

HTML, PHP DAN JAVA SCRIPT (CSS)

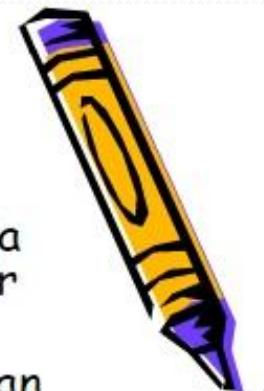
SEJARAH HTML

- Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman web.
- HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu Standard Generalized Markup Language (SGML).
- HTML sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.



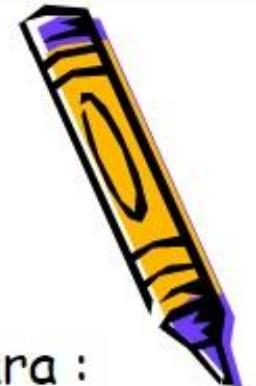
SEJARAH HTML

- HTML dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN dan dipopulerkan pertama kali oleh browser mosaic.
- Selama awal tahun 1990, HTML mengalami perkembangan pesat.
- Perkembangan resmi dikeluarkan pada bulan November 1995 oleh IETF.
- HTML 2.0 merupakan penyempurnaan dari HTML+ (1993).
- HTML 3.0 (1995)
- HTML 3.2(Januari 1997) oleh W3C HTML Working group yg dibentuk tahun 1996.
- HTML 4.01 adalah versi terbaru keluaran W3C pada tanggal 24 April 1998 yg merupakan perbaikan dari HTML 4.0 yg diterbitkan 18 Desember 1997.



KONSEP HTML

- Mendesain HTML dapat dilakukan dengan 2 cara :
 1. Menggunakan HTML Editor, Seperti Microsoft Front Page Editor, Macromedia Dreamweaver, Nescape Composer, Adobe pagemill, Homesite Pro, dll.
 2. Dengan cara menuliskan sendiri secara manual satu persatu tag-tag HTML kedalam dokumen HTML dalam hal ini bisa menggunakan Notepad pada windows ...



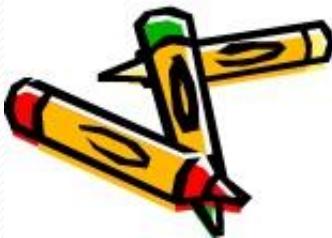
Konsep HTML

- Cara pertama kelebihannya adalah HTML Editor merupakan sebuah program yg khusus didesain untuk membuat, melakukan editing bahkan mem-publish ke internet. Dgn kemampuannya menggabungkan kemudahan dan kecanggihan teknologi internet ke dalam dokumen HTML maka cara ini disukai oleh PEMULA dan desainer yang tidak ingin belajar lebih mendalam mengenai HTML.



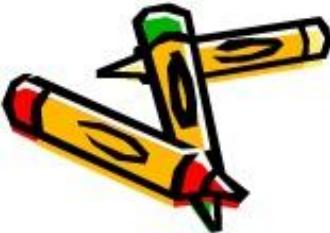
Konsep HTML

- Cara Kedua adalah menuliskan secara manual satu persatu tag-tag HTML. Hal ini sangat sulit karena akan memakan waktu ekstra dan tenaga, ditambah anda juga harus melakukan cara-cara konvensional untuk melihat hasilnya pada browser.
- Namun cara kedua adalah dasar dari segala bentuk HTML, karena dengan cara ini anda akan lebih paham mengenai cara kerja dari berbagai perintah dalam bahasa HTML



Struktur Dokumen HTML

- Secara sederhana HTML terdiri dari dua bagian yaitu "Header" dan "Body".
- Struktur HTML diapit oleh tag awal <HTML> dan tag Akhir </HTML>



Struktur Dokumen HTML

- Standar Penulisannya adalah :

