

MINGGU 10 (PEMROGRAMAN BERBASIS WEB)

Javascript (BASIC)

Apa itu javascript?

Javascript adalah bahasa pemrograman yang pada tahap awal dirancang untuk berjalan di atas browser. Seiring perkembangan, javascript tidak hanya berjalan di atas browser saja (Client Side). Javascript juga dapat digunakan pada sisi Server, Game, IoT, Desktop, dan lain sebagainya.

Javascript pada awalnya bernama **Mocha**, lalu berubah menjadi **LiveScript** saat browser Netscape Navigator 2.0 rilis versi beta (September 1995). Namun, setelah itu dinamai ulang menjadi Javascript. Terinspirasi dari kesuksesan Javascript, Microsoft mengadopsi teknologi serupa. Microsoft membuat 'Javascript' versi sendiri bernama Jscript dan di tanam pada Internet Explorer 3.0, ini mengakibatkan 'Perang Browser', karena JScript milik Microsoft berbeda dengan Javascript racikan Netscape. Akhirnya pada tahun 1996, Netscape mengirimkan standarisasi ECMA-262 ke ECMA International. Sehingga lahirlah standarisasi kode Javascript bernama ECMAScript atau ES. Saat ini ECMAScript sudah mencapai versi 8 (ES8).

Versi ECMAScript	Tahun Rilis
ES 1	Juni 1997
ES 2	Juni 1998
ES 3	Desember 1999
ES 4	Terbengkalai
ES 5	Desember 2009
ES 5.1	Juni 2011
ES 6	Juni 2015
ES 7	Juni 2016
ES 8	Juni 2017

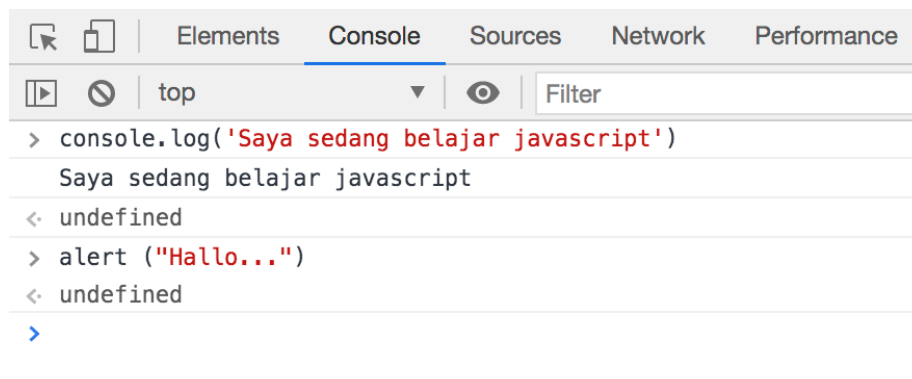
Yang perlu dipersiapkan untuk tahap awal memulai belajar javascript:

- Web Browser (Google Chrome, Firefox, Opera, dll)
- Teks Editor

Mengenal console javascript?

Console Javascript dapat kita buka melalui **Inspect Element->Console**, ini dapat digunakan untuk melihat pesan error yang memang tidak muncul pada tampilan browser pada saat development javascript.

Contoh menuiiskan perintah langsung pada console:



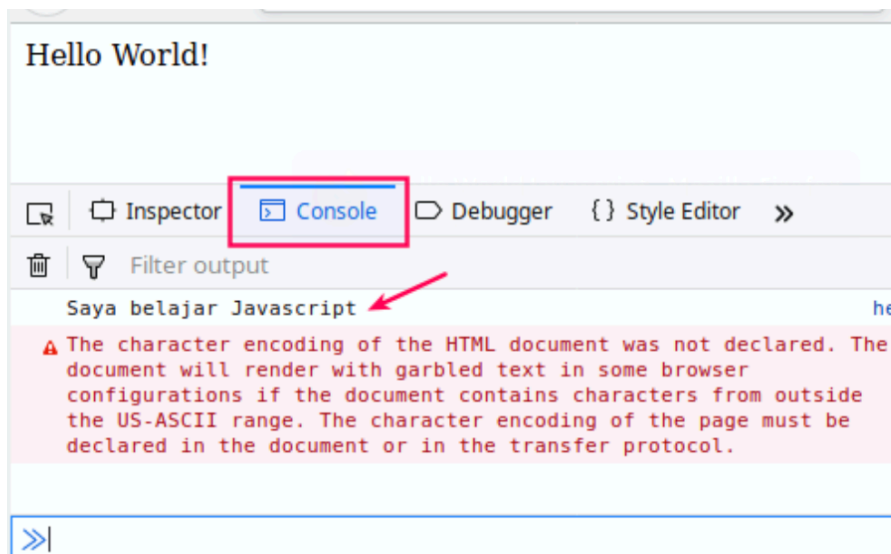
Di dalam *console*, bisa langsung menulis fungsi atau kode-kode javascript dan hasilnya akan langsung ditampilkan.

