



Vocational School Graduate Academy

Web Developer

Pertemuan #4: Mengidentifikasi mekanisme running atau eksekusi source code

Profil Pengajar

Jabatan Akademik <tahun dan jabatan terakhir Pengajar>

Pendidikan

☐ <riwayat pendidikan Pengajar>

Riwayat Pekerjaan

☐ <riwayat pekerjaan Pengajar>

Foto
Pengajar

Contact

HP WA only :<no hp Pengajar>

Email :<email Pengajar>

Foto
Pengajar

Contact

HP WA only :<no hp Pengajar>

Email :<email Pengajar>

Mengidentifikasi Mekanisme Running Atau Eksekusi Source Code



Deskripsi Singkat mengenai Topik

Topik dalam pertemuan kali ini akan fokus ke mengidentifikasi mekanisme running atau eksekusi source code

Tujuan Pelatihan

Mengidentifikasi mekanisme running atau eksekusi source code

Materi Yang akan disampaikan:

1. Cara dan tools untuk mengeksekusi source code diidentifikasi
2. Parameter untuk mengeksekusi source code diidentifikasi
3. Peletakan source code sehingga bisa dieksekusi dengan benar diidentifikasi

Tugas :

1. Membuat form sederhana dengan php
2. Mengkoneksikan script dengan basis data

Outcome/Capaian Pelatihan

1. Siswa dapat menjelaskan cara dan tools untuk mengeksekusi source code
2. Siswa dapat menjelaskan parameter untuk mengeksekusi source code
3. Siswa dapat menjelaskan dimana source code dapat dieksekusi

Review Tools Dasar Pemrograman

- ❖ XAMPP adalah perangkat yang menggabungkan tiga aplikasi kedalam satu paket, yaitu Apache, MySQL, dan PHPMyAdmin.
- ❖ Apache adalah server web yang dapat dijalankan di banyak sistem operasi, yang berguna untuk melayani dan memfungsikan situs web.
- ❖ MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread dan multi-user.
- ❖ PHP: Hypertext Preprocessor adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML.



Review Tools Dasar Pemrograman

- ❖ HTML (HyperText Markup Language) yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web Internet (Browser).
- ❖ CSS (Cascading Style Sheets) berisi rangkaian instruksi yang menentukan bagaimana suatu text akan tertampil di halaman web.
- ❖ JavaScript adalah bahasa pemrograman web yang bersifat Client Side Programming Language, yang pemrosesannya dilakukan oleh client.



Tools Pemrograman

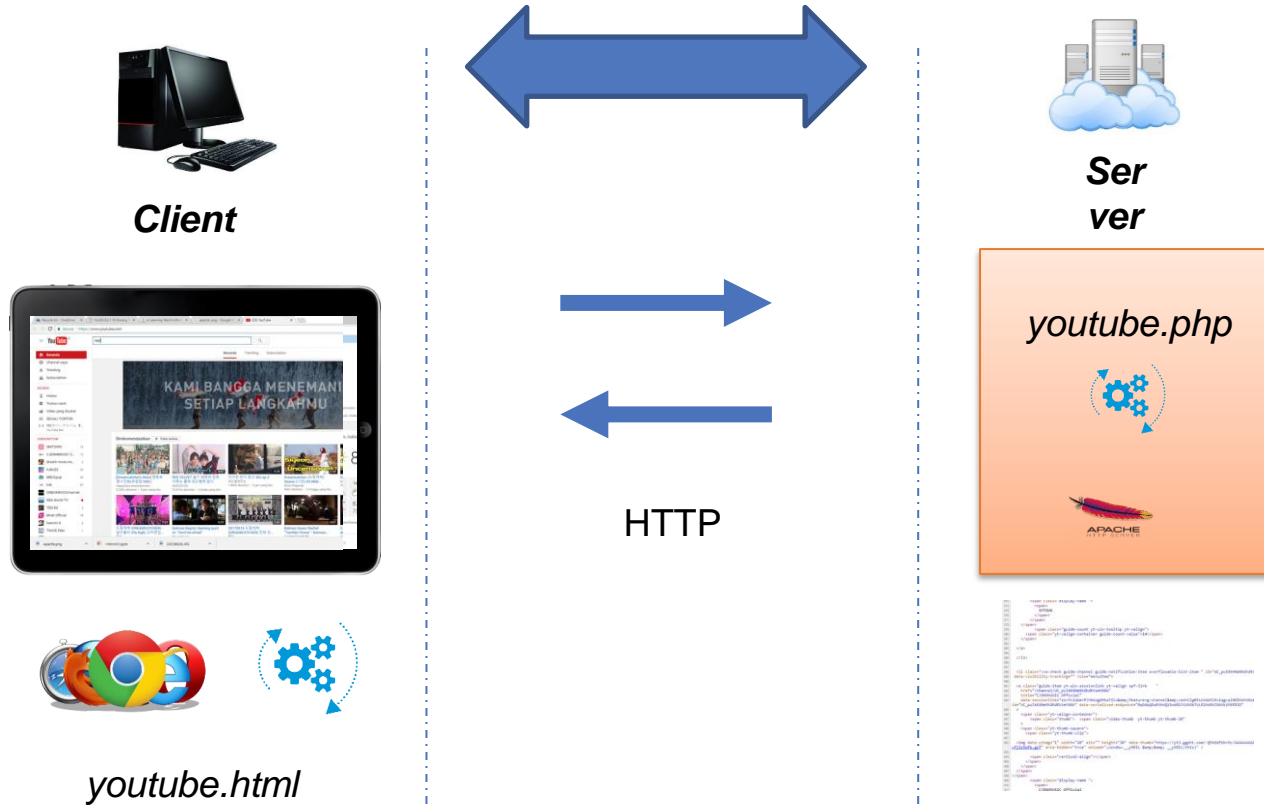


Tools pemrograman adalah perangkat lunak yang digunakan untuk mengeksekusi source code hasil pemrograman, tools pemrograman bisa berupa compiler atau interpreter.