<HTML>

<HEAD>

Deskripsi dokumen

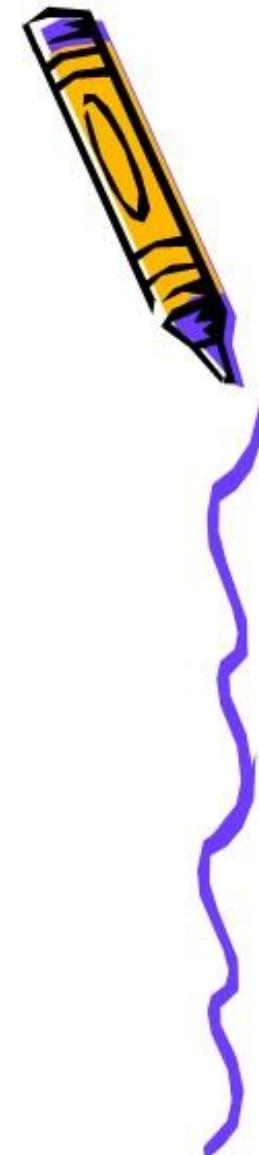
</HEAD>

<BODY>

Isi Dokumen

</BODY>

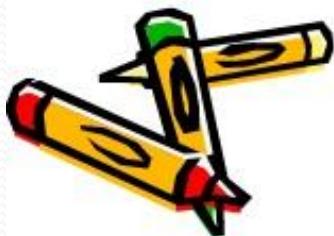
</HTML>



Contoh :

```
<HTML>
<HEAD>
    <TITLE>Selamat Datang Di Homepage Saya</TITLE>
</HEAD>
<BODY>
    Halo, Apa kabar? Homepage ini merupakan karya saya
    yang pertama.
</BODY>
</HTML>
```

Keterangan : Tag <TITLE> digunakan untuk memberi judul
HTML yang dapat dilihat pada pojok kiri atas (title bar)
browser



Apa Itu PHP..?????

PHP adalah singkatan dari “PHP: Hypertext Preprocessor”, yang merupakan sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat.

Sejarah PHP

- PHP pertama dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada awalnya dia menamakan script ini "Personal Home Page Tool" yang merupakan bahasa sederhana dari bahasa pemrograman C, di mana Personal Home Page Tool ini dapat berkomunikasi dengan database dan bersifat Open Source.
- Pada awalnya Rasmus membuat bahasa pemrograman ini bertujuan untuk menyimpan data pengunjung yang melihat biodata pada Website-nya. PHP/FI merupakan cikal bakal PHP yang sekarang.

Keuntungan Bahasa PHP

- Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaanya.
- Web Server yang mendukung PHP dapat ditemukan dimana – mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah. Dalam sisi pengembangan lebih mudah, karena banyaknya milis – milis dan developer yang siap membantu dalam pengembangan.
- Dalam sisi pemahamanan, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
- PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

Ruang Lingkup PHP

➤ Sintax penulisan PHP

```
<?php  
echo "Hello World";  
?>
```

➤ Comment dalam PHP

Pembuatan comment dapat menggunakan “//” untuk membuat komentar satu baris dan /* */ untuk membuat comment yang lebih besar blok-nya (lebih dari 1 baris).

```
<?php  
echo "Hello World"; //Comment untuk satu baris penulisan  
/*  
Comment  
Mencakup  
Blok  
Besar  
*/  
?>
```

Ruang Lingkup PHP

Contoh Deklarasi Variabel

```
<?php  
$teks="Hello World";  
$angka=2011;  
$nilai1 = 5;  
$nilai2 = 3;  
echo $teks."<br>"  
?>
```

Operator Dalam PHP

Arithmatic Operator

Assignment Operator

Comparison Operator

Logical Operator

Statement

- If Statement

```
<?php  
sangka = 1; //deklarasi dan inisialisasi variabel angka=1  
if (sangka == 1) { // mengecek KONDISI apakah variabel angka bernilai 1  
//aksi yang akan ditampilkan jika KONDISI diatas bernilai benar/TRUE  
echo "Angka yang terpilih adalah angka 1 !";  
}?>
```