Praktikum Javascript

Praktikum 1 : buat file **hello-javascript.html**

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello World Javascript</title>
</head>
<body>
  <script>
    console.log("Saya belajar Javascript");
    document.write("Hello World!");
  </script>
</body>
</html>
```

Pada contoh script diatas yang akan ditampilkan pada halaman browser hanya tulisan **Hello World!**, sedangkan text yang ada pada `console.log` hanya ditampilkan melalui **console** javascript.



Selain `document.write` dan `console.log`, untuk dapat mencetak output pada javascript bisa menggunakan fungsi **alert()** maupun **innerHTML**.

Praktikum 2 : buat file *hello2-javascript.html*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Belajar Javascript</title>
</head>
<body>

  <h1>Tutorial Javascript untuk Pemula</h1>
  <div id="hasil-output"></div>

  <script>
    // membuat objek elemen
    var hasil = document.getElementById("hasil-output");

    // menampilkan output ke elemen hasil
    hasil.innerHTML = "<p>Aku suka Javascript</p>";
  </script>

</body>
</html>
```

Kesimpulan proses cetak output pada javascript sebagai berikut:

- **Fungsi console.log()** untuk menampilkan output ke console Javascript;
- **Fungsi alert()** untuk menampilkan output ke jendela dialog;
- **Fungsi document.write()** untuk menulis output ke dokumen HTML;
- **Atribut innerHTML** untuk menampilkan output ke elemen HTML yang lebih spesifik.

Penulisan javascript kedalam HTML

Beberapa cara meletakkan script javascript kedalam HTML adalah sebagai berikut:

1. *Embed* (Kode Javascript ditempel langsung pada HTML)
2. *Inline* (kode Javascript ditulis pada atribut HTML)
3. *Eksternal* (Kode Javascript ditulis terpisah dengan file HTML)

Embed Kode

Pada cara ini, digunakan tag `<script>`, Tag dapat ditulis di dalam tag `<head>` dan `<body>`. (contoh : seperti pada praktikum 1 diatas)

Inline Kode

Pada cara ini, kode javascript ditulis di dalam atribut HTML. Cara ini biasanya digunakan untuk memanggil suatu fungsi pada event tertentu.

Praktikum 3 : tambahkan script pada **file praktikum 1**

```
<a href="#" onclick="alert('Hallo!')">Click me!</a>
```

Eksternal Kode

Menulis kode javascript secara terpisah dengan file HTML, cara ini biasa digunakan untuk kebutuhan yang kompleks, karena cara ini dapat lebih mudah mengelola kode project.

Praktikum 4 : buat file dengan struktur seperti dibawah ini:

```
eksternal-js/  
├── kode.js  
└── index.html
```

Isi kode.js

```
alert("Hello, ini adalah program JS eksternal!");
```

isi index.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Belajar Javascript</title>  
  </head>  
  <body>  
    <p>Tutorial Javascript Pemula</p>  
  
    <!-- Menyisipkan kode js eksternal -->  
    <script src="kode-program.js"></script>  
  </body>  
</html>
```

Variabel dan Tipe Data

Variabel adalah sebuah nama yang mewakili sebuah nilai. Variabel bisa diisi dengan berbagai macam nilai seperti string (teks), number (angka) dan lain sebagainya. Tipe data adalah jenis-jenis data yang bisa disimpan di dalam variabel.

Cara Pembuatan Variabel pada Javascript

Cara membuat variabel yang umum digunakan di javascript adalah menggunakan kata kunci `var` lalu diikuti dengan nama variabel dan nilainya.

Contoh:

```
var title = "Belajar Pemrograman Javascript";  
var siteName = "Belajar Kode";  
var url = "http://dinus.ac.id";  
var visitorCount = 1000;
```

perhatikan deklarasi untuk dua suku kata, Pada Javascript dianjurkan menggunakan `camelCase` dalam penamaan. Boleh-boleh saja menggunakan *snake case* atau *underscore* seperti dibawah ini:

```
var site_name = "Belajar Kode";  
var visitor_count = 1243;
```

Hal tersebut tidak menjadi masalah, program masih tetap valid. Namun, mayoritas programmer Javascript menggunakan `camelCase`.

Menampilkan isi variable

Untuk menampilkan isi variable sama dengan proses cetak nilai yang sudah dijelaskan diatas.

Praktikum 5 : buat file belajar_variable.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Belajar Variabel dalam Javascript</title>
  <script>
    // membuat variabel
    var name = "belajar Kode";
    var visitorCount = 1000;
    var isActive = true;
    var url = "http://dinus.ac.id";

    // menampilkan variabel ke jendela dialog (alert)
    alert("Selamat datang di " + name);

    // menampilkan variabel ke dalam HTML
    document.write("Nama Situs: " + name + "<br>");
    document.write("Jumlah Pengunjung: " + visitorCount + "<br>");
    document.write("Status Aktif: " + isActive + "<br>");
    document.write("Alamat URL: " + url + "<br>");
  </script>
</head>
<body>

</body>
</html>
```

Mengisi ulang nilai variable

Variabel bersifat *mutable*, artinya nilai yang tersimpan di dalamnya dapat diisi ulang (berubah).

Contoh:

```
// mula-mula kita buat variabel dengan isi seperti ini
var age = 18;

// lalu kita isi ulang
age = 21;
```

pada saat mengisi ulang nilai variabel tidak perlu menggunakan kata kunci var, karena kata kunci var dibutuhkan saat membuat variabel saja. Apabila menggunakan kata kunci var, berarti artinya membuat variabel baru, bukan mengisi ulang.

Menghapus Variable

Penghapusan variabel dalam Javascript memang jarang dilakukan. Namun, untuk program yang membutuhkan ketelitian dalam alokasi memori, penghapusan variabel perlu dilakukan agar penggunaan memori lebih optimal.

Penghapusan variabel dapat dilakukan dengan kata kunci **delete**.

Contoh:

```
bookTitle = "Belajar Pemrograman Javascript";  
delete bookTitle;
```

Penghapusan variabel hanya bisa dilakukan pada variabel yang dibuat tanpa awalan. Sedangkan variabel yang dibuat dengan kata kunci `var` dan `let` akan dihapus otomatis.

TIPE DATA

Ada beberapa tipe data dalam pemrograman Javascript:

- String (teks)
- Integer atau Number (bilangan bulat)
- Float (bilangan Pecahan)
- Boolean
- Object

Javascript adalah bahasa yang bersifat *dynamic typing*, artinya tidak harus menuliskan tipe data pada saat pembuatan variabel seperti pada bahasa [C](#), [C++](#), [Java](#), dsb. yang bersifat *static typing*.

Contoh:

```
var name = "Dian";  
var age = 22;  
var single = true;
```

Javascript akan otomatis mengenali tipe data yang kita berikan pada variabel.

Javascript akan otomatis mengenali tipe data yang kita berikan pada variabel.

Pada contoh di atas variabel:

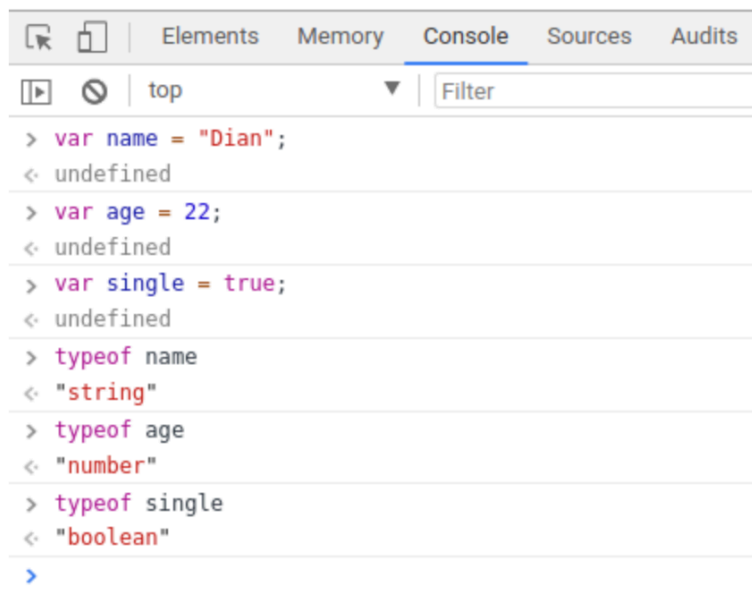
- **name** bertipe data **String**;
- **age** bertipe data **integer**;
- dan **single** bertipe data **boolean**.

Hal ini bisa juga kita cek dengan kata kunci `typeof`.

Contoh:

```
typeof name;  
typeof age;  
typeof single;
```

Hasilnya pada console:



The screenshot shows a browser's developer console with the 'Console' tab selected. It displays the following sequence of commands and results:

```
> var name = "Dian";  
< undefined  
> var age = 22;  
< undefined  
> var single = true;  
< undefined  
> typeof name  
< "string"  
> typeof age  
< "number"  
> typeof single  
< "boolean"  
>
```

Aturan penulisan nama Variable

Ada beberapa aturan penulisan variabel dalam Javascript:

- Penamaan variabel **tidak boleh** menggunakan angka di depannya.

```
// salah  
var 123nama = "Belajar Kode";  
// benar  
var nama123 = "Belajar Kode";
```

- Penamaan variabel **boleh** menggunakan awal underscore.

```
var _nama = "Petani Kode";
```

- Penamaan variabel **dianjurkan** menggunakan camelCase apabila terdiri dari dua suku kata.

```
var fullName = "Petani Kode";
```

- Penamaan variabel **dianjurkan** menggunakan bahasa inggris

```
var postTitle = "Tutorial Javascript untuk Pemula";
```

Jendela Dialog

Jendela dialog merupakan jendela yang digunakan untuk berinteraksi dengan pengguna.

Ada tiga macam jendela dialog pada Javascript:

1. Jendela dialog **alert()**;
2. Jendela dialog **confirm()**;
3. Jendela dialog **prompt()**;

Dialog Alert

Dialog alert() biasa digunakan untuk menampilkan pesan peringatan atau informasi, Fungsi alert() berada dalam objek window.

syntax seperti ini:

```
window.alert("Hello World!");
```

atau bisa juga seperti ini:

```
alert("Hello kawan");
```

Karena objek window bersifat global, boleh tidak dituliskan, dialog **alert()** tidak akan mengembalikan nilai apa-apa saat dieksekusi.

Praktikum 6 : buat file **alert.html** dan **alert2.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Dialog Alert</title>
  </head>
  <body>
    <script>
      alert("Selamat datang di tutorial Javascript");
    </script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Dialog Alert</title>
  </head>
  <body>
    <button onClick="alert('Tombol diklik!')">Klik Saya</button>
  </body>
</html>
```

Dialog Confirm

Dialog confirm() digunakan untuk melakukan konfirmasi dalam melakukan tindakan tertentu, misalnya pada saat akan menghapus sesuatu, maka ada baiknya menampilkan dialog confirm(). Karena tindakan tersebut cukup berbahaya.

Dialog confirm dapat dibuat dengan fungsi **confirm()**.

Praktikum 7 : buat file **confirm.html**

```
<!DOCTYPE html><html>
  <head>
    <title>Dialog Confirm</title>
  </head>
  <body>
    <script>
      var yakin = confirm("Apakah kamu yakin akan mengunjungi website UDINUS?");

      if (yakin) {
        window.location = "http://dinus.ac.id ";
      } else {
        document.write("belum, tetap di sini saja :");
      }
    </script>
  </body>
</html>
```

Dialog `confirm()` akan mengembalikan nilai `true` apabila memilih tombol **OK** dan akan mengembalikan nilai `false` apabila memilih **Cancel**.

Dialog Prompt

Dialog **`prompt()`** berfungsi untuk mengambil sebuah inputan dari pengguna, dialog **`prompt()`** akan mengembalikan sebuah nilai string dari apa yang diinputkan oleh pengguna.

Praktikum 8 : buat file `prompt.html`