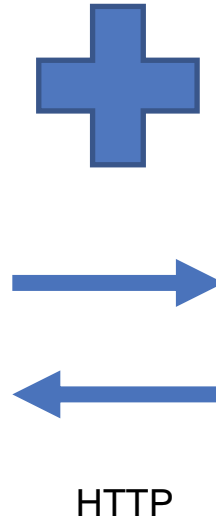
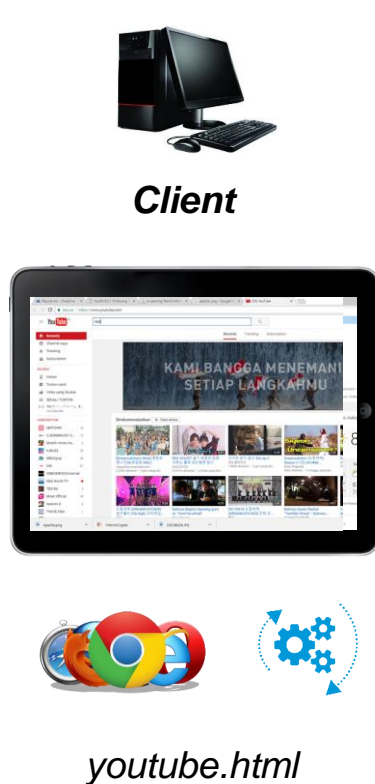
Contoh tools pemrograman pada Web misalnya HTML dan PHP

Mekanisme eksekusi Source Code dilakukan dengan menggunakan Tools yakni Tag HTML , Tag PHP, Comand Prompt dan XAMPP.

Cara Kerja Web PHP dengan Server

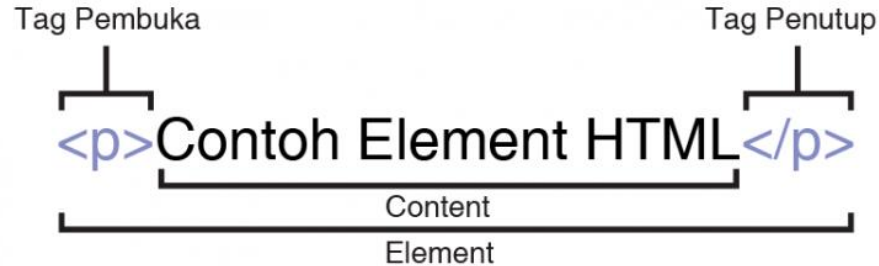


Cara Kerja Web PHP dengan Server Local (Localhost)



Element dan Tag pada HTML

Element pada HTML membentuk sebuah struktur dan memberitahu browser untuk halaman website yang akan ditampilkan. Biasanya untuk element terdiri dari tag pembuka, konten, dan tag penutup.



Berikut adalah contoh element pada HTML:
`Contoh element.`

Attribut pada HTML

Setelah Anda memahami element dan tag, Anda perlu memahami apa itu attribute. Seperti diketahui jika sebuah element berupa struktur pada HTML yang berfungsi untuk memberitahu browser mengenai apa yang akan ditampilkan.

Attribute



```
<p class="contoh-attribut">contoh attribut</p>
```

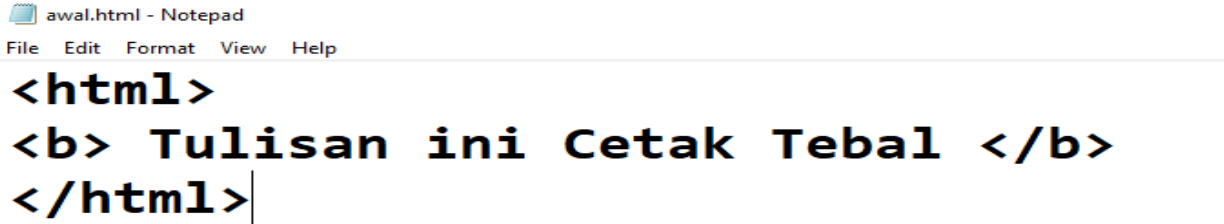
Berikut adalah contoh element yang disertai sebuah attribute:

Berikut adalah a href="https://www.google.com/" title="Search" target="_blank"Engine terbaik/a./p

Contoh HTML

Langkah Percobaan :

Bukalah sebuah editor teks pada Kasus ini menggunakan notepad



```
awal.html - Notepad
File Edit Format View Help
<html>
<b> Tulisan ini Cetak Tebal </b>
</html>
```

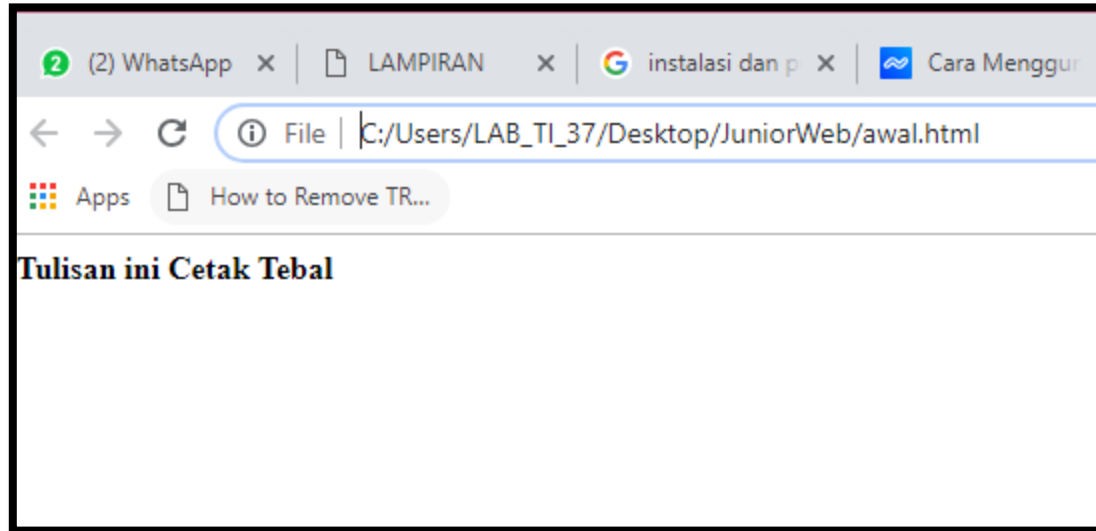
Kemudian lakukan penyimpanan file pada sebuah folder dengan ekstensi .html

