

JAVASCRIPT

**disertai contoh-contoh javascript
yang kompatibel dengan Firefox**

by Ek kian

S U R A B A Y A - 2010

PRAKTIKUM I

TUJUAN:

Mahasiswa mengenal tentang Javascript serta dapat menuliskan kode-kode Javascript terutama untuk membuat teks di layar dan mendapatkan nilai dari user.

MATERI:

- Pengenalan Javascript
- Cara penulisan Javascript
- Tag <SCRIPT>...</SCRIPT>
- pengenalan tentang variabel
- document.write dan document.writeln
- prompt dan alert

TUGAS PRAKTIKUM I:

1. Javascript adalah bahasa pemrograman yang berjalan di sisi klien (klien web / browser, mis: Internet Explorer). Teknologi Javascript dibuat dengan tujuan agar dapat memperingan kerja server serta menambah sifat dinamis dan interaktivitas dari sebuah situs HTML. Penggunaan Javascript terutama untuk hal-hal yang tidak bersifat penting atau kritis, seperti pemeriksaan format input, animasi teks, efek kursor mouse, dan aplikasi-aplikasi ringan seperti kalkulator maupun games.
2. Teknologi Javascript pertama kali diperkenalkan oleh Netscape sejak Netscape 2.0 yang dapat menjalankan Javascript versi 1.0, kemudian Netscape 3.0 menggunakan Javascript versi 1.1 dan Netscape 4 ke atas menggunakan Javascript versi 1.2. Sedangkan untuk Internet Explorer melakukan implementasi script dengan menggunakan standar tersendiri yaitu VBscript serta Jscript yang kompatibel dengan Javascript, sehingga Javascript dapat berjalan pada IE tetapi VBscript tidak dapat berjalan di Netscape.
3. Penulisan Javascript pada HTML menggunakan tag <SCRIPT>...</SCRIPT> yang dapat ditempatkan pada area <HEAD> ataupun <BODY>. Penempatan tag <SCRIPT> pada area <HEAD> dimaksudkan agar Javascript dijalankan terlebih dahulu sebelum menampilkan halaman HTML, tetapi ada beberapa Javascript yang menggunakan elemen HTML justru harus ditulis pada area <BODY>.
4. Penulisan tag <SCRIPT> untuk Javascript adalah sebagai berikut:

```
<HTML>
  <HEAD>
    <TITLE> Javascript 1 </TITLE>
  </HEAD>

  <BODY>
    <SCRIPT LANGUAGE="Javascript">
      <!--
        //-->
      </SCRIPT>
    </BODY>
  </HTML>
```

6. Atribut "LANGUAGE" dipergunakan untuk memberitahukan bahwa bahasa script yang dipergunakan adalah Javascript dan bukanlah VBscript, sedangkan untuk menunjukkan versi dari Javascript dapat dipergunakan, mis: Javascript1.1 atau Javascript1.2, kalau Javascript saja diasumsikan menggunakan Javascript versi 1.0. Sedangkan tanda <!-- //--> dipergunakan untuk mencegah apabila ada browser yang tidak support tidak akan menampilkan kode Javascript tersebut sebagai kesalahan / error.
7. Perintah Javascript untuk menampilkan suatu teks atau tulisan pada layar browser dapat mempergunakan document.write atau document.writeln, seperti contoh di bawah ini:

```
<HTML>
  <HEAD>
    <TITLE> Javascript 1 </TITLE>
  </HEAD>

  <BODY>
    <SCRIPT LANGUAGE="Javascript">
      <!--
        document.write("Halo Javascript! ")
        document.write("Halo Javascript! ")
      //-->
    </SCRIPT>
  </BODY>
</HTML>
```

8. Perintah document.write juga dapat menampilkan seluruh tag-tag HTML termasuk tag untuk efek tulisan, seperti kode program di bawah ini:

```
<HTML>
  <HEAD>
    <TITLE> Javascript 1 </TITLE>
  </HEAD>

  <BODY>
    <SCRIPT LANGUAGE="Javascript">
      <!--
        document.writeln("<B>Halo Javascript!</B> <BR>")
        document.writeln("<FONT COLOR='BLUE'>Halo</FONT>")
      //-->
    </SCRIPT>
  </BODY>
</HTML>
```

9. Bahkan dapat juga untuk membuat tabel seperti tampak di bawah ini:

```
<BODY>
  <SCRIPT LANGUAGE="Javascript">
    <!--
      document.writeln("<TABLE BORDER='1'"")
      document.writeln("<TR>")
      document.writeln("<TD>1,1</TD> <TD>1,2</TD>")
      document.writeln("</TR>")
      document.writeln("<TR>")
      document.writeln("<TD>2,1</TD> <TD>2,2</TD>")
      document.writeln("</TR>")
    //-->
  </SCRIPT>
</BODY>
```

10. Pada dunia pemrograman salah satu konsep terpenting adalah VARIABEL, sama halnya pada bahasa pemrograman yang lain Javascript juga memerlukan penggunaan VARIABEL. Yang dimaksud dengan VARIABEL adalah sebuah penampung data dimana sewaktu-waktu isinya dapat diubah, ditambah, dikurangi, dan sebagainya. Bayangkan sebuah VARIABEL seperti halnya sebuah keranjang dimana keranjang tersebut dapat diisi dengan segala hal, kemudian bisa juga keranjang tersebut sudah ada isinya dan kemudian ditambahkan isinya atau dikurangi isinya, bahkan dikosongkan kembali seperti semula.
11. Isi VARIABEL pada Javascript dibedakan menjadi 2 yaitu angka atau kalimat. Isi variabel yang berupa angka seringkali disebut NUMERIK dan isi variabel yang berupa kalimat seringkali disebut STRING. Untuk membuat variabel dalam Javascript sangat mudah, yaitu tinggal disebutkan saja nama variabel yang akan digunakan kemudian isi dengan nilai yang dikehendaki, dengan perintah "var", seperti tampak di bawah ini:

```
<BODY>
  <SCRIPT LANGUAGE="Javascript">
    <!--
      var Nilai;
      var Nama1, Nama2;
      var Jumlah = 100;

      Nilai = 10;
      Nilai = 20;
      Nama1 = "Budi";
      Nama2 = "Iwan";

      document.write(Nilai + "<BR>");
      document.write(Nama1 + "<BR>");
      document.write(Nama2 + "<BR>");
      document.write(Jumlah + "<BR>");
    //-->
  </SCRIPT>
</BODY>
```

12. Perhatikan bahwa pada setiap baris perintah diakhiri dengan tanda ";" sebaiknya tanda ";" diberikan pada setiap akhir baris kode program, hal ini disebabkan karena standar Javascript yang berlaku sehingga lebih kompatibel dengan semua jenis browser yang ada.
13. Dari contoh di atas dapat dilihat bahwa variabel Nilai dan Jumlah berisi data Numerik sedangkan variabel Nama1 dan Nama2 berisi data String. Untuk menampilkan isi dari masing-masing variabel dipergunakan perintah document.write dan diisi dengan nama variabel yang hendak ditampilkan.
14. Nilai pada variabel dapat dikenai proses perhitungan matematis, seperti penambahan, pengurangan, pembagian maupun perkalian. Sehingga dengan kemampuan seperti itu maka dapat dibuat rumus-rumus matematis yang menggunakan variabel tersebut seperti tampak di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
  <!--
    var Panjang,Lebar,Luas;

