih pasti nomer telepon tersebut tidak valid. Untuk melakukan pemeriksaan panjang input user dapat dilakukan dengan menggunakan properti **length** yang dimiliki tiap-tiap objek data String, seperti contoh di bawah ini:

Nama File: **ValidTelp.htm**

```

7 <body>
8 <script type="text/javascript">
9 function cekTelp() {
10     var no = "";
11     no=telp.value;
12
13     if((no.length < 6) || (no.length > 8)){
14         alert("Panjang no Telp Tidak Sesuai (6-8)");
15     }else{
16         document.write("No. Telepon: " + no);
17     }
18 }
19 </script>
20
21 No Telepon: <input type="text" id="telp"/>
22 <input type="submit" value="Cek" onclick="cekTelp()" />
23 </body>

```



- Setelah dapat melakukan validasi terhadap jenis data suatu input user, maka fungsi validasi diperluas ke arah pemeriksaan terhadap validasi String yang dilakukan oleh user, validasi string seringkali dilakukan terhadap input email user, karena dalam sebuah alamat email dapat didektesi ada beberapa karakter khusus yang dipergunakan agar sebuah alamat email itu valid atau benar. Seperti contoh alamat email berikut: **nama@yahoo.com** dari alamat email tersebut dapat dilihat bahwa ternyata ada 2 karakter khusus yang dipergunakan yaitu @ dan . dan apabila ke-2 tanda tersebut tidak ada maka alamat email tersebut tidak valid atau tidak benar.
- Untuk melakukan pemeriksaan terhadap karakter tertentu apakah ada di dalam suatu String atau Kata dapat dipergunakan fungsi **indexOf()**, seperti contoh di bawah ini:

Nama File: **ValidEmail.htm**

```

8 <script type="text/javascript">
9 function cekEmail() {
10     var alamat = "";
11     alamat = email.value;
12
13     if(alamat.indexOf("@") == -1){
14         alert("Tidak ada karakter @");
15     }
16     if(alamat.indexOf(".") == -1){
17         alert("Tidak ada karakter .");
18     }
19 }
20 </script>
21
22 Email: <input type="text" id="email" />
23 <input type="submit" value="Cek" onclick="cekEmail()" />

```

6. Setelah dapat melakukan pemeriksaan terhadap data String yang umumnya diinputkan dengan menggunakan komponen Form HTML Textbox, sekarang akan dilakukan pemeriksaan terhadap Checkbox, untuk menentukan apakah sebuah Checkbox dipilih atau tidak, seperti contoh di bawah ini dimana Javascript dapat mengetahui berapa banyak Checkbox yang dipilih.

Nama File: **ValidChexbox.htm**

```

7 <body>
8 <script type="text/javascript">
9 function cekHobi() {
10     var total = 0;
11     if(hobiOR.checked) {
12         total=total+1;
13     }
14     if(hobiBC.checked) {
15         total=total+1;
16     }
17     if(hobiTV.checked) {
18         total=total+1;
19     }
20     alert("Hobi yang dipilh sebanyak = "+total);
21 }
22 </script>
23
24 Hobi Anda: </br>
25 <input type="checkbox" id="hobiOR" />Olahraga</br>
26 <input type="checkbox" id="hobiBC" />Baca Buku</br>
27 <input type="checkbox" id="hobiTV" />Nonton TV</br>
28 <input type="submit" value="Cek" onclick="cekHobi()" />
29 </body>

```

7. Javascript juga dapat dipergunakan untuk memeriksa Radio. Komponen Radio merupakan salah satu komponen Form yang agak berbeda dari komponen Form yang lain, hal ini disebabkan karena Radio harus mempergunakan sebuah Nama yang sama sebagai identitas untuk beberapa Radio yang berbeda, sehingga pengaksesan dilakukan dengan menggunakan Index, seperti halnya pengaksesan pada objek Array. Pada contoh di bawah ini dapat dideteksi apabila user belum memasukkan input dan juga dapat menampilkan pilihan yang dipilih user.

Nama File: **ValidRadio.htm**

```

7 <body>
8 <script type="text/javascript">
9 function cekkel() {
10     if(rbkel[0].checked) {
11         alert("Anda memilih pria");
12     } else if(rbkel[1].checked) {
13         alert("Anda memilih wanita");
14     } else {
15         alert("Anda belum memilih");
16     }
17 }
18 </script>
19
20 Jenis Kelamin Anda: </br>
21 <input type="radio" id="rbkel" />Pria</br>
22 <input type="radio" id="rbkel" />Wanita</br>
23 <input type="submit" value="Cek" onclick="cekkel()" />
24 </body>

```

6. Komponen Combobox juga dapat diperiksa apakah sudah ada suatu nilai yang telah dipilih atau masih belum ada nilai yang dipilih, tetapi agar nilai tersebut dapat dideteksi jangan lupa untuk mengisikan atribut VALUE pada tag <OPTION> seperti contoh di bawah ini:

Nama File: **ValidCombo.htm**

```

7 <body>
8 <script type="text/javascript">
9 function cekNegara() {
10     if(negara.value == "ID"){
11         alert("Anda Memilih Indonesia");
12     }else if(negara.value == "MY"){
13         alert("Anda Memilih Malaysia");
14     }else if(negara.value == "AU"){
15         alert("Anda Memilih Australia");
16     }else{
17         alert("Anda Belum Memilih");
18     }
19 }
20 </script>
21
22 Negara Anda: <br>
23 <select id="negara">
24     <option value="">--Pilih Negara--</option>
25     <option value="ID">Indonesia</option>
26     <option value="MY">Malaysia</option>
27     <option value="AU">Australia</option>
28 </select>
29 <input type="submit" value="Cek" onclick="cekNegara()" />
30 </body>