Contoh: awal.html

Contoh HTML

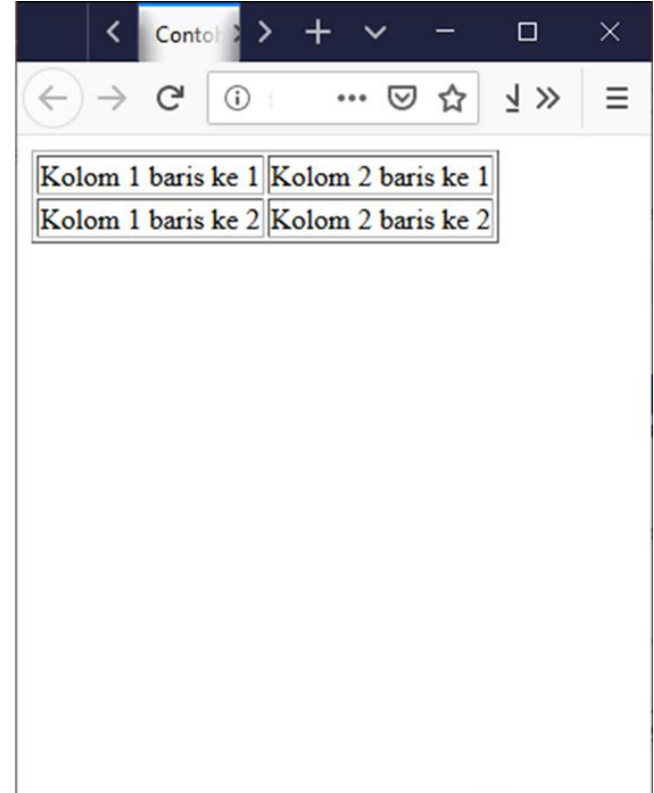
Setelah file tersebut disimpan maka buka kembali file tersebut menggunakan salah satu Browser (Firefox, Chrome, dll)

Tampilan Browser pada saat dieksekusi:



Contoh Table HTML

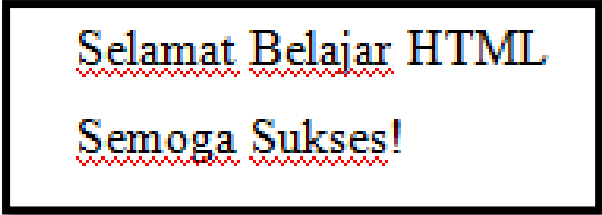
```
1 <html>
2 <head>
3 <title>Contoh Table 1</title>
4 </head>
5 <body>
6
7 <table border="1">
8   <tr>
9     <td>Kolom 1 baris ke 1</td>
10    <td>Kolom 2 baris ke 1</td>
11  </tr>
12  <tr>
13    <td>Kolom 1 baris ke 2</td>
14    <td>Kolom 2 baris ke 2</td>
15  </tr>
16 </table>
17 </body>
18 </html>
```



Kolom 1 baris ke 1	Kolom 2 baris ke 1
Kolom 1 baris ke 2	Kolom 2 baris ke 2


Buatlah sebuah halaman Web Sederhana dengan Tampilan Berikut :

1. Buat 2 baris kalimat seperti disamping.



Selamat Belajar HTML
Semoga Sukses!

2. Buat judul dan list data pada halaman website seperti gambar disamping.



Aneka Tanaman Hias

Sri Rejeki

Begonia

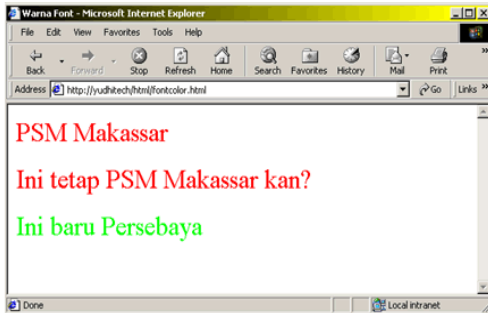
Bromelia

dan lain-lain

3. Buat kalimat dengan font dan size yang berbeda.



4. Buat kalimat dengan warna yang berbeda-beda.



Apa itu javascript?

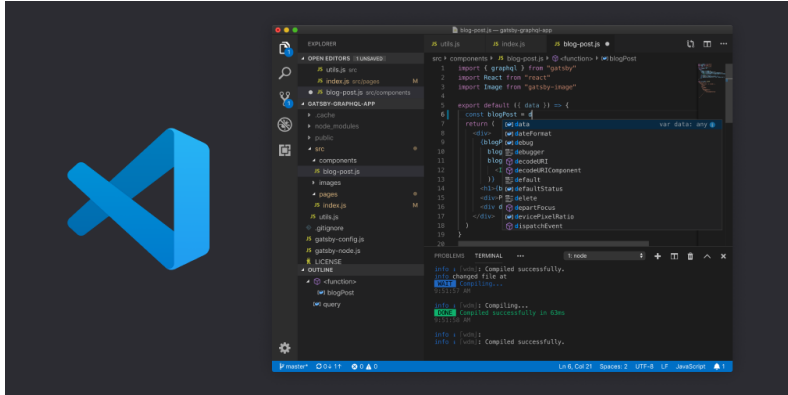
- ❖ Javascript adalah bahasa pemrograman yang awalnya dirancang untuk berjalan di atas browser.
- ❖ Namun, seiring perkembangan zaman, javascript tidak hanya berjalan di atas browser saja. Javascript juga dapat digunakan pada sisi Server, Game, IoT, Desktop, dan sebagainya.
- ❖ Javascript awalnya bernama Mocha, lalu berubah menjadi LiveScript saat browser Netscape Navigator 2.0 rilis versi beta (September 1995). Namun, setelah itu dinamai ulang menjadi Javascript.