    Panjang = 10;
    Lebar   = 20;

    Luas = Panjang * Lebar;

    document.write(Luas);
  //-->
</SCRIPT>
```

15. Coba ubahlah kode di atas untuk nilai Panjang dan Lebar, kemudian simpan dan lakukan Refresh pada Browser untuk melihat hasil yang ditampilkan. Proses matematis hanya dapat berfungsi apabila isi data dari variabel tersebut adalah Numerik sedangkan untuk jenis data String hanya dapat menggunakan proses penambahan saja, seperti terlihat di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
    var Lengkap,Status,Nama,Gelar;

    Status = "Bpk.";
    Nama    = "Budi";
    Gelar   = ",S.E.";

    Lengkap = Status + Nama + Gelar;

    document.write(Lengkap);
//-->
</SCRIPT>
```

16. Coba ubahlah kode di atas untuk nilai Status, Nama ataupun Gelar, kemudian simpan dan lakukan Refresh pada Browser untuk melihat hasil yang ditampilkan. Nilai data yang bersifat String selalu ditandai dengan tanda petik pembuka dan penutup seperti: "Jl. Jambu 20", sehingga apabila nilai data Numerik diberi tanda petik pembuka dan penutup akan diperlakukan sebagai String, seperti tampak di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
    var Nilai1,Nilai2,Jumlah;

    Nilai1 = "10";
    Nilai2 = "20";

    Jumlah = Nilai1 + Nilai2;

    document.write(Jumlah);
//-->
</SCRIPT>
```

17. Perhatikan bahwa hasil dari Jumlah tidak sesuai dengan yang diinginkan, hal ini disebabkan karena Nilai1 dan Nilai2 merupakan data String dan bukan data Numerik. Coba ubahlah nilai variabel Nilai1 dan Nilai2, perhatikan lagi perubahan yang terjadi pada nilai yang disimpan di variabel Jumlah.
18. Selain dapat menampilkan pada halaman browser, Javascript dapat juga menampilkan pesan pada kotak dialog tersendiri dengan mempergunakan perintah alert, seperti di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
    alert("Halo Javascript")
//-->
</SCRIPT>
```

19. Apabila dijalankan di browser, maka akan tampil pesan pada kotak dialog seperti di bawah ini:



20. Selain dapat menampilkan teks atau pesan dengan menggunakan kotak dialog, Javascript juga dapat meminta input atau masukan dari user dengan menggunakan kotak dialog. Perintah yang dipergunakan untuk meminta masukan dari user adalah prompt, seperti di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
    var nama;

    nama=prompt("Isi Nama Anda","");

    document.write(nama);
//-->
</SCRIPT>
```

21. Pada saat script tersebut dijalankan di Browser maka akan tampil kotak dialog seperti:



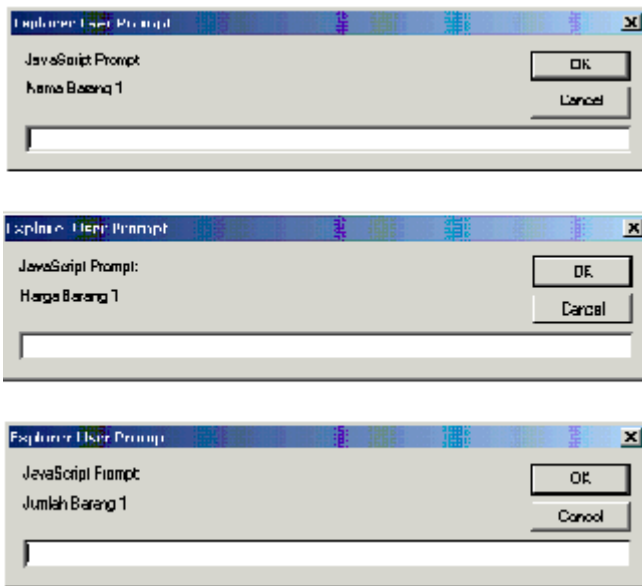
22. Dan input atau masukan dari user tersebut disimpan dalam variabel "nama" dan kemudian dicetak ke Browser dengan menggunakan perintah document.write.

Tugas Latihan:

a. Ubahlah kode program Javascript untuk perhitungan Luas, dengan menggunakan kotak dialog untuk memasukkan nilai Panjang dan Lebar, serta menggunakan kotak dialog untuk menampilkan hasil perhitungan Luas.

b. Buatlah kode program Javascript yang pada saat dijalankan akan menanyakan Nama Barang, Harga Barang, dan Jumlah Barang untuk 3 barang yang berbeda dan akan menampilkan hasil penjumlahan dari total harga yang harus dibayar dari ke-3 barang tersebut, seperti tampak di bawah ini:

Program akan menanyakan Nama Barang, Harga Barang dan Jumlah Barang:



The image shows three sequential JavaScript prompts in a browser window. The first prompt asks for 'Nama Barang 1', the second for 'Harga Barang 1', and the third for 'Jumlah Barang 1'. Each prompt has an 'OK' button and a 'Cancel' button.

Hasil perhitungan akan menampilkan tabel seperti tampak di bawah ini:

Nama	Harga	Jumlah	Sub Total
Kemeja	20000	3	60000
Celana	30000	2	60000
Sepatu	50000	1	50000
		TOTAL	170000

-- selamat mengerjakan --

PRAKTIKUM II

TUJUAN:

Mahasiswa mengerti dan dapat mempergunakan perintah untuk melakukan penyeleksian kondisi yang terdapat pada Javascript.

MATERI:

- If...Else...
- If...Else If...
- Switch
- == (sama dengan), <> (tidak sama dengan)
- > (lebih besar), < (lebih kecil), >= (lebih besar sama dengan), <= (lebih kecil sama dengan)
- && (AND), || (OR)

TUGAS PRAKTIKUM II:

1. Selain Variabel, komponen terpenting lainnya dari sebuah bahasa pemrograman, termasuk Javascript adalah kemampuan untuk melakukan penyeleksian kondisi terhadap nilai yang dimiliki oleh variabel. Dengan adanya kemampuan melakukan penyeleksian kondisi maka sebuah script dapat menentukan perintah atau kode program yang harus dijalankan sesuai dengan kondisi tertentu.
2. Perintah Javascript untuk melakukan penyeleksian kondisi dengan menggunakan If...Else... dengan cara penulisan (sintaks) sebagai berikut:

```
if (kondisi)
{
    // kode program akan yang dijalankan apabila kondisi terpenuhi
}
else
{
    // kode program yang akan dijalankan apabila kondisi TIDAK terpenuhi
}
```

3. Yang dimaksud dengan kondisi pada umumnya adalah ekspresi matematis atau pemeriksaan nilai suatu variabel sesuai dengan yang dikehendaki atau tidak. Pemeriksaan tersebut dapat dilakukan pada variabel Numerik maupun variabel String. Untuk melakukan pemeriksaan tersebut diperlukan sebuah operator, umumnya merupakan operator matematis, yaitu tanda == (sama dengan), <> (tidak sama dengan), > (lebih besar), < (lebih kecil), >=(lebih besar sama dengan), <=(lebih kecil sama dengan).
4. Untuk lebih jelasnya perhatikan contoh script di bawah ini yang melakukan pendeteksian nilai variabel nilai, apakah bernilai positif atau tidak:


```
<SCRIPT LANGUAGE="Javascript">
<!--
var nilai;

nilai=prompt("Isi Nilai","");

if (nilai > 0)
{
document.write("Nilai Positif");
}
else
{
document.write("Nilai bukan Positif");
}
}
//-->
</SCRIPT>
```

5. Penjelasan kode program di atas adalah sebagai berikut:

KODE PROGRAM	ARTI KODE
var nilai;	siapkan variabel dengan nama "nilai"
nilai=prompt("Isi Nilai","");	minta input dari user dan disimpan di variabel "nilai"
if (nilai > 0)	jika isi variabel "nilai" lebih besar dari 0 maka
document.write("Nilai Positif");	cetak ke browser tulisan "Nilai Positif"
else	selain itu
document.write("Nilai bukan Positif");	cetak ke browser tulisan "Nilai bukan Positif"

6. Selain dapat dipergunakan untuk melakukan pendeteksian terhadap variabel dengan nilai Numerik, perintah penyeleksi kondisi juga dapat dipergunakan untuk melakukan pemeriksaan terhadap nilai variabel dengan nilai String atau kalimat, seperti pada contoh script di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
var nama;

nama=prompt("Isi Nama","");

if (nama == "Budi wijaya")
{
document.write("Halo Budi wijaya");
}
else
{
document.write("Gak kenal tuh!");
}
}
//-->
</SCRIPT>
```

7. Perintah penyeleksi kondisi If...Else...dapat dikembangkan menjadi If...Else If...seperti contoh script berikut ini, yang merupakan modifikasi dari contoh pertama di atas:

```
<SCRIPT LANGUAGE="Javascript">
<!--
var nilai;

nilai=prompt("Isi Nilai","");

if (nilai > 0)
{
document.write("Nilai Positif");
}
else if (nilai == 0)
{
document.write("Nilai adalah 0");
}
else
{
document.write("Nilai Negatif");
}
}
//-->
</SCRIPT>
```

8. Kondisi yang diperiksa pada penyeleksi kondisi dapat lebih dari 1. Untuk melakukan pemeriksaan kondisi yang lebih daripada 1 dipergunakan operator && (AND) dan || (OR). Operator && dipergunakan apabila dikehendaki semua kondisi terpenuhi dan operator || dipergunakan apabila dikehendaki salah satu kondisi terpenuhi.

CONTOH	ARTI KONDISI
a > 0 && a < 10	apabila nilai a = 1 sampai 9 kondisi terpenuhi
a > 0 a < 10	apabila nilai a > 0 ATAU nilai a < 10 kondisi terpenuhi

9. Untuk lebih jelas perhatikan contoh script di bawah ini, kode script ini bertujuan untuk melakukan konversi atau pengubahan dari nilai menjadi grade, dengan ketentuan sebagai berikut:

nilai 90-100 : A
 nilai 70-89 : B
 nilai 60-69 : C
 nilai 50-59 : D
 nilai < 50 : E

```
<SCRIPT LANGUAGE="Javascript">
<!--
    var nilai;

    nilai=prompt("Isi Nilai","");

    if ((nilai <= 100) && (nilai >= 90))
    {
        document.write("Nilai A");
    }
    else if ((nilai <= 89) && (nilai >= 70))
    {
        document.write("Nilai B");
    }
    else if ((nilai <= 69) && (nilai >= 60))
    {
        document.write("Nilai C");
    }
    else if ((nilai <= 59) && (nilai >= 50))
    {
        document.write("Nilai D");
    }
    else if (nilai < 50)
    {
        document.write("Nilai E");
    }
//-->
</SCRIPT>
```

10. Selain menggunakan If...Else If... untuk melakukan penyeleksian kondisi dapat juga mempergunakan perintah **Switch**, seperti contoh di bawah ini, dimana program akan meminta input berupa angka dan nilai angka tersebut akan diubah menjadi nama angka tersebut.

```
<SCRIPT LANGUAGE="Javascript">
<!--
    var nilai;

    nilai=prompt("Isi Nilai","");

    switch(nilai)
    {
        case "0": document.write("Nol"); break;
        case "1": document.write("Satu"); break;
        case "2": document.write("Dua"); break;
        case "3": document.write("Tiga"); break;
        case "4": document.write("Empat"); break;
        case "5": document.write("Lima"); break;
        case "6": document.write("Enam"); break;
        case "7": document.write("Tujuh"); break;
        case "8": document.write("Delapan"); break;
        case "9": document.write("Sembilan"); break;
        default: document.write("Gak Tau")
    }
//-->
</SCRIPT>
```

Tugas Latihan

- a. Buatlah sebuah script yang mengambil input dari user berupa jam sekarang, misal: jam 9, jam 10 atau jam 13, kemudian berdasarkan jam tersebut script akan menampilkan salam yang sesuai seperti contoh berikut ini:

Script meminta input berupa jam sekarang, masukkan nilai jam tanpa menit.



Maka script akan menampilkan salam yang sesuai untuk jam tersebut, seperti pada contoh di atas jam 13.00 maka salam yang ditampilkan adalah "Selamat Siang".

Jam 6 - 10 : Selamat Pagi

Jam 11 - 14 : Selamat Siang

Jam 15 - 17 : Selamat Sore

Jam 18 - 24 : Selamat Malam

Jam 1 - 5 : Selamat Tidur

- b. Buatlah sebuah program yang akan dapat menghitung luas dan keliling dengan menggunakan input lebar dan panjang dari area tersebut. Pada awal script, script akan menanyakan yang hendak dihitung apakah luas atau keliling dari area, kalau user memasukkan kata "luas" maka script akan menghitung luas tapi kalau user memasukkan kata "keliling" maka keliling yang akan dihitung.

-- selamat mengerjakan --

PRAKTIKUM III

TUJUAN:

Mahasiswa mengerti cara menggunakan perintah perulangan pada Javascript.

MATERI:

- Menggunakan Perulangan Pasti / For...
- Menggunakan Perulangan Kondisi / While...
- Menggunakan Perulangan Kondisi / Do...While
- Menggunakan Break dan Continue.

TUGAS PRAKTIKUM III:

1. Fungsi pembuatan program komputer selain dapat menentukan suatu tindakan berdasarkan suatu kondisi tertentu juga berfungsi untuk melakukan pekerjaan yang berulang-ulang. Hal ini juga dapat dilakukan dengan script Javascript dengan menggunakan perintah-perintah perulangan seperti For..., While..., atau Do...While.
2. Pada dunia pemrograman dikenal ada 2 jenis perulangan, yaitu perulangan pasti dan perulangan kondisi, perulangan pasti dipergunakan untuk memecahkan masalah-masalah perulangan yang sudah dapat diketahui dengan pasti berapa kali pekerjaan tersebut harus diulangi. Sedangkan perulangan kondisi digunakan apabila tidak diketahui dengan pasti berapa kali pekerjaan tersebut harus diulangi, tetapi yang diketahui adalah bahwa pekerjaan tersebut harus diulangi terus sampai suatu kondisi tertentu terpenuhi.
3. Untuk perulangan pasti dapat dipergunakan perintah For... seperti contoh di bawah ini, yang dipergunakan untuk melakukan pekerjaan penulisan kata-kata "Test" sebanyak 10 kali di browser daripada harus menuliskan sebanyak 10 baris script.

```
<SCRIPT LANGUAGE="Javascript">
<!--
  for (i=1; i<=10; i++)
  {
    document.write("test " + i + "<BR>");
  }
//-->
</SCRIPT>
```

4. Penjelasan perintah "for (i=1; i<=10; i++)" adalah sebagai berikut:

i=1 : perulangan menggunakan nama variabel "i" dan bernilai awal 1.

i<=10 : nilai akhir dari perulangan adalah 10.

i++ : setiap kali selesai menjalankan pekerjaan (menulis "Test") nilai "i" akan ditambah 1.

secara keseluruhan perintah di atas dapat dibaca sebagai berikut "mulai dari nilai 1, apabila nilai lebih kecil atau sama-dengan 10 maka tambahkan nilai dengan 1".