- If..Else Statement

```
<?php  
sangka = 2;//deklarasi variabel dan inisialisasi variabel angka  
if (sangka == 1) { //mengecek KONDISI variabel angka  
//AKSI yang dijalankan jika KONDISI bernilai benar/TRUE  
echo "Angka yang terpilih adalah angka 1 !";  
}else{  
//AKSI yang dijalankan jika KONDISI bernilai salah/FALSE  
echo "Angka yang terpilih adalah angka 2 !";  
}  
?>
```

Statement

- **If Elseif Else Statement**

```
<?php  
sangka = 3;  
if ($sangka == 1) { //mengecek KONDISI pertama  
//AKSI jika KONDISI pertama terpenuhi  
echo "Angka yang terpilih adalah angka 1 !";  
//AKSI jika KONDISI pertama tidak terpenuhi, maka cek KONDISI kedua  
}elseif ($sangka == 2){  
echo "Angka yang terpilih adalah angka 2 !";  
}else{  
//AKSI jika seluruh KONDISI tidak terpenuhi  
echo "Bukan angka yang terpilih !";  
}  
?>
```

Switch Statement

Switch statement digunakan untuk memilih kode yang ingin dijalankan. Hal pertama yang dilakukan adalah membuat sebuah variabel. Variabel ini akan mengecek jika variabel nya sama dengan statement yang ada di switch maka akan menjalankan statement sesuai dengan statement yang terpilih.

Switch Statement

```
• <?php  
•     $x = 3;  
•     switch ($x){  
•         case 1:  
•             echo "Angka Satu";  
•             break;  
•         case 2:  
•             echo "Angka Dua";  
•             break;  
•         case 3:  
•             echo "Angka Tiga";  
•             break;  
•         default:  
•             echo "Angka yang dimaksud tidak ada !!";  
•     }  
• ?>
```

Looping atau Perulangan

- **For Loop Statement**

```
<?php  
for ($i=1; $i<=5; $i++){  
echo "Angka : ". $i ."  
br />";  
}  
?>
```

Pada script diatas, pengulangan for dimulai dengan melakukan inisialisasi nilai variabel \$i=1. Kemudian menentukan kondisi jika i lebih kecil sama dengan 5, maka proses pengulangan akan terus dilakukan sampai kondisi ini bernilai salah.

Looping atau Perulangan

- **While Loop Statement**

While statement digunakan untuk memanggil kode secara berulang-ulang hingga menjadi TRUE/FALSE. Terdapat 2 while loop statement yaitu while dan do while:

```
<?php
•     $i = 1;
•     echo "Mengurutkan angka hingga 5<br>";
•     while($i <= 5){
•         echo "Angka ".$i."<br>";
•         $i++;
•     }
• ?>
```

Looping atau Perulangan

- **Looping Do While**

Pada statement do ... while, system terlebih dahulu menjalankan aksi yang ada di blok do baru kemudian mengecek kondisi dalam blok while. Dengan menggunakan while, maka aksi tetap akan dijalankan minimal sekali. Berbeda dengan while, aksi ada kemungkinan tidak dijalankan sama sekali karena kondisi dari awal bernilai salah.

- <?php
- \$si = 1;
- do{
- \$si++;
- echo "Angka ".\$si."
";
- }while (\$si <= 5);
- ?>

Fungsi

- Function adalah bagian script dalam PHP yang hanya dijalankan ketika fungsi tersebut dipanggil. Sebagai contoh ketika kita membuat fungsi seperti ini:

```
function fungsiNama(){  
echo "Seameo Seamolec";  
}
```

- Jika kita hanya menjalankan script diatas, maka yang ditampilkan di layar hanyalah tampilan kosong. Karena scrip pada baris echo tersebut hanya berjalan ketika fungsiNama() dipanggil. Sehingga script harus dibuat seperti ini:

```
<?php  
function fungsiNama(){  
echo "Seameo Seamolec";  
}  
echo "Nama lembaga ini adalah ";  
fungsiNama();  
?>
```

Menggunakan Form

- Dalam membuat sebuah form kita akan menggunakan tag-tag HTML. Namun untuk melakukan proses input dan outputnya kita akan menggunakan PHP. Pada PHP terdapat dua method yang digunakan untuk mendistribusikan yaitu metode GET dan POST.