Peralatan untuk Belajar Javascript



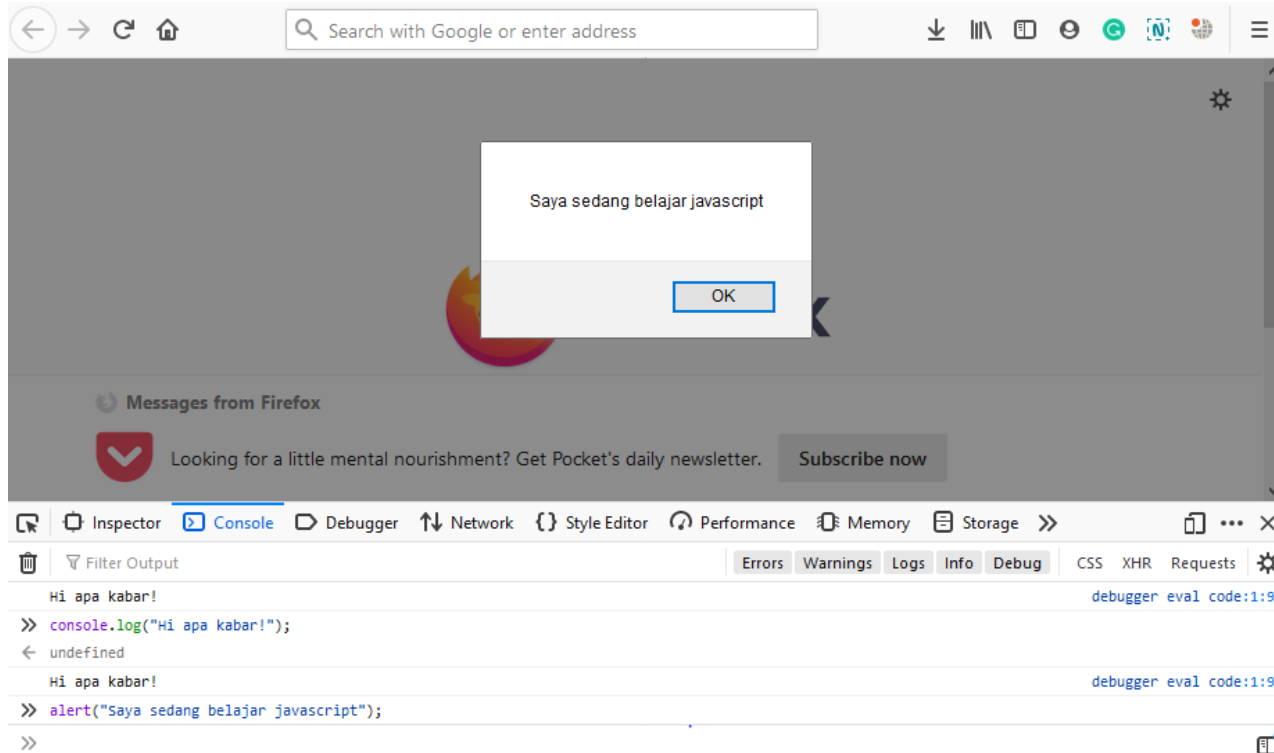
Apa saja peralatan yang harus disiapkan untuk belajar Javascript?

- ❖ Web Browser (Google Chrome, Firefox, Opera, dll)
- ❖ Teks Editor (rekomendasi: VS Code)



Mengenal Console JavaScript

- ❖ Console Javascript dapat di buka melalui Inspect Element ? Console.

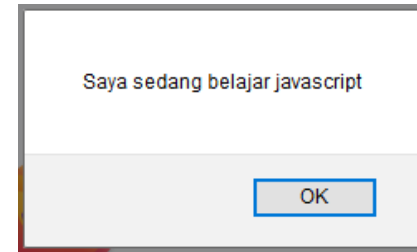
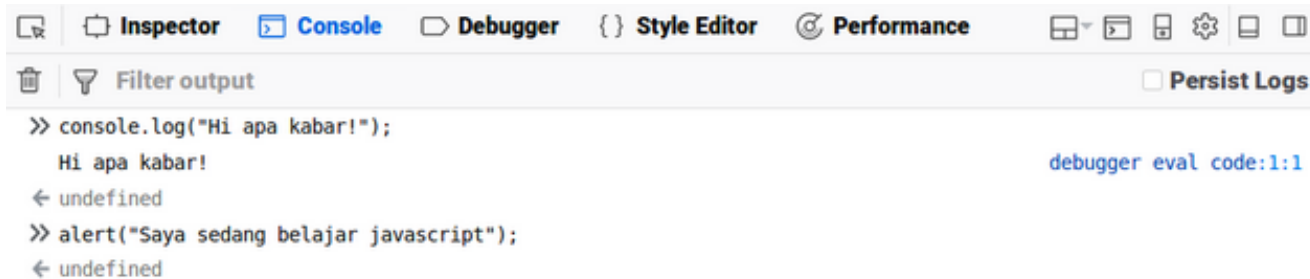


Mengenal Console JavaScript

- ❖ Di dalam console, kita bisa menulis fungsi atau kode-kode javascript dan hasilnya akan langsung ditampilkan.
- ❖ Misalnya, mari kita coba kode berikut:

```
console.log("Hi apa kabar!");  
alert("Saya sedang belajar javascript");
```

- ❖ Maka hasilnya:



Membuat Program Javascript

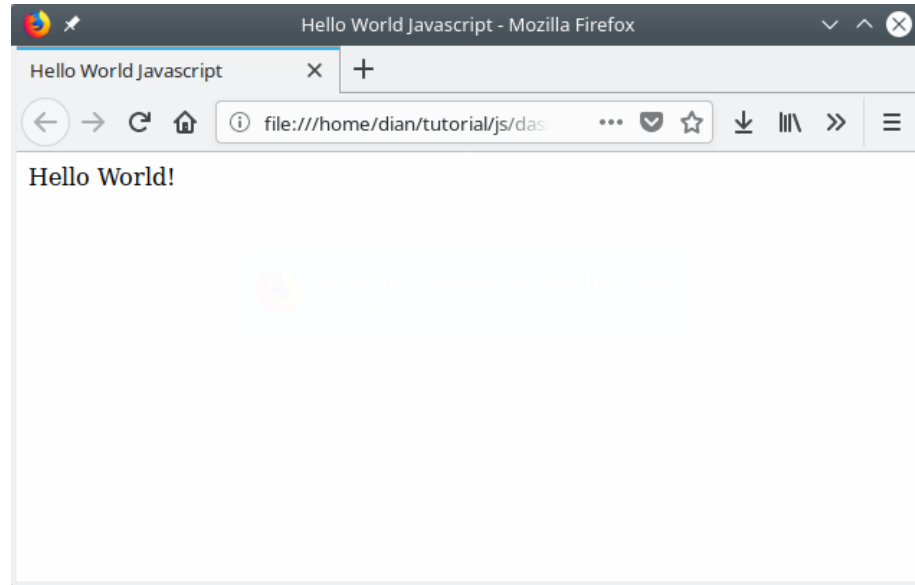
Buka teks editor, kemudian buat file baru bernama hello_world.html dan isi dengan kode berikut:

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

Membuat Program Javascript

Simpan dengan nama `hello_world.html`, kemudian buka file tersebut dengan web browser.