5. Proses perulangan ini juga dapat dibuat menjadi bertingkat, sehingga terdapat beberapa tingkatan perulangan atau sering juga disebut perulangan bersarang dimana di dalam sebuah proses perulangan terdapat proses perulangan yang lainnya.

6. Dalam pembuatan program dengan perulangan yang bertingkat, harus diingat bahwa **jangan mempergunakan nama variabel yang sama**. Hal ini disebabkan karena variabel perulangan tersebut dipergunakan sebagai counter atau penghitung bahwa perulangan tersebut sudah mengerjakan perintah berapa kali. Dalam pemrograman dengan menggunakan perulangan diharapkan teliti dalam menuliskan perintah perulangan karena apabila salah dapat membuat browser menjadi hang.

7. Sebagai contoh perulangan pasti bertingkat / bersarang seperti tampak di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
  for (i=1; i<=10; i++)
    for (j=11; j<=20; j++)
    {
      document.write("i = " + i + " - ");
      document.write("j = " + j + "<BR>");
    }
//-->
</SCRIPT>
```

8. Dari perintah perulangan di atas dapat dilihat bahwa terdapat 2 buah perulangan yang mempergunakan 2 buah variabel yang berbeda yaitu variabel "i" dan variabel "j". Proses pengerjaan perulangan di atas sebagai berikut:

- a. variabel "i" bernilai 1
 - b. variabel "j" bernilai 11
 - c. jalankan perintah dalam kurung {...}
 - d. periksa apakah nilai "j" lebih kecil sama dengan 20, apabila benar tambah "j" dengan 1.
 - e. jalankan perintah dalam kurung {...}
 - f. ulangi perintah d dan e sampai nilai variabel "j" sama dengan 20.
 - g. periksa apakah nilai "i" lebih kecil sama dengan 10, apabila benar tambah "i" dengan 1.
 - h. variabel "i" bernilai 2
 - i. ulangi perintah b sampai g.
 - j. variabel "i" bernilai 3
 - k. ulangi perintah b sampai g.
 - l. dan seterusnya, perulangan akan berhenti saat nilai variabel "i" sama dengan 10.
9. Jumlah perulangan pasti dapat juga digantikan dengan nilai sebuah variabel yang lain, seperti contoh di bawah ini, dimana jumlah perulangan akan disesuaikan dengan nilai masukan user.

```
<SCRIPT LANGUAGE="Javascript">
<!--
  tampil = prompt("Ditampilkan sebanyak:", "");
  for (i=1; i<=tampil; i++)
  {
    document.write("Tampilan ke-" + i + "<BR>");
  }
//-->
</SCRIPT>
```

10. Sedangkan untuk perulangan yang tidak diketahui dengan pasti berapa kali harus diulangi dapat mempergunakan perulangan berdasarkan kondisi, dimana perulangan akan berhenti apabila suatu kondisi atau keadaan tertentu terpenuhi. Perulangan kondisi mempergunakan perintah While...
11. Contoh di bawah ini akan menampilkan tulisan "Positif" sampai nilai variabel "i" sama dengan 0

```
<SCRIPT LANGUAGE="Javascript">
<!--
  var i = 10;

  while (i > 0)
  {
    document.write("Angka " + i + " adalah Positif <BR>")
    i = i - 1;
  }
//-->
</SCRIPT>
```

12. Pada perulangan kondisi yang harus diperhatikan adalah jangan lupa menyertakan perintah untuk menambah variabel penghitung, seperti tampak di atas pada baris `i = i - 1`; apabila baris ini tidak ada atau kondisi yang diminta tidak pernah tercapai maka browser bisa menjadi hang.
13. Untuk perulangan kondisi, selain mempergunakan perintah While... dapat juga dipergunakan perintah Do...While, seperti tampak di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
  var i = 10;

  do
  {
    document.write("Angka " + i + " adalah Positif <BR>")
    i = i - 1;
  } while (i > 0)
//-->
</SCRIPT>
```

14. Hasil dari kedua script di atas adalah sama, baik yang mempergunakan perintah While maupun yang mempergunakan perintah Do...While. Perbedaannya adalah apabila pada perintah While, kondisi diperiksa terlebih dahulu baru perintah dijalankan, sedangkan pada perintah Do...While perintah dijalankan terlebih dahulu baru kondisi diperiksa. Jadi apabila mempergunakan perintah While maka ada kemungkinan perintah tidak dikerjakan sama sekali, tetapi apabila mempergunakan perintah Do...While maka perintah pasti akan dikerjakan minimal sekali.

Tugas Latihan

- a. Buatlah perulangan untuk menampilkan angka dari 1-50
- b. Buatlah perulangan untuk menampilkan angka ganjil antara 1-50
- c. Buatlah perulangan untuk menampilkan angka kelipatan 3 antara 1-50
- d. Buatlah perulangan untuk menampilkan deretan 1,2,3,5,8,13,21,34,55
- e. Buatlah script yang akan menanyakan password dan apabila password yang ditanyakan tidak sesuai maka user akan ditanyai lagi, tetapi apabila password sesuai maka script akan menampilkan salam selamat datang.



f. Buatlah script untuk menanyakan jumlah siswa yang masuk untuk hari Senin, Selasa, Rabu, Kamis dan Jumat, setelah itu akan menggambarkan grafik berdasarkan nilai tersebut, seperti tampak di bawah ini:

Murid yang masuk hari:

```
Senin : xxxxxxxxxxxx
Selasa: xxxxxxxxxx
Rabu  : xxxxxx
Kamis : xxxxxxxxxxxx
Jumat : xxxxxx
```

-- selamat mengerjakan--

PRAKTIKUM IV

TUJUAN:

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

MATERI:

- Mengetahui 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

TUGAS 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:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
  var a1,a2;
  var b1,b2;
  var c;

  a1 = prompt("Masukkan Angka 1","");
  a2 = prompt("Masukkan Angka 2","");
  c = a1 + a2;
  document.write("Nilai a1 + a2 = " + c + "<BR>");

  b1 = parseInt(a1);
  b2 = parseInt(a2);
  c = b1 + b2;
  document.write("Nilai b1 + b2 = " + c + "<BR>");
//-->
</SCRIPT>
```

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.


```
<SCRIPT LANGUAGE="Javascript">
<!--
    function tulis (kata)
    {
        document.write (kata);
    }

    tulis("Selamat Datang");
//-->
</SCRIPT>
```

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 berguna dapat juga 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.