Menggunakan Form

- Method \$_GET

Pada method ini, seluruh informasi yang dikirimkan akan tampil dan dapat dilihat oleh semua orang. Informasi ini nantinya akan terlihat pada bar alamat website. Berikut adalah contoh script penggunaan form menggunakan method \$_GET:

```
<html><body>
<form action="form_proses.php" method="get">
Name : <input typt="text" name="id_nama" />
<input type="submit" />
</form>
</body></html>
```

Menggunakan Form

- Method `$_POST`

`$_POST` memiliki fungsi yang sama dengan method `$_GET`, yaitu untuk mengirimkan informasi. Bedanya, jika kita menggunakan method ini maka informasi yang dikirim tidak dapat dilihat oleh pengakses website. Berikut contoh penggunaan form dengan method `$_POST`:

- <html><body>
- <form action="form_proses.php" method="post">
- Name : <input type="text" name="id_nama" />
- <input type="submit" />
- </form>
- </body></html>



Pengenalan JavaScript

Apa itu JavaScript?

- JavaScript adalah bahasa “Scripting”, bukan bahasa pemrograman.
- JavaScript didisain untuk membuat halaman HTML menjadi lebih interaktif.
- JavaScript biasanya disisipkan langsung pada halaman HTML.
- Client Side scripting.
- Apakah JavaScript sama dengan Java? **TIDAK**. JavaScript dan Java sangat berbeda baik itu dari bahasa maupun dari konsep dan disainnya.
- **JavaScript bersifat Case Sensitive.**



Pengenalan JavaScript

- Nama resmi JavaScript : **ECMAScript**.
- ECMAScript dikembangkan oleh ECMA Organization.
- **ECMA-262** adalah standar resmi JavaScript.
- Diciptakan oleh **Brendan Eich**. Muncul pertama kali di semua browser Netscape dan Microsoft pada tahun 1996.
- Disetujui sebagai standar internasional pada tahun 1998.



Keunggulan JavaScript

- JavaScript dapat menempatkan text secara dinamis pada halaman HTML.
- JavaScript dapat dieksekusi berdasarkan *event* tertentu.
- JavaScript dapat membaca dan menuliskan elemen-elemen HTML.
- JavaScript dapat digunakan untuk memvalidasi data dari form, sebelum dikirimkan ke server.
- JavaScript dapat digunakan untuk mendeteksi browser pengunjung website.
- JavaScript dapat digunakan untuk menyimpan dan mengambil informasi dari komputer pengunjung website.



Penggunaan JavaScript

- Menggunakan tag HTML <script>

Contoh

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



Penempatan JavaScript

- Ditempatkan pada tag <head>
- Ditempatkan pada tag <body>
- Ditempatkan pada tag <head> dan <body>
- Sebagai file external.



JavaScript pada Tag <head>

- Dengan menempatkan sintax JavaScript pada tag head, tidak akan mengganggu isi dari halaman web karena semua script disatukan pada satu tempat.

Contoh :

```
<head>
    <script type="text/javascript">
        alert("JavaScript pada tag Head.");
    </script>
</head>
```



JavaScript pada Tag <body>

Contoh :

```
<body>
    <h1>Belajar JavaScript</h1>
    <p>

        <script type="text/javascript">
            document.write("JavaScript pada tag Body.");
        </script>

    </p>
</body>
```



JavaScript Sebagai File External

- Jika ingin menjalankan script JavaScript yang sama di banyak halaman, gunakan JavaScript sebagai file independen.
- Simpan file tersebut dengan ekstensi .js

Contoh :

```
<head>
    <script type="text/javascript" src="external.js">
    </script>
</head>
```

Penggunaan JavaScript External



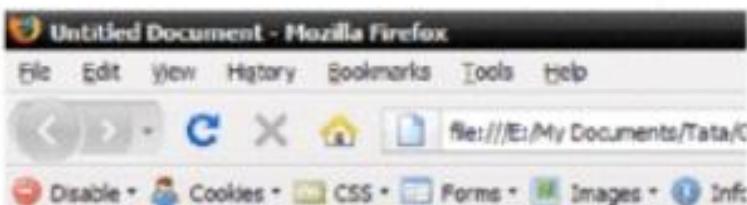
File contoh1.js

```
document.write("<h2>Text ini ditulis dengan JavaScript</h2>");
```

File contoh1.html

```
<head>
    <script type="text/javascript" src="contoh1.js"></script>
</head>
```

Hasil



Text ini ditulis dengan JavaScript



Variabel

- Aturan penulisan Variabel :
 - Nama variabel bersifat **Case Sensitive**
 - Harus dimulai dengan huruf atau karakter *underscore*.
- Deklarasi Variabel dengan statement **var**
 - `var nama;`
 - `var nama="Sandhika";`
- Variabel yang tidak dideklarasikan
 - `nama="Sandhika";`
 - `x=y+5;`



Variabel

- Contoh :

```
<body>
    <h3>Operasi Penjumlahan</h3>
    <b>Dua + Empat =
        <script type="text/javascript">
            var jumlah;
            jumlah = "Enam";
            document.write(jumlah);
        </script>
    </b>
</body>
```

- Hasil

The screenshot shows a simple web page with the title "Untitled Document". Inside the body, there is an

element containing "Operasi Penjumlahan" and a **element containing "Dua + Empat =". Below this, a**



Operator Aritmatika

- Macam-macam Operator Aritmatika pada JavaScript

| Operator | Deskripsi | Contoh | Hasil |
|----------|-------------|--------------------------|----------------------|
| + | Penjumlahan | <code>x = 5 + 2;</code> | <code>x = 7</code> |
| - | Pengurangan | <code>x = 10 - 2;</code> | <code>x = 8</code> |
| * | Perkalian | <code>x = 5 * 4;</code> | <code>x = 20</code> |
| / | Pembagian | <code>x = 10 / 2;</code> | <code>x = 5</code> |
| % | Modulus | <code>x = 5 % 2;</code> | <code>x = 1</code> |
| ++ | Increment | <code>x++;</code> | <code>x = x+1</code> |
| -- | Decrement | <code>x--;</code> | <code>x = x-1</code> |



Contoh

- Penggunaan operator + pada string.

Contoh :

```
<body>
    <script type="text/javascript">
        var x="Belajar";
        var y="JavaScript";
        z = x + y;

        document.write(z);
    </script>
</body>
```

Hasil :



BelajarJavaScript



Contoh

```
x=5+5;  
document.write(x);
```

→ 10

```
x="5"+"5";  
document.write(x);
```

→ 55

```
x=5+"5";  
document.write(x);
```

→ 55

```
x="5"+5;  
document.write(x);
```

→ 55

Jika
menggabungkan
angka dengan
string, maka
hasilnya akan
string.



Operator Assignment

- Macam-macam operator *Assignment* pada JavaScript.

| Operator | Contoh | Sama Dengan | Hasil |
|----------|-------------------|-------------|----------------|
| = | x = "Belajar JS"; | | x = Belajar JS |
| += | x += 6; | x = x + 6; | x = 8 |
| -= | x -= 2; | x = x - 2; | x = 6 |
| *= | x *= y; | x = x * y; | x = 20 |
| /= | x /= 2; | x = x / 2; | x = 5 |
| %= | x %= y; | x = x % y; | x = 1 |



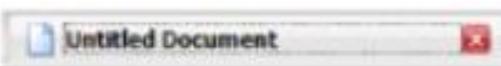
Contoh

- Penggunaan operator Assigment

Contoh :

```
<script type="text/javascript">
    var x = 9;
    var y = 5;
    x += 20;
    x++;
    x -= y;
    document.write("Nilai x = "+x);
</script>
```

Hasil :



Nilai x = 25



Komentar

- Digunakan untuk menerangkan script JavaScript yang dibuat.
- Mencegah script tersebut dieksekusi oleh browser.