Maka hasilnya:



Membuat Program Javascript



Tunggu sebentar...
Tadi kita menulis perintah:

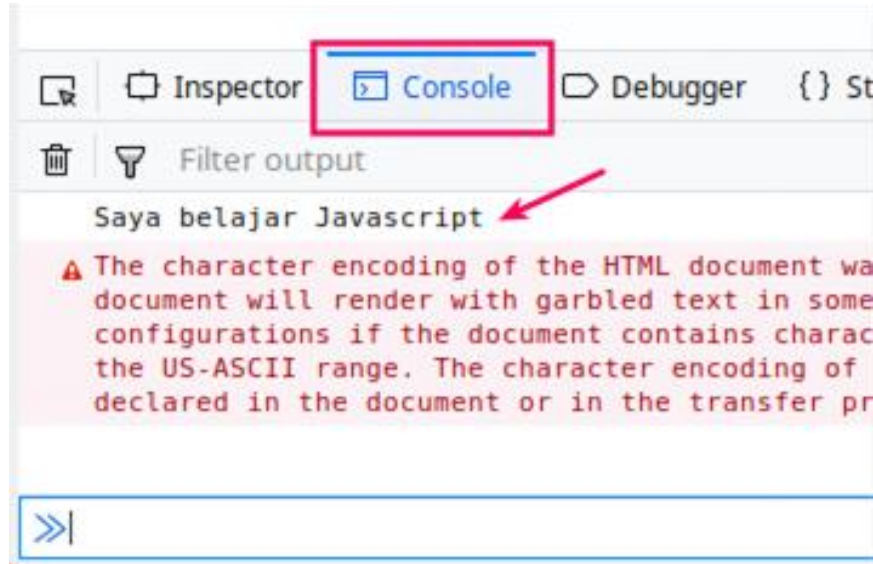
```
console.log("Saya belajar Javascript");
```

Mengapa tidak ditampilkan?

Membuat Program Javascript

Karena perintah atau fungsi `console.log()` akan menampilkan pesan ke dalam console javascript. Sedangkan perintah `document.write()` berfungsi untuk menulis ke dokumen HTML.

Buka console javascript.
Maka kita akan melihat pesan:
"Saya Belajar Javascript"



Cara Menulis Kode Javascript di HTML



Pada contoh di atas, kita sudah menulis kode javascript di dalam HTML. Cara tersebut merupakan cara penulisan *embedded* (ditempel). Masih ada beberapa cara lagi yang perlu kita ketahui:

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)

Penulisan Kode javascript dengan Embed

Pada cara ini, kita menggunakan tag script untuk menempelkan (embed) kode Javascript pada HTML. Tag ini dapat ditulis di dalam tag head dan body.

Contoh:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Belajar Javascript dari Nol</title>
    <script>
      // ini adalah penulisan kode javascript
      // di dalam tag <head>
      console.log("Hello JS dari Head");
    </script>
  </head>
  <body>
    <p>Tutorial Javascript untuk Pemula</p>
    <script>
      // ini adalah penulisan kode javascript
      // di dalam tag <body>
      console.log("Hello JS dari body");
    </script>
  </body>
</html>
```

Penulisan Kode javascript Inline



Pada cara ini, kita akan menulis kode javascript di dalam atribut HTML. Cara ini biasanya digunakan untuk memanggil suatu fungsi pada *event* tertentu. Misal: saat link diklik.

Contoh:

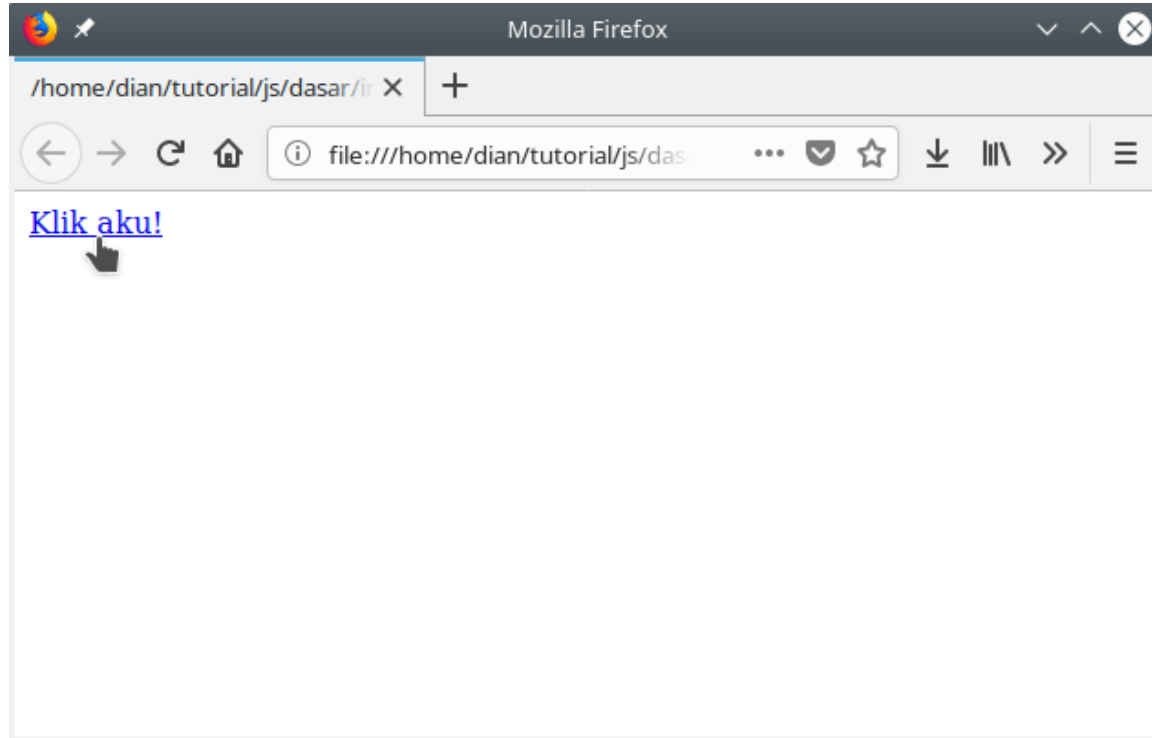
```
<a href="#" onclick="alert('Yey!')">Klik aku!</a>
```

atau bisa juga seperti ini:

```
<a href="javascript:alert('Yey!')">Klik aku!</a>
```

Penulisan Kode javascript Inline

Hasilnya:



Penulisan Kode javascript Inline



Perhatikan,

Atribut onclick merupakan atribut HTML untuk menyatakan fungsi yang akan dieksekusi saat elemen itu diklik.

Pada contoh sebelumnya, kita menjalankan fungsi `alert()`. Fungsi ini merupakan fungsi untuk menampilkan dialog.

Lalu pada atribut href, kita juga memanggil fungsi `alert()` dengan didahului javascript:.

Atribut href sebenarnya digunakan untuk mengisi alamat link atau URL.

Karena kita ingin memanggil kode javascript di sana, maka alamat link tersebut kita ubah menjadi javascript: lalu diikuti dengan fungsi yang akan dipanggil.

Penulisan Kode javascript Eksternal



Pada cara ini, kita akan menulis kode javascript secara terpisah dengan file HTML.

Cara ini biasanya digunakan pada proyek-proyek besar, karena diyakini—dengan cara ini—dapat lebih mudah mengelola kode project.

buat dua file, yaitu: file HTML dan Javascript :