```
<SCRIPT LANGUAGE="Javascript">
<!--
    var a = 20;
    var b = 10;
    var hasil;

    function tambah (angka1,angka2)
    {
        return (angka1 + angka2);
    }

    function kurang (angka1,angka2)
    {
        return (angka1 - angka2);
    }

    hasil = tambah (10,20);
    document.write("Nilai 10+20 = " + hasil + "<BR>");

    hasil = kurang (a,b);
    document.write("Nilai a - b = " + hasil + "<BR>");

//-->
</SCRIPT>
```

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, perhatikan contoh di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
  var a = 10;

  function fungsi()
  {
    var a = 20;
    document.write("Nilai var a di dalam fungsi = " + a + "<BR>");
  }

  fungsi();
  document.write("Nilai var a di luar fungsi = " + a + "<BR>");
//-->
</SCRIPT>
```

9. Dari contoh di atas dapat dilihat bahwa terdapat perbedaan antara nilai variabel a di luar fungsi dan nilai variabel a di dalam fungsi. Nilai variabel a di luar fungsi adalah nilai variabel global, sedangkan nilai variabel a di dalam fungsi adalah nilai variabel lokal milik fungsi tersebut, kemudian perbandingan dengan apabila fungsi tersebut tidak memiliki variabel lokal seperti contoh berikut:

```
<SCRIPT LANGUAGE="Javascript">
<!--
  var a = 10;


  function fungsi_1()
  {
    var a = 20;
    document.write("Nilai var a di fungsi 1 = " + a + "<BR>");
  }

  function fungsi_2()
  {
    document.write("Nilai var a di fungsi 2 = " + a + "<BR>");
  }

  fungsi_1();
  fungsi_2();
  document.write("Nilai var a di luar fungsi = " + a + "<BR>");
//-->
</SCRIPT>
```

10. Pada contoh di atas, karena function "fungsi_1" memiliki variabel lokal dengan nama "a" maka nilai "a" yang dipergunakan adalah nilai variabel lokal "a" yaitu 20, sedangkan pada function "fungsi_2" 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.

11. Untuk menanyakan sesuatu dapat dipergunakan perintah confirm seperti tampak di bawah ini:

<pre><SCRIPT LANGUAGE="Javascript"> <!-- var nilai,test,jenis; jawab = confirm("Anda suka Javascript?"); document.write(jawab); //--> </SCRIPT></pre>	
--	--

Tugas Latihan

- Ubahlah fungsi "tuliskan" di atas, sehingga pada saat menulis ke browser sekaligus menambahkan baris baru di bawahnya.
- Buatlah sebuah fungsi yang dapat menghitung luas dengan parameter panjang dan lebar.
- Buatlah sebuah fungsi yang dapat menghitung keliling dengan parameter panjang dan lebar.
- Buatlah sebuah fungsi dengan kemampuan sebagai berikut:
 - apabila parameter ("luas",10,20) maka akan mengembalikan nilai luas.
 - apabila parameter ("keliling",10,20) maka akan mengembalikan nilai keliling.
- 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 V

TUJUAN:

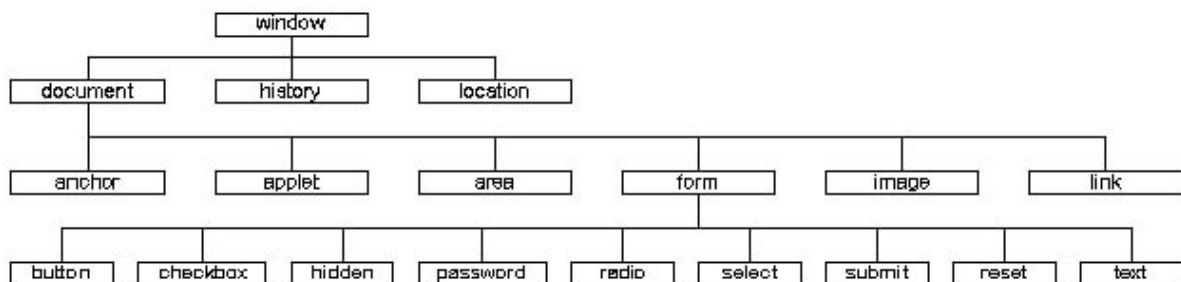
Mahasiswa mengerti tentang Document Object Model (DOM), hubungannya dengan Javascript dan cara penggunaan DOM dengan Javascript.

MATERI:

- Mengetahui Document Object Model.
- Mengetahui Objek
- Mengetahui Event
- Mengetahui Properti
- Mengetahui 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. Salah satu contoh penggunaan objek dan propertinya adalah script untuk mengubah status bar dari browser, dimana objek yang memiliki status bar adalah objek window dan properti untuk status bar adalah defaultStatus atau status, maka penulisannya adalah window.defaultStatus, seperti contoh di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
    window.defaultstatus = "Hallo, teks ini tampil di status bar";
//-->
</SCRIPT>
```

5. 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:

```
<BODY>
  <SCRIPT LANGUAGE="Javascript">
  <!--
    function tampil()
    {
      alert(teks.value);
    }
  //-->
  </SCRIPT>

  <INPUT NAME="teks" TYPE=Text>
  <INPUT TYPE=Button VALUE="Tampil" onClick=tampil()>

</BODY>
```

6. 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 nama "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.

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

```
<BODY>
  <SCRIPT LANGUAGE="Javascript">
  <!--
    function tambah(a,b)
    {
      a=parseInt(a);
      b=parseInt(b);

      alert(a+b);
    }
  //-->
  </SCRIPT>

  Angka 1 =<INPUT NAME="angka1" TYPE=Text><BR>
  Angka 2 =<INPUT NAME="angka2" TYPE=Text><BR>
  <INPUT TYPE=Button VALUE="Tambah" onClick=tambah(angka1.value,angka2.value)>

</BODY>
```

8. 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 Latihan

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

Angka 1 =	<input type="text"/>
Angka 2 =	<input type="text"/>
Hasil =	<input type="text"/>
<input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="x"/> <input type="button" value=":"/>	

-- selamat mengerjakan--

PRAKTIKUM VI

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

TUGAS PRAKTIKUM VI:

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. Objek Array, Boolean, Function, dan Number baru dikenal pada versi Javascript 1.1. Sedangkan objek Screen baru dikenal pada versi Javascript 1.2. 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:

```
<SCRIPT LANGUAGE="Javascript">
<!--
    var salam = new String ("Halo Javascript");

    document.write (salam + "<BR>");
    document.write (salam.bold() + "<BR>");
    document.write (salam.italics() + "<BR>");

//-->
</SCRIPT>
```

6. 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()
7. Contoh metode untuk mengubah tampilan objek String yang lebih lengkap, seperti tampak pada contoh di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
    var salam = new String ("Halo Javascript");

    document.write (salam + "<BR>");
    document.write (salam.bold() + "<BR>");
    document.write (salam.italics() + "<BR>");
    document.write (salam.big() + "<BR>");
    document.write (salam.blink() + "<BR>");
    document.write (salam.fixed() + "<BR>");
    document.write (salam.small() + "<BR>");
    document.write (salam.strike() + "<BR>");
    document.write (salam.sub() + "<BR>");
    document.write (salam.sup() + "<BR>");
    document.write (salam.toUpperCase() + "<BR>");
    document.write (salam.toLowerCase() + "<BR>");
    document.write (salam.fontcolor("Blue") + "<BR>");
    document.write (salam.fontsize("20") + "<BR>");

//-->
</SCRIPT>
```