```
<!DOCTYPE html>
<html>
  <head>
    <title>Dialog Promp</title>
  </head>
  <body>
    <script>
      var nama = prompt("Siapa nama kamu?", "");
      document.write("<p>Hello " + nama + "</p>");
    </script>
  </body>
</html>
```

Kapan Waktu yang Tepat Menggunakan Alert, Confirm, dan Prompt?

Berdasarkan contoh-contoh di atas, saat hanya ingin **menampilkan informasi** saja, maka gunakan `alert()`, kemudian saat ingin **jawaban konfirmasi** dari pengguna, maka gunakan `confirm()`, dan apabila ingin **mengambil data** teks dari pengguna, maka gunakan `prompt()`.

Operator Javascript

Operator adalah simbol yang digunakan untuk melakukan operasi pada suatu nilai dan variabel.

Operator dalam pemrograman terbagi dalam 6 jenis:

1. Operator aritmatika;
2. Operator Penugasan (Assignment);
3. Opearator relasi atau perbandingan;

4. Operator Logika;
5. Operator Bitwise;
6. Operator Ternary;

Operator aritmatika

Operator aritmatika merupakan operator untuk melakukan operasi aritmatika seperti penjumlahan, pengurangan, pembagian, perkalian, dan lain-lain.

Operator aritmatika terdiri dari:

Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pemangkatan	**
Pembagian	/
Sisa Bagi	%

Untuk melakukan operasi perkalian, gunakan simbol asterik *, tidak bisa menggunakan simbol x, karena simbol x bukan termasuk dalam operator di dalam pemrograman, kemudian untuk pemangkatan gunakan asterik ganda **, untuk pembagian, gunakan simbol garis miring /.

Praktikum 9 : buat file **aritmatika.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Operator Aritmatika</title>
</head>

<body>

  <script>
    var a = 15;
    var b = 4;
```

```

    var c = 0;

    // pengurangan
    c = a - b;
    document.write(`${a} - ${b} = ${c}<br/>`);

    // Perkalian
    c = a * b;
    document.write(`${a} * ${b} = ${c}<br/>`);

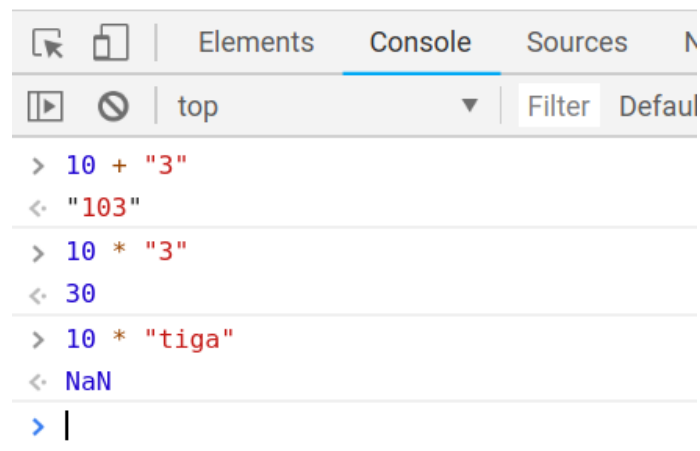
    // pemangkatan
    c = a ** b;
    document.write(`${a} ** ${b} = ${c}<br/>`);

    // Pembagian
    c = a / b;
    document.write(`${a} / ${b} = ${c}<br/>`);

    // Modulo
    c = a % b;
    document.write(`${a} % ${b} = ${c}<br/>`);
</script>
</body>
</html>

```

Untuk operasi yang lainnya, silahkan dicoba-coba melalui *console*.



Operator Penugasan

Operator penugasan adalah operator yang digunakan untuk memberikan tugas kepada variabel. Biasanya digunakan untuk mengisi variabel.

Contoh :

```
var a = 19;
```

Pada contoh diatas, variabel a diberikan tugas untuk menyimpan nilai 19.