```
belajar-js/  
├── kode-program.js  
└── index.html
```

Penulisan Kode javascript Eksternal

Isi file kode-program.js:

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

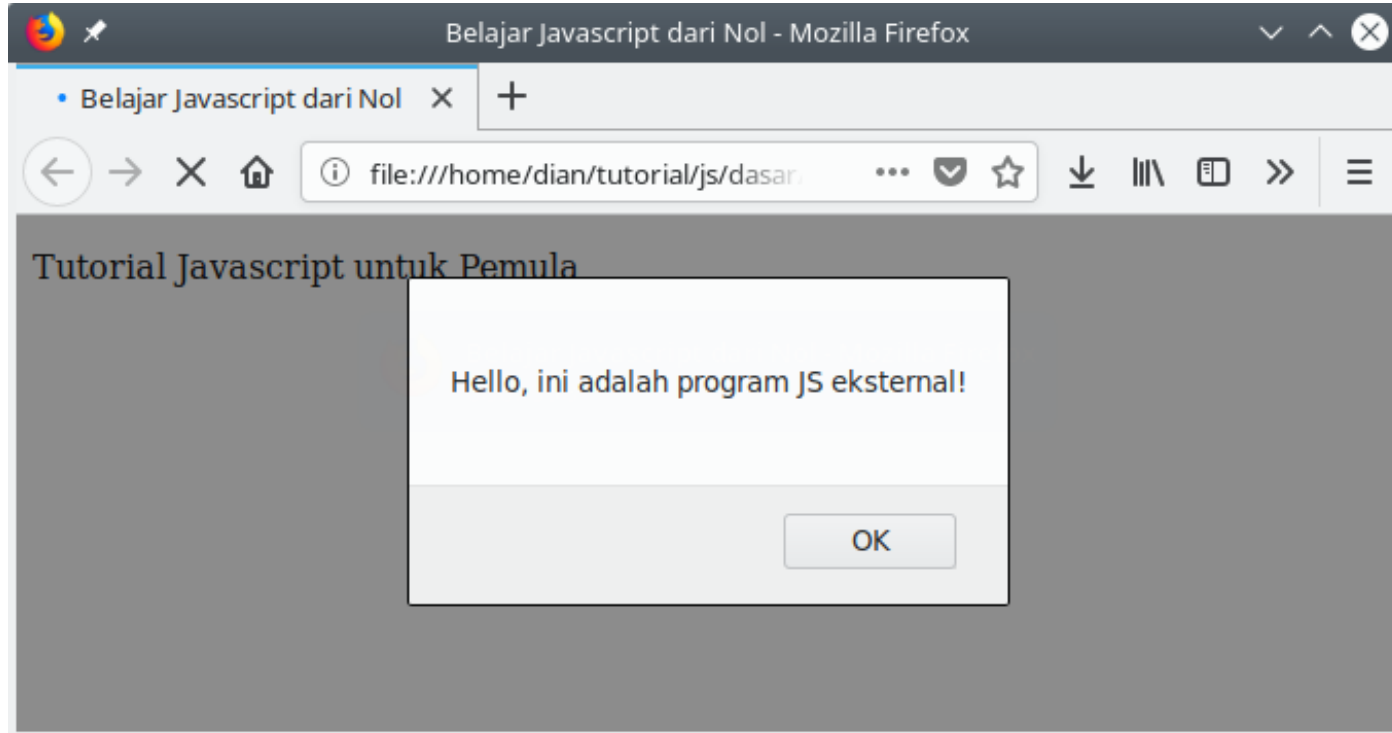
Isi file index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Belajar Javascript dari Nol</title>
  </head>
  <body>
    <p>Tutorial Javascript untuk Pemula</p>

    <!-- Menyisipkan kode js eksternal -->
    <script src="kode-program.js"></script>
  </body>
</html>
```

Penulisan Kode javascript Eksternal

Hasilnya:



Penulisan Kode javascript Eksternal



Pada contoh tadi, kita menulis kode javascript terpisah dengan kode HTML.

Lalu, pada kode HTML Kita menyisipkannya dengan memberikan atribut src pada tag script.

```
<!-- Menyisipkan kode js eksternal -->  
<script src="kode-program.js"></script>
```

Maka apapun yang ada di kode-program.js akan dapat dibaca dari file index.html

Penulisan Kode javascript Eksternal



Bagaimana kalau file javascriptnya berada di folder yang berbeda?

Kita bisa menuliskan alamat lengkap foldernya.

Contoh:

Misal kita punya struktur folder seperti ini:

```
belajar-js/  
├── js/  
│   └── kode-program.js  
└── index.html
```

Penulisan Kode javascript Eksternal



Maka untuk menyisipkan file kode-program.js ke dalam HTML, kita bisa menuliskannya seperti ini:

```
<script src="js/kode-program.js"></script>
```

Karena file kode-program.js berada di dalam direktori js. Kita juga bisa menyisipkan javascript yang ada di internet dengan memberikan alamat URL lengkapnya. Contoh:

```
<script src="https://www.petanikode.com/js/kode.js"></script>
```

Penulisan Kode javascript HTML



Javascript sebagai bahasa pemrograman yang berjalan di atas browser harus ditulis di dalam HTML.

Ada empat cara penulisan javascript pada HTML.

1. Tag **script**

Cara yang umum yang dipakai adalah menuliskannya dalam tag **script**. Tag **script** bisa dibuat di dalam tag **head**, maupun di dalam tag **body**. Contoh:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Penulisan Javascript</title>
  <script>
    console.log("Hi, ini kode Javascript");
  </script>
</head>
<body>

  <script>
    document.write("Javascript itu keren!");
  </script>
</body>
</html>
```

2. File Eksternal

Bila tidak ingin kode Javascript bercampur aduk dengan HTML, kita bisa menuliskannya pada file terpisah. Caranya, buatlah sebuah file yang berekstensi .js, misalnya isinya sebagai berikut.

```
// file-eksternal.js  
alert("Kode Javascript dari File Eksternal");
```

Penulisan JavaScript pada HTML

Kemudian, kita perlu menghubungkan file eksternal tersebut dengan file HTML. Caranya, gunakan tag **script** dengan atribut **src** untuk menentukan lokasi file Javascriptnya.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Penulisan Javascript</title>

<script src="file-eksternal.js"></script>

</head>
<body>

</body>
</html>
```

3. Atribut Event

Cara yang ketiga ini sering digunakan untuk memanggil fungsi pada event tertentu. Misalnya ketika suatu elemen diklik, maka jalankan fungsi Javascript.

```
<button onclick="alert('Ok Terima kasih!')">Klik donk!</button>
```

Selain *event onclick* ada juga *event-event* yang lain, seperti *onsubmit*, *onload*, *ondoubleclick*, *onmouseover*, *onmouseout*, dsb.