8. Dari kemampuan metode yang dimiliki oleh objek String dapat dibuat sebuah program Javascript yang dapat mengubah tampilan huruf yang di layar dengan mempergunakan tombol HTML, seperti di bawah ini:

```
<BODY>
<SCRIPT LANGUAGE="Javascript">
<!--
    var str;

    function tebal()
    {
        str = teks.value;
        str = str.bold();
        document.write(str);
    }

    function miring()
    {
        str = teks.value;
        str = str.italics();
        document.write(str);
    }

    function tebalmiring()
    {
        str = teks.value;
        str = str.bold().italics();
        document.write(str);
    }

//-->
</SCRIPT>

<INPUT TYPE=Text NAME="teks">
<INPUT TYPE=Button VALUE="B" onClick=tebal();>
<INPUT TYPE=Button VALUE="I" onClick=miring();>
<INPUT TYPE=Button VALUE="BI" onClick=tebalmiring();>

</BODY>
```


9. 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:

```
<SCRIPT LANGUAGE="Javascript">
<!--
var pecahan = "9.5";
var bulat;

document.write("nilai pecahan = " + pecahan + "<BR>");

bulat = Math.round(pecahan);
document.write("pembulatan normal = " + bulat + "<BR>");

bulat = Math.ceil(pecahan);
document.write("pembulatan ke atas = " + bulat + "<BR>");

bulat = Math.floor(pecahan);
document.write("pembulatan ke bawah = " + bulat + "<BR>");

//-->
</SCRIPT>
```

10. 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 getYear (tahun), seperti contoh script di bawah ini:

```
<SCRIPT LANGUAGE="Javascript">
<!--
var tgl = new Date();
var tanggal = "";

tanggal = tanggal + tgl.getDay() + ", ";
tanggal = tanggal + tgl.getDate() + "-";
tanggal = tanggal + (tgl.getMonth()+1) + "-";
tanggal = tanggal + tgl.getYear();

document.write (tanggal);

//-->
</SCRIPT>
```

11. 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 Latihan

- Perbaiki fungsi untuk menampilkan tanggal di atas sehingga dapat menampilkan tanggal dalam format bahasa Indonesia, seperti: Jumat, 4 Maret 2005.
- 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 VII

TUJUAN:

Mahasiswa mengenal mengenai Array serta mengetahui cara menggunakan Array di dalam program.

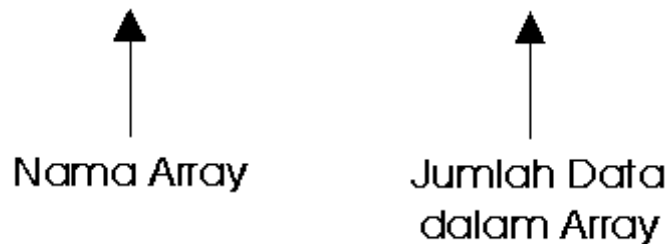
MATERI:

- Pengertian Array
- Pengertian Index
- Array 1 Dimensi
- Array 2 Dimensi

TUGAS PRAKTIKUM VII:

1. Array adalah tempat sementara di memori komputer yang berfungsi untuk menyimpan sekumpulan data-data, baik data itu berupa data String ataupun data Numerik. Masing-masing data yang tersimpan di dalam sebuah Array dapat diakses dengan menggunakan sebuah Index. Index dapat dianggap sebagai identitas data dalam deretan data-data Array tersebut. Array dapat dibuat dengan menggunakan perintah:

```
var nama = new Array (3);
```







2. Dengan menggunakan perintah di atas, telah dibuat sebuah Array yang bernama "nama" dan memiliki tempat penyimpanan data sebanyak 3 buah. Untuk melakukan pengisian data ke dalam Array dapat dilakukan dengan menggunakan perintah:

```
nama[0] = "Budi";
```



3. Nilai Index dimulai dari 0 jadi seperti contoh di atas pada saat pembuatan Array "nama" yang memiliki tempat penyimpanan data sebanyak 3 buah maka telah tersedia 3 buah tempat di memori komputer yang siap untuk diisi dengan data, dimana tempat tersebut dapat diakses dengan menggunakan identitas nama[0], nama[1], dan nama[2].

Untuk lebih jelasnya perhatikan bagan di bawah ini:

KODE PROGRAM	BAGAN ARRAY
<code>var nama = new Array (3);</code>	<p>Array nama</p>  <p>nama[0] nama[1] nama[2]</p>
<code>nama[0] = "Budi";</code>	 <p>nama[0] nama[1] nama[2]</p>
<code>nama[1] = "Dodi";</code>	 <p>nama[0] nama[1] nama[2]</p>
<code>nama[2] = "Rina";</code>	 <p>nama[0] nama[1] nama[2]</p>

4. Apabila hendak melakukan pengambilan data dari dalam Array dapat dilakukan dengan menggunakan nama Array beserta nomer Indexnya, seperti pada contoh program di bawah ini untuk melakukan pencetakan nilai Array ke dalam browser.

```
document.write (nama[0]);
```

maka akan ditampilkan tulisan "Budi" yang merupakan isi data dari Array dengan identitas "nama[0]" ke web browser.

5. Array selain dapat berdimensi 1 seperti contoh di atas dapat juga berdimensi 2, Array yang berdimensi 2 pada umumnya seringkali dipergunakan untuk melakukan penyimpanan data-data yang berbentuk tabel, seperti daftar nilai siswa, daftar harga barang, dan lain sebagainya.

Untuk membuat array berdimensi 2 dapat dilakukan dengan terlebih dahulu membuat Array 1 dimensi kemudian untuk masing-masing data Array tersebut dilakukan pembuatan Array baru sebanyak yang diinginkan.

Pada Array berdimensi 1 hanya dipergunakan 1 buah Index, sedangkan pada Array berdimensi 2 dipergunakan 2 buah Index, anggap saja 1 buah Index itu sebagai penanda baris pada tabel dan 1 buah Index yang lainnya sebagai penanda kolom pada tabel, seperti di bawah ini:

```

Nama Array      Jumlah Baris
  ↓             ↓
var nilai = new Array (3);
nilai[0] = new Array (2);
nilai[1] = new Array (2);
nilai[2] = new Array (2);
                ↑
            Jumlah Kolom
  
```

6. Apabila digambarkan dalam bentuk bagan, perintah di atas akan membentuk sebuah Array 2 dimensi dengan bentuk sebagai berikut:

Array nilai (3)(2)

	0	1
0	nilai[0][0]	nilai[0][1]
1	nilai[1][0]	nilai[1][1]
2	nilai[2][0]	nilai[2][1]

7. Dan untuk mengisi data-data ke dalam Array tersebut dapat dilakukan dengan menggunakan perintah nama array diikuti dengan index baris dan index kolom diikuti dengan tanda sama dengan dan nilai data yang hendak dimasukkan seperti tampak di bawah ini serta tempat data-data tersebut di dalam bagan Array.