Operator penugasan terdiri dari:

Nama Operator	Sombol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pemangkatan	**=
Pengisian dan Pembagian	/=
Pengisian dan Sisa bagi	%=

Praktikum 10 : buat file **op-penugasan.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Operator Penugasan</title>
</head>

<body>

  <script>
    document.write("Mula-mula nilai score...<br>");
    // pengisian nilai
    var score = 100;
    document.write("score = "+ score + "<br/>");

    // pengisian dan menjumlahan dengan 5
    score += 5;
    document.write("score = "+ score + "<br/>");

    // pengisian dan pengurangan dengan 2
    score -= 2;
    document.write("score = "+ score + "<br/>");

    // pengisian dan perkalian dengan 2
    score *= 2;
    document.write("score = "+ score + "<br/>");

    // pengisian dan pembagian dengan 4
    score /= 4;
    document.write("score = "+ score + "<br/>");
```

```
// pengisian dan pemangkatan dengan 2
score *= 2;
document.write("score = "+ score + "<br/>");

// pengisian dan modulo dengan 3;
score %= 3;
document.write("score = "+ score + "<br/>");
</script>
</body>
</html>
```

Apa bedanya operator penugasan dengan operator aritmatika?

Operator aritmatika hanya melakukan operasi aritmatika saja, sedangkan operator penugasan melakukan operasi aritmatika dan juga pengisian.

Operator Perbandingan

Operator relasi atau perbandingan adalah operator yang digunakan untuk membandingkan dua nilai, operator perbandingan akan menghasilkan sebuah nilai *boolean* **true** dan **false**.

Operator perbandingan terdiri dari:

Nama Operator	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	== atau ===
Tidak Sama dengan	!= atau !==
Lebih Besar Sama dengan	>=
Lebih Kecil Sama dengan	<=

Praktikum 10 : buat file op-perbandingan.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Operator Perbandingan</title>
</head>
```



```

<body>

  <script>
    var aku = 20;
    var kamu = 19;

    // sama dengan
    var hasil = aku == kamu;
    document.write(`${aku} == ${kamu} = ${hasil}<br/>`);

    // lebih besar
    var hasil = aku > kamu;
    document.write(`${aku} > ${kamu} = ${hasil}<br/>`);

    // lebih besar samadengan
    var hasil = aku >= kamu;
    document.write(`${aku} >= ${kamu} = ${hasil}<br/>`);

    // lebih kecil
    var hasil = aku < kamu;
    document.write(`${aku} < ${kamu} = ${hasil}<br/>`);

    // lebih kecil samadengan
    var hasil = aku <= kamu;
    document.write(`${aku} <= ${kamu} = ${hasil}<br/>`);

    // tidak samadengan
    var hasil = aku != kamu;
    document.write(`${aku} != ${kamu} = ${hasil}<br/>`);
  </script>
</body>

</html>

```

Apa perbedaan == (dua simbol samadengan) dengan === (tiga simbol samadengan)?

Perbandingan menggunakan simbol == hanya membandingkan nilai, sedangkan menggunakan === akan membandingkan nilai dan tipe data.

```

// ini akan bernilai true
var a = "4" == 4; //-> true

// sedangkan ini akan bernilai false
var b = "4" === 4; //-> false

```

Mengapa nilai b bernilai false? karena "4" (string) dan 4 (integer). Tipe datanya berbeda.

Operator Logika

Operator logika digunakan untuk melakukan operasi terhadap dua nilai **boolean**.

Operator ini terdiri dari:

Nama Operator	Simbol
Logika AND	&&
Logika OR	
Negasi/kebalikan	!

Praktikum 11 : buat file **op-logika.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Operator Logika</title>
</head>

<body>

  <script>
    var aku = 20;
    var kamu = 19;

    var benar = aku > kamu;
    var salah = aku < kamu;

    // operator && (and)
    var hasil = benar && salah;
    document.write(`${benar} && ${salah} = ${hasil}<br/>`);

    // operator || (or)
    var hasil = benar || salah;
    document.write(`${benar} || ${salah} = ${hasil}<br/>`);

    // operator ! (not)
    var hasil = !benar;
    document.write(`${!benar} = ${hasil}<br/>`);