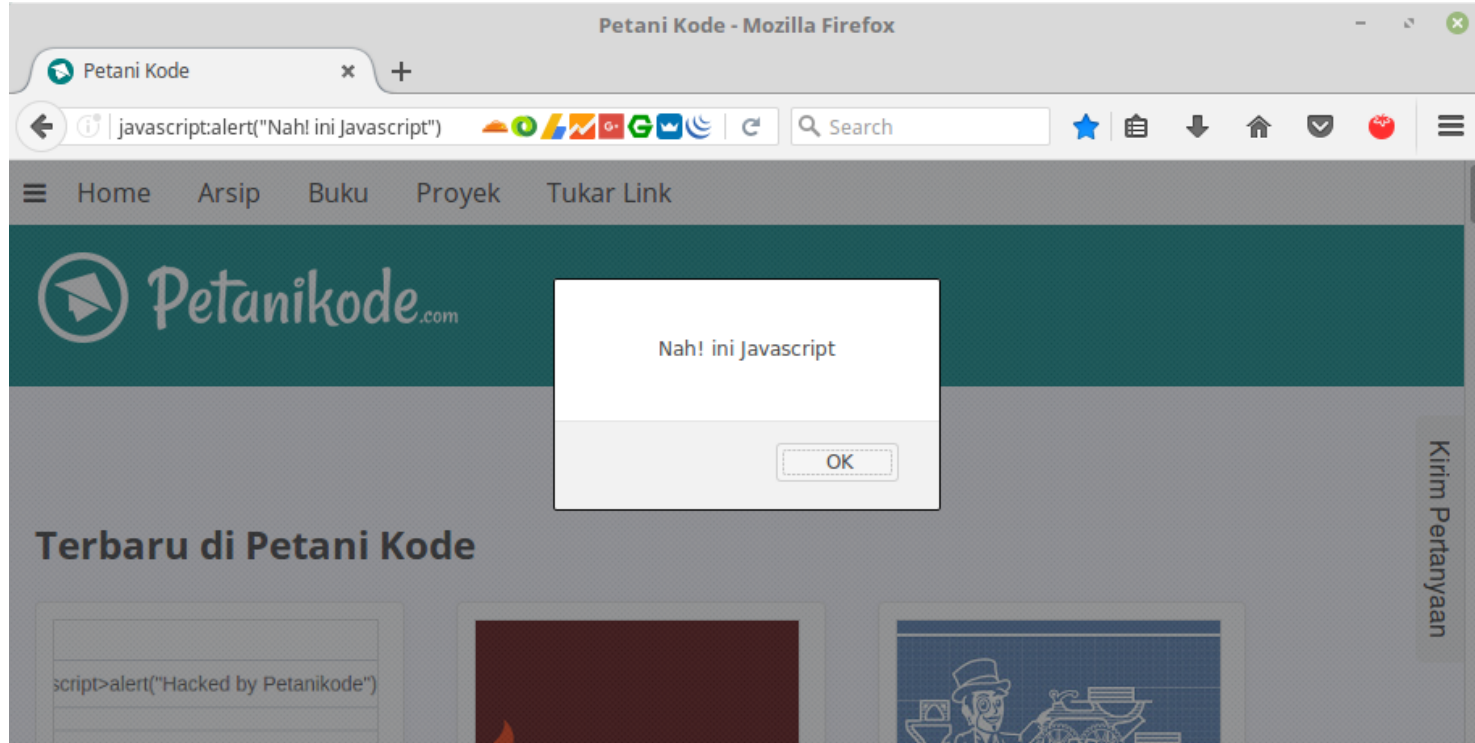
4. URL

Terakhir, penulisan Javascript pada URL. Cara ini jarang digunakan, namun kita juga perlu mengetahuinya. Penulisan Javascript pada URL menggunakan protokol Javascript. Misal, cobalah untuk menuliskan kode seperti ini pada URL browser.

```
javascript:alert("Nah! ini Javascript")
```

Hasilnya, Javascript akan dieksekusi browser.

Output



Lalu, bagaimana kita menggunakan cara ini di HTML?

Penulisan JavaScript pada HTML

Cara ini bisa kita gunakan pada tag **a**, kemudian mengisi kode javascript pada atribut href. Cara ini bisa menggantikan event onclick.

```
<a href="javascript:alert('Wih! hebat bukan?')">Klik Aku</a>
```

Menampilkan Output

Menampilkan Output Javascript



Output adalah sebuah tampilan program yang biasanya digunakan untuk memperlihatkan hasil akhir.

Output biasanya ditampilkan dalam bentuk teks dengan fungsi `print()`.

Ada 4 cara menampilkan output pada Javascript:

1. Fungsi `console.log()`
2. Fungsi `alert()`
3. Fungsi `document.write()`
4. `innerHTML`

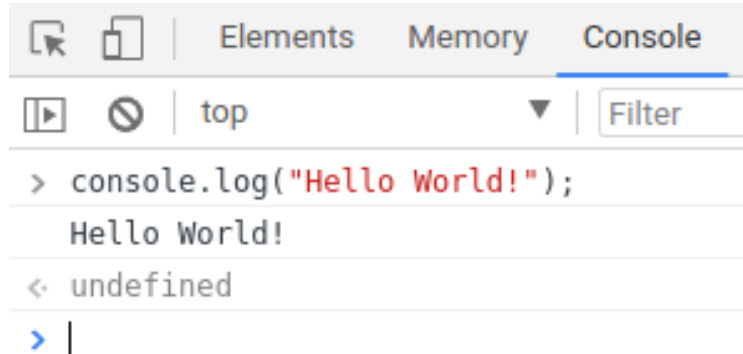
Fungsi `console.log()`

Fungsi `console.log()` adalah fungsi untuk menampilkan teks ke console Javascript.