```

7. Komponen Textarea juga dapat dideteksi isinya dengan menggunakan cara sama dengan Combobox atau Textbox, yaitu dengan memeriksa properti value dari komponen yang bersangkutan.
8. Dan supaya proses validasi dapat dilakukan secara otomatis tanpa harus melakukan penekanan tombol tertentu terlebih dahulu, hal ini dapat dilakukan dengan menggunakan event On-Submit pada tombol Submit, seperti contoh di bawah ini:

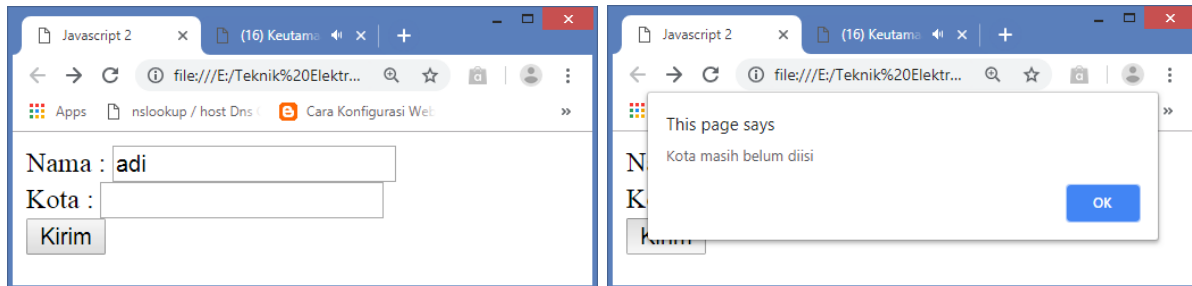
Nama File: **ValidasiForm.htm**

```

<body>
<script type="text/javascript">
function cekForm() {
    if(form1.nama.value == ""){
        alert("Nama masih belum diisi");
        form1.nama.focus();
    }else if(form1.kota.value == ""){
        alert("Kota masih belum diisi");
        form1.kota.focus();
    }else{
        document.form1.submit();
    }
}
</script>

```

```
<form name="form1" method="post" action="proses.php" onsubmit="cekForm()" >
Nama : <input type="text" id="nama" /><br>
Kota : <input type="text" id="kota" /><br>
<input type="button" value="Kirim" onclick="cekForm()" />
</form>
</body>
```



TUGAS PRAKTIKUM

Buatlah form dengan kode Javascript untuk melakukan pemeriksaan input user sebelum dikirim ke server dengan ketentuan sebagai berikut:

- Nama : teks harus berupa String dan harus diisi.
- Alamat : teks harus berupa String.
- Telepon : harus terdiri dari Angka, terdiri dari 6-8 karakter dan harus diisi.
- Kota : teks harus berupa String.
- Tgl lahir : terbuat dari 3 textbox: tgl (1-31), bulan (1-12), tahun (1990-2018).
- Pendidikan : Combobox (SD, SMP, SMA, D3, S1) dan harus diisi.
- Pengalaman : Textarea dan harus diisi.
- Tombol Kirim dan tombol Hapus.

-- selamat mengerjakan --

PRAKTIKUM IX – CSS & JAVASCRIPT

TUJUAN:

Mahasiswa mengerti tentang Cascading Style Sheet (CSS) dan dapat menggunakannya dengan Javascript.

MATERI:

- Interaksi Javascript dengan CSS
- Efek-efek Javascript dengan CSS

LATIHAN PRAKTIKUM

1. Cascading Style Sheet (CSS) adalah suatu sistem pemformatan efek untuk huruf, paragraf, tepian (border), dan lain sebagainya. CSS dibuat untuk mempermudah proses pemberian efek tampilan/desain pada komponen-komponen HTML.
2. Penggunaan CSS dan Javascript seringkali dipergunakan untuk membuat efek rollover pada hyperlink atau untuk menu, hal ini dapat dibuat dengan memanfaatkan event OnMouseOver dan OnMouseOut yang dimiliki oleh komponen-komponen HTML, seperti contoh di bawah ini:

Nama File: **CSSdanJS.htm**

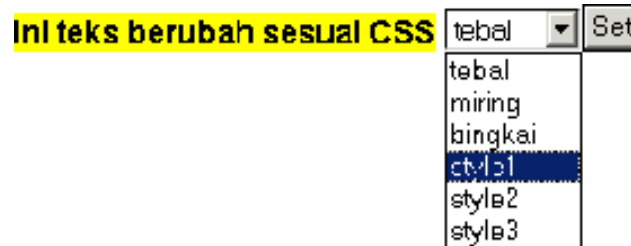
```
1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <title>CSS dan Javascript</title>
5      <style type="text/css">
6          .tebal{
7              font-family: arial;
8              font-size: 12px;
9              color: blue;
10             font-weight: bold;
11         }
12         .miring{
13             font-family: arial;
14             font-size: 12px;
15             color: red;
16             font-style: italic;
17         }
18     </style>
19 </head>
20
21 <body>
22 <script type="text/javascript">
23 function msOver(obj) {
24     obj.className = "miring";
25 }
26 function msOut(obj) {
27     obj.className = "tebal";
28 }
29 </script>
30
31 <a class="tebal" href="http://google.com"
32     onmouseover="msOver(this)" onmouseout="msOut(this)" >
33     Ini Link ke Google</a>
34 </body>
35 </html>
```


TUGAS PRAKTIKUM

- a. Buatlah rollover teks menu dengan menggunakan Javascript dan CSS background-color, seperti contoh di bawah ini:



- b. Buatlah combobox untuk mengganti style dari sebuah teks seperti contoh di bawah ini:



-- selamat mengerjakan --