  </script>
</body>
</html>
```

Operator Bitwise

Operator bitwise merupakan operator yang digunakan untuk operasi berdasarkan bit (biner).

Operator ini terdiri dari:

Nama	Simbol di Java
AND	&
OR	
XOR	^
Negasi/kebalikan	~
Left Shift	«
Right Shift	»
Left Shift (unsigned)	«<
Right Shift (unsigned)	»>

Operator ini berlaku untuk tipe data ***int, long, short, char, dan byte***.

Operator ini akan menghitung dari ***bit-ke-bit***.

Misalnya, variabel a = 60 dan b = 13.

Bila dibuat dalam bentuk biner, akan menjadi seperti ini:

```
a = 00111100
b = 00001101
```

Kemudian, dilakukan operasi bitwise

Operasi **AND**

```
a      = 00111100
b      = 00001101
a & b  = 00001100
```

Operasi **OR**

```
a      = 00111100
b      = 00001101
a | b  = 00111101
```

Operasi **XOR**

```
a      = 00111100
b      = 00001101
a ^ b  = 00110001
```

Operasi **NOT** (Negasi / Kebalikan)

```
a      = 00111100
~a     = 11000011
```

(perhatikan bilangan binernya, angka 0 dan 1)

Konsepnya memang hampir sama dengan operator Logika. Bedanya, Bitwise digunakan untuk biner.

Praktikum 12 : buat file **op-bitwise.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Operator Bitwise</title>
</head>

<body>

  <script>
    var x = 4;
    var y = 3;

    // operator bitwise and
    var hasil = x & y;
    document.write(`${x} & ${y} = ${hasil}<br/>`);

    // operator bitwise or
    var hasil = x | y;
    document.write(`${x} | ${y} = ${hasil}<br/>`);

    // operator bitwise xor
    var hasil = x ^ y;
    document.write(`${x} ^ ${y} = ${hasil}<br/>`);

    // operator negasi
    var hasil = ~x;
    document.write(`~${x} = ${hasil}<br/>`);

    // operator bitwise right shift >>
    var hasil = x >> y;
    document.write(`${x} >> ${y} = ${hasil}<br/>`);

    // operator bitwise left shift <<
    var hasil = x << y;
```

```

    document.write(`${x} << ${y} = ${hasil}<br/>`);

    // operator bitwise right shift (unsigned) >>>
    var hasil = x >>> y;
    document.write(`${x} >>> ${y} = ${hasil}<br/>`);

</script>
</body>
</html>

```

Operator Ternary

Operator ternary merupakan operator yang terdiri dari **tiga bagian**, operator-operator sebelumnya hanya **dua bagian** saja, yaitu: bagian **kiri dan kanan**. Ini disebut operator binary, sementara operator trinary ada bagian **kiri, tengah, dan kanan**.

```
bagian kiri <operator> bagian tengah <operator> bagian kanan
```

Operator ternary pada Javascript, biasanya digunakan untuk membuat sebuah percabangan **if/else**.

Simbol operator ternary terdiri dari tanda tanya dan titik dua (**?:**).

Bentuknya seperti ini:

```
<kodisi> ? "benar" : "salah"
```

Perhatikan! **<kondisi>** dapat diisi dengan ekspresi yang menghasilkan nilai true dan false, apabila kondisi bernilai true, maka "benar" yang akan dipilih dan sebaliknya—apabila false—maka "salah" yang akan dipilih.

Operator ini unik, seperti membuat pertanyaan.

Operator Ternary

kamu suka aku ? ya : tidak;

jawaban benar jawaban salah

Praktikum 13 : buat file **op-ternary.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Operator Ternary</title>
</head>

<body>

  <script>
    var pertanyaan = confirm("Apakah kamu berumur diatas 18 tahun?")

    var hasil = pertanyaan ? "Selamat datang" : "Kamu tidak boleh di
sini";
    document.write(hasil);
  </script>
</body>

</html>
```

Repository Assignment #7 Pertemuan 10

Push hasil latihan ke Github (repository webdas) dan kirim urlnya melalui kulino pada blok (Repository Assignment #7 Pertemuan 10).

Untuk susunan folder dan file sebagai berikut

- repominggu10 (folder utama)
 - latihanJavascript
 - hello-javascript.html
 - dan seterusnya sesuai susunan folder Latihan diatas.