Contoh penggunaan:

```
console.log("Hello World!");
```

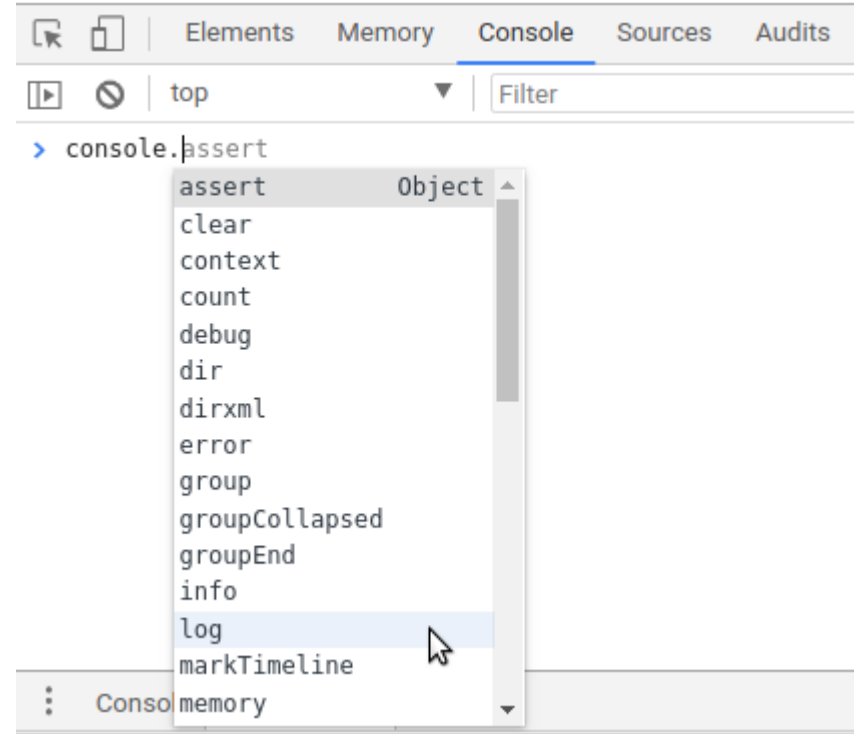
Hasil:



Menampilkan Output Javascript

Fungsi `console.log()` biasanya digunakan untuk debugging. Karena setiap pesan error di Javascript selalu ditampilkan di dalam Console.

Selain `console.log()`, terdapat juga beberapa fungsi untuk debugging seperti `console.debug()`, `console.info()`, `console.error()`, `console.dir()`, dsb.



Menampilkan Output Javascript

Fungsi `alert()` adalah fungsi untuk menampilkan jendela dialog. Fungsi sebenarnya berada pada objek window.

Secara lengkap bisa ditulis seperti ini:

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

Bisa juga ditulis `alert()` saja seperti ini:

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

Fungsi `alert()`, hanya bisa digunakan di dalam browser saja. Sedangkan pada Nodejs fungsi ini tidak ada.

Menampilkan Output Javascript

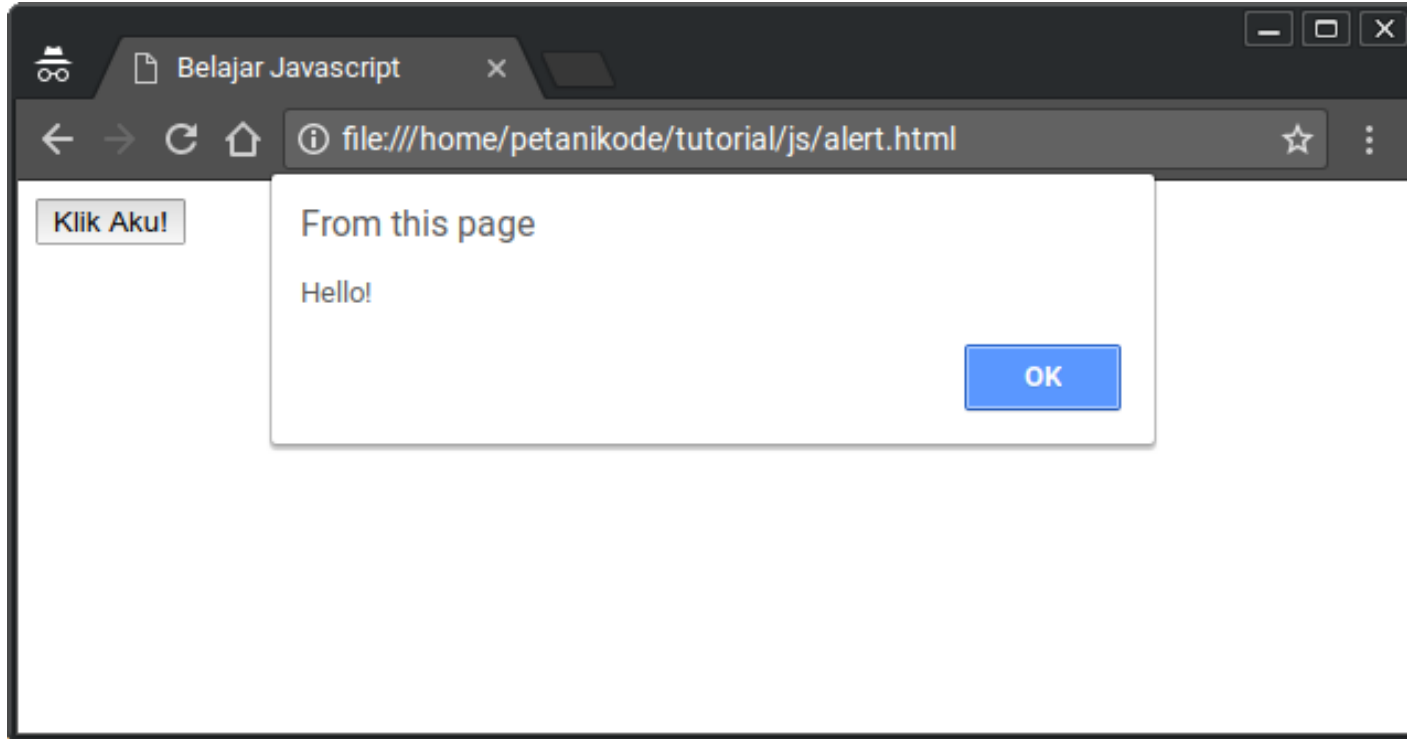
Contoh penggunaan fungsi **alert()**:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Belajar Javascript</title>
  <script>
    alert("Selamat datang di tutorial belajar Javascript");

    function sayHello(){
      alert("Hello!");
    }
  </script>
</head>
<body>
  <button onclick="sayHello()">Klik Aku!</button>
</body>
</html>
```


Menampilkan Output Javascript

Hasil:



Fungsi `document.write()`

Objek document adalah objek yang mewakili dokumen HTML di dalam Javascript.

Dalam objek document, terdapat fungsi `write()` untuk menulis sesuatu ke dokumen HTML.