Macam-macam komentar.

- *Single Line*
 - Dimulai dengan karakter : //
- *Multi Line*
 - Dimulai dengan karakter : /*
 - Diakhiri dengan karakter : */
 - **Contoh :**

```
/*
    document.write("Script ini tidak akan dieksekusi");
*/
```



Operator Pembanding

- Macam - macam operator pembanding
 - Misal nilai $x = 5$.

| Operator | Deskripsi | Contoh |
|--------------------|--|--|
| <code>==</code> | Sama dengan | $x == 8 \rightarrow$ salah |
| <code>==</code> | Sama persis dengan (tipe data dan nilai) | $x === 5 \rightarrow$ benar $x === "5" \rightarrow$ salah |
| <code>!=</code> | Tidak sama dengan | $x != 8 \rightarrow$ benar |
| <code>></code> | Lebih besar dari | $x > 8 \rightarrow$ salah |
| <code><</code> | Lebih kecil dari | $x < 8 \rightarrow$ benar |
| <code>>=</code> | Lebih besar sama dengan dari | $x >= 8 \rightarrow$ salah |
| <code><=</code> | Lebih kecil sama dengan dari | $x <= 8 \rightarrow$ benar |



Operator Logika

- Macam – macam operator logika
 - Misal nilai $x = 6$, dan $y = 3$.

| Operator | Deskripsi | Contoh |
|-------------------------|-------------|---|
| <code>&&</code> | Dan (and) | $(x < 10 \&\& y > 1) \rightarrow$ benar |
| <code> </code> | Atau (or) | $(x == 5 y == 3) \rightarrow$ benar |
| <code>!</code> | Tidak (not) | $!(x==y) \rightarrow$ benar |



Pengkondisian

- Pada JavaScript terdapat beberapa macam pengkondisian
 - **if (kondisi)**
 - Gunakan statement ini untuk mengeksekusi *source code* hanya jika satu kondisi bernilai true.
 - **if (kondisi1) else (kondisi2)**
 - Gunakan statement ini untuk mengeksekusi *soure code* jika satu kondisi bernilai true, dan kondisi lain bernilai false.
 - **if (kondisi1) else if (kondisi2) else (kondisi3)**
 - Gunakan statement ini untuk memilih satu dari banyak kondisi.
 - **switch**
 - Gunakan statement ini untuk memilih satu dari banyak kondisi.



Pengkondisian

- if statement

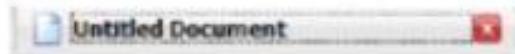
Sintaks

```
if (kondisi)
{
    aksi
}
```

Contoh

```
<script type="text/javascript">
    var jam = 5;
    if (jam <= 10)
    {
        document.write("Selamat Pagi.");
    }
</script>
```

Hasil



Selamat Pagi.



Pengkondisian

- if (kondisi1) else (kondisi2)

Sintaks

```
if (kondisi)
{
    aksi
}
else
{
    aksi
}
```



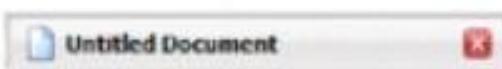
Pengkondisian

- if (kondisi) else

Contoh

```
<script type="text/javascript">
    var jam = 13;
    if (jam <= 10)
    {
        document.write("Selamat Pagi.");
    }
    else
    {
        document.write("Selamat Siang.");
    }
</script>
```

Hasil



Selamat Siang.



Pengkondisian

- **if (kondisi1) else if (kondisi2) else (kondisi3)**

Contoh

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

Hasil



Selamat Sore.



Pengkondisian

- Switch

Contoh

```
<script type="text/javascript">
    var hari = 1;
    var pesan = "";

    switch (hari)
    {
        case 0 : { pesan = "Hari Minggu"; } break;
        case 1 : { pesan = "Hari Senin"; } break;
        case 2 : { pesan = "Hari Selasa"; } break;
        case 3 : { pesan = "Hari Rabu"; } break;
        case 4 : { pesan = "Hari Kamis"; } break;
        case 5 : { pesan = "Hari Jumat"; } break;
        case 6 : { pesan = "Hari Sabtu"; } break;
        default : pesan = "Hari apa?";
    }

    document.write(pesan);
</script>
```

Hasil

Untitled Document

Hari Senin



Ternary

- Menggunakan tanda “?”
- Merupakan model penyederhanaan dari **if...else**
- Cocok untuk melakukan proses pengisian variabel suatu hasil pengujian.

Sintaks

```
namaVariabel = (kondisi)? Nilai1 : Nilai2;
```

Contoh

```
<script type="text/javascript">
    var jam = 20;
    var pesan = "";

    pesan = (jam <= 10)? "Selamat pagi." : "Selamat datang.";

    document.write(pesan);
</script>
```

Hasil

A screenshot of a web browser window titled "Untitled Document". The page content displays the text "Selamat datang.".



Pengulangan

- Pada JavaScript, ada dua macam pengulangan :
 - **FOR**, pengulangan sebanyak jumlah yang ditentukan,
 - **WHILE**, pengulangan sampai bertemu kondisi “true”.



Pengulangan - for

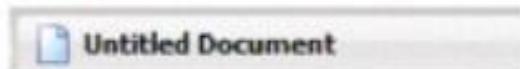
- Sintaks

```
for (var=startValue; var<=endValue; var=var+increment)
{
    source code
}
```

- Contoh

```
<script type="text/javascript">
    var i=0;
    for (i=0;i<=5;i++)
    {
        document.write("Nomor : " + i);
        document.write("<br />");
    }
</script>
```

Hasil



Nomor : 0
Nomor : 1
Nomor : 2
Nomor : 3
Nomor : 4
Nomor : 5



Pengulangan - while

- Sintaks

```
var=startValue;
while (var<=endValue)
{
    source code
}
```

- Contoh

```
<script type="text/javascript">
    var i=0;
    while (i<=5)
    {
        document.write("Nomor : " + i);
        document.write("<br />");
        i++;
    }
</script>
```

Hasil

Untitled Document

Nomor : 0
Nomor : 1
Nomor : 2
Nomor : 3
Nomor : 4
Nomor : 5



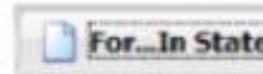
Pengulangan For...In

- Sintaks **For...In** melakukan pengulangan sebanyak elemen array atau sebanyak properti dari suatu objek.
- **Contoh**

```
<script type="text/javascript">
    var x;
    var mycars = new Array();
    mycars[0] = "Saab";
    mycars[1] = "Volvo";
    mycars[2] = "BMW";

    for (x in mycars)
    {
        document.write(mycars[x] + "<br />");
    }
</script>
```

Hasil



Saab
Volvo
BMW

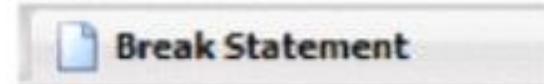


Break Statement

- Perintah **break** akan menghentikan pengulangan dan akan mengeksekusi sintaks berikutnya setelah pengulangan.
- Contoh

```
<script type="text/javascript">
    var i=0;
    for (i=0;i<=10;i++)
    {
        if (i==3)
        {
            break;
        }
        document.write("Nomor : " + i);
        document.write("<br />");
    }
    document.write("Akhir pengulangan.");
</script>
```

Hasil



Nomor : 0
Nomor : 1
Nomor : 2
Akhir pengulangan.



Continue Statement

- Perintah **continue** akan melewati *value* pengulangan yang ditentukan, kemudian melanjutkan ke *value* berikutnya.
- Contoh

```
<script type="text/javascript">
    for (i=0;i<=5;i++)
    {
        if (i==3)
        {
            continue;
        }
        document.write("Nomor : " + i);
        document.write("<br />");
    }
</script>
```

Hasil

Continue Statement

Nomor : 0
Nomor : 1
Nomor : 2
Nomor : 4
Nomor : 5



SELESAI
TERIMAKASIH