KODE PROGRAM	BAGAN ARRAY												
<pre>nilai[0][0] = "Budi"; nilai[0][1] = 90; nilai[1][0] = "Dodi"; nilai[1][1] = 80; nilai[2][0] = "Rina"; nilai[2][1] = 100;</pre>	<table><tr><th></th><th>0</th><th>1</th></tr><tr><th>0</th><td>"Budi"</td><td>90</td></tr><tr><th>1</th><td>"Dodi"</td><td>80</td></tr><tr><th>2</th><td>"Rina"</td><td>100</td></tr></table>		0	1	0	"Budi"	90	1	"Dodi"	80	2	"Rina"	100
	0	1											
0	"Budi"	90											
1	"Dodi"	80											
2	"Rina"	100											

8. Sedangkan untuk mengambil nilai data dari Array 2 dimensi dapat dilakukan seperti halnya pada saat proses pengisian data, yaitu dengan menggunakan nama Array dan diikuti 2 buah index, seperti pada contoh di bawah ini untuk pencetakan data Array 2 dimensi yang terletak pada baris 1 dari bagan di atas:

```

document.write (nilai[1][0] + " = ");
document.write (nilai[1][1] + "<BR>");
    
```

kode program akan menampilkan tulisan "Dodi = 80" di web browser.

9. Index pada sebuah Array dapat juga diakses dengan menggunakan perintah perulangan For... seperti pada contoh berikut ini dimana program akan meminta input dari user mengenai data Nama Barang beserta Harganya dan program akan menyimpan data tersebut ke dalam sebuah Array 2 dimensi dan setelah itu program akan melakukan penghitungan Total Harga Barang dan menampilkan Nama Barang, Harga Barang dan Total Harga ke dalam sebuah tabel.

```

<SCRIPT LANGUAGE="JavaScript">
<!--
var i, jumlah;
var barang = new Array (3);
barang[0] = new Array (2);
barang[1] = new Array (2);
barang[2] = new Array (2);

// ambil input dari user
for (i=0; i<=2; i++)
{
    barang[i][0] = prompt("Nama Barang:", "");
    barang[i][1] = parseInt(prompt("Harga Barang:", ""));
}

// hitung total
jumlah = 0;
for (i=0; i<=2; i++)
    jumlah = jumlah + barang[i][1];

// tampilkan dalam tabel
document.write("<TABLE BORDER=1>");

for (i=0; i<=2; i++)
    document.write("<TR><TD>" + barang[i][0] + "</TD><TD>" + barang[i][1] + "</TD></TR>");

document.write("<TR><TD> TOTAL HARGA </TD><TD>" + jumlah + "</TD></TR>");
document.write("</TABLE>");

//-->
</SCRIPT>

```

Baju	100
Celana	150
Handuk	50
TOTAL HARGA	300

Tugas Latihan

- Buatlah program untuk menampilkan daftar nilai sebuah kelas yang terdiri 10 siswa, dimana data akan ditampilkan dalam bentuk tabel dan terdiri dari data Nama Siswa, Nilai Siswa, dan Nilai rata-rata kelas tersebut.
- Buatlah program password yang dapat menerima beberapa user dengan password yang berbeda-beda (simpan data user dan passwordnya di dalam tabel).

-- selamat mengerjakan --

PRAKTIKUM VIII

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

TUGAS PRAKTIKUM VIII:

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:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
function cektelep()
{
    var no = "";
    no = telep.value;
    if ((no.length < 6) || (no.length > 8))
        alert("Panjang no telep tidak sesuai (6-8)");
}
//-->
</SCRIPT>

No Telepon: <INPUT TYPE=TEXT NAME="telep">
<INPUT TYPE=SUBMIT VALUE=" Cek " onClick="cektelep()">
```

4. 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.
5. Untuk melakukan pemeriksaan terhadap karakter tertentu apakah ada di dalam suatu String atau Kata dapat dipergunakan fungsi **indexOf()**, seperti contoh di bawah ini:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
function cekemail()
{
    var alamat = "";
    alamat = email.value;
    if (alamat.indexOf("@") == -1) alert("Tidak ada karakter @");
    if (alamat.indexOf(".") == -1) alert("Tidak ada karakter .");
}
//-->
</SCRIPT>

Email: <INPUT TYPE=TEXT NAME="email">
<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.

```
<SCRIPT LANGUAGE="JavaScript">
<!--
function cekhobi()
{
    var total = 0;
    if (cbHobi1OR.checked) total = total + 1;
    if (cbHobi1BC.checked) total = total + 1;
    if (cbHobi1TV.checked) total = total + 1;
    alert ("Hobi yang dipilih sebanyak = " + total);
}
//-->
</SCRIPT>

Hobi Anda:<BR>
<INPUT TYPE=Checkbox NAME="cbHobi1OR"> Olah Raga <BR>
<INPUT TYPE=Checkbox NAME="cbHobi1BC"> Baca Buku <BR>
<INPUT TYPE=Checkbox NAME="cbHobi1TV"> Nonton TV<BR>
<INPUT TYPE=SUBMIT VALUE=" Cek " onClick="cekhobi()">
```

7. Selain dapat memeriksa komponen Checkbox, 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 untuk melakukan pengaksesan dapat dilakukan dengan menggunakan Index, seperti halnya pengaksesan pada objek Array. Pada contoh program Javascript di bawah ini dapat dideteksi apabila user belum memasukkan input dan selain itu juga dapat menampilkan pilihan yang dipilih user.

```
<SCRIPT LANGUAGE="Javascript">
<!--

function cekkel()
{
  if (rbkel[0].checked)
  {
    alert("Anda memilih Pria");
  }
  else if (rbkel[1].checked)
  {
    alert("Anda memilih Wanita");
  }
  else
  {
    alert("Anda belum memilih");
  }
}

//-->
</SCRIPT>

Jenis Kelamin Anda:<BR>
<INPUT TYPE=Radio NAME="rbkel"> Pria <BR>
<INPUT TYPE=Radio NAME="rbkel"> Wanita <BR>
<INPUT TYPE=SUBMIT VALUE=" Cek " onClick="cekkel()">
```

8. 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:

```
<SCRIPT LANGUAGE="Javascript">
<!--

function ceknegara()
{
  if (negara.value == "ID")
  {
    alert("Anda memilih Indonesia");
  }
  else if (negara.value == "MY")
  {
    alert("Anda memilih Malaysia");
  }
  else if (negara.value == "AU")
  {
    alert("Anda memilih Australia");
  }
  else
  {
    alert("Anda belum memilih");
  }
}

//-->
</SCRIPT>

Negara Anda:
<SELECT NAME="negara">
  <OPTION VALUE="">--Pilih Negara Anda--
  <OPTION VALUE="ID"> Indonesia
  <OPTION VALUE="MY"> Malaysia
  <OPTION VALUE="AU"> Australia
</SELECT>
<INPUT TYPE=SUBMIT VALUE=" cek " onClick="ceknegara()">
```


9. Komponen Textarea juga dapat dideteksi isinya dengan menggunakan cara sama dengan Combobox atau Textbox, yaitu dengan memeriksa properti value dari komponen yang bersangkutan.
10. 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:

```
<SCRIPT LANGUAGE="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 NAME="nama"><BR>
KOTA: <INPUT TYPE=TEXT NAME="kota"><BR>
<INPUT TYPE=BUTTON VALUE="Kirim" onClick="cekform()">
</FORM>
```

Tugas Latihan

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 (1930-1995).
- Pendidikan : Combobox (SD, SMP, SMA, D3, S1) dan harus diisi.
- Pengalaman : Textarea dan harus diisi.
- Tombol Kirim dan tombol Hapus.

-- selamat mengerjakan --

PRAKTIKUM IX

TUJUAN:

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

MATERI:

- Pengertian Cascading Style Sheet (CSS)
- Cara penulisan CSS secara Inline
- Cara penulisan CSS secara embedding
- Pembuatan CSS Class
- Interaksi Javascript dengan CSS
- Efek-efek Javascript dengan CSS

TUGAS PRAKTIKUM IX:

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 pada komponen-komponen HTML. Selain mempermudah pemberian efek, CSS juga membuat ukuran file HTML yang menggunakan CSS lebih kecil, hal ini disebabkan karena dengan menggunakan CSS sebuah style hanya perlu ditulis sekali saja, tidak perlu ditulis berulang kali seperti halnya pada HTML. Cara penulisan style dengan menggunakan CSS ada beberapa cara, yaitu: secara inline, embedding dan dengan menggunakan external link, yang dibahas di sini hanya secara inline dan embedding.
2. Untuk menuliskan style CSS secara inline dapat dilakukan dengan menyisipkan style tersebut dengan menggunakan atribut STYLE pada tag yang bersangkutan seperti contoh di bawah ini:

```
<HTML>
<BODY>

  <P STYLE="font-family: Arial; color: blue">
    CSS huruf Arial
  </P>

</BODY>
</HTML>
```

atau menggunakan tag seperti contoh di bawah ini:

```
<HTML>
<BODY>

  <P> Ini teks normal, sedangkan

  <SPAN STYLE="font-family: Arial; color: blue">
    Ini teks dengan style CSS
  </SPAN>

  dan ini teks normal lagi.
</P>

</BODY>
</HTML>
```

3. Berikut beberapa contoh style CSS yang dapat dipergunakan:

```
<HTML>
<BODY>

<H1> Efek Huruf </H1>

<SPAN STYLE="font-family: Arial;"> CSS: tipe huruf Arial </SPAN> <BR>
<SPAN STYLE="font-size: 13px;"> CSS: huruf ukuran 12 pixel </SPAN> <BR>
<SPAN STYLE="color: blue;"> CSS: warna biru </SPAN> <BR>
<SPAN STYLE="font-weight: bold;"> CSS: efek tebal </SPAN> <BR>
<SPAN STYLE="font-style: italic;"> CSS: efek miring </SPAN> <BR>
<SPAN STYLE="font-weight: italic;"> CSS: efek miring </SPAN> <BR>
<SPAN STYLE="background-color: yellow;"> CSS: warna background yellow </SPAN> <BR>

<H1> Efek Teks </H1>

<SPAN STYLE="text-transform: uppercase;"> CSS: teks huruf besar </SPAN> <BR>
<SPAN STYLE="text-transform: lowercase;"> CSS: teks huruf kecil </SPAN> <BR>
<SPAN STYLE="text-decoration: none;">
  CSS: teks tanpa efek (digunakan untuk menghilangkan garis bawah di hyperlink)
</SPAN> <BR>
<SPAN STYLE="text-decoration: underline;"> CSS: teks bergaris bawah </SPAN> <BR>
<SPAN STYLE="text-decoration: line-through;"> CSS: teks dicoret </SPAN> <BR>
<SPAN STYLE="text-decoration: blink;"> CSS: teks berkedip </SPAN> <BR>
<P STYLE="text-align: center;">
  CSS: perataan teks (left, right, center, justify)
</P> <BR>

<H1> Efek Bingkai </H1>

<SPAN STYLE="border-style: solid;"> CSS: bingkai solid </SPAN> <BR> <BR>
<SPAN STYLE="border-style: double;"> CSS: bingkai double </SPAN> <BR> <BR>
<SPAN STYLE="border-style: inset;"> CSS: bingkai inset </SPAN> <BR> <BR>
<SPAN STYLE="border-style: outset;"> CSS: bingkai outset </SPAN> <BR> <BR>
<SPAN STYLE="border-style: groove;"> CSS: bingkai groove </SPAN> <BR> <BR>
<SPAN STYLE="border-style: ridge;"> CSS: bingkai ridge </SPAN> <BR> <BR>

<SPAN STYLE="border-style: solid; border-width: 1px;">
  CSS: bingkai sebesar 1 pixel
</SPAN> <BR> <BR>

<SPAN STYLE="border-style: solid; border-color: blue;">
  CSS: bingkai warna biru
</SPAN> <BR> <BR>

</BODY>
</HTML>
```

4. Cara penulisan dengan menggunakan metode inline tidak jauh berbeda dengan menggunakan tag HTML sehingga efisiensi dari kode CSS tidak maksimal, oleh karena itu CSS juga mengenal cara penulisan dengan menggunakan metode embedding dimana setiap style CSS akan didefinisikan ke dalam sebuah kelas CSS sehingga tidak perlu ditulis berulang kali seperti halnya pada metode inline, seperti contoh di bawah ini:

```
<HTML>
<HEAD>
  <STYLE>
    <!--

      .tebal {font-family: verdana;
             font-size: 12px;
             color: blue;
             font-weight: bold;}

      .miring {font-family: verdana;
              font-size: 12px;
              color: blue;
              font-style: italic;}

    //-->
  </STYLE>
</HEAD>

<BODY>
  <P CLASS="tebal"> ini paragraf dicetak tebal </P>
  <P CLASS="miring"> ini paragraf dicetak miring </P>

</BODY>
</HTML>
```

Perhatikan bahwa style CSS didefinisikan di bagian <HEAD> dari HTML dan mempergunakan tag <STYLE> kemudian diikuti dengan nama CLASS CSS yang didahului dengan tanda titik "."

seperti contoh di atas didefinisikan 2 buah kelas style CSS yaitu: .tebal dan .miring.

5. 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:

```
<HTML>
<HEAD>
  <STYLE>
    <!--
      .tebal {font-family: verdana;
            font-size: 12px;
            color: blue;
            font-weight: bold;}

      .miring {font-family: verdana;
              font-size: 12px;
              color: blue;
              font-style: italic;}
    -->
  </STYLE>
</HEAD>
<BODY>
  <SCRIPT LANGUAGE="javascript">
    <!--
      function msover(obj)
      {
        obj.className = "miring";
      }

      function msout(obj)
      {
        obj.className = "tebal";
      }
    -->
  </SCRIPT>

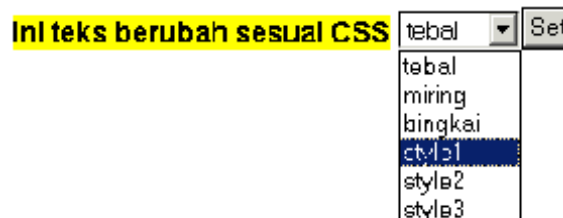
  <A CLASS="tebal" HREF="http://www.google.com" onmouseover="msover(this)" onmouseout="msout(this)">
    ini link ke Google </A>
</BODY>
</HTML>
```

Tugas Latihan

- 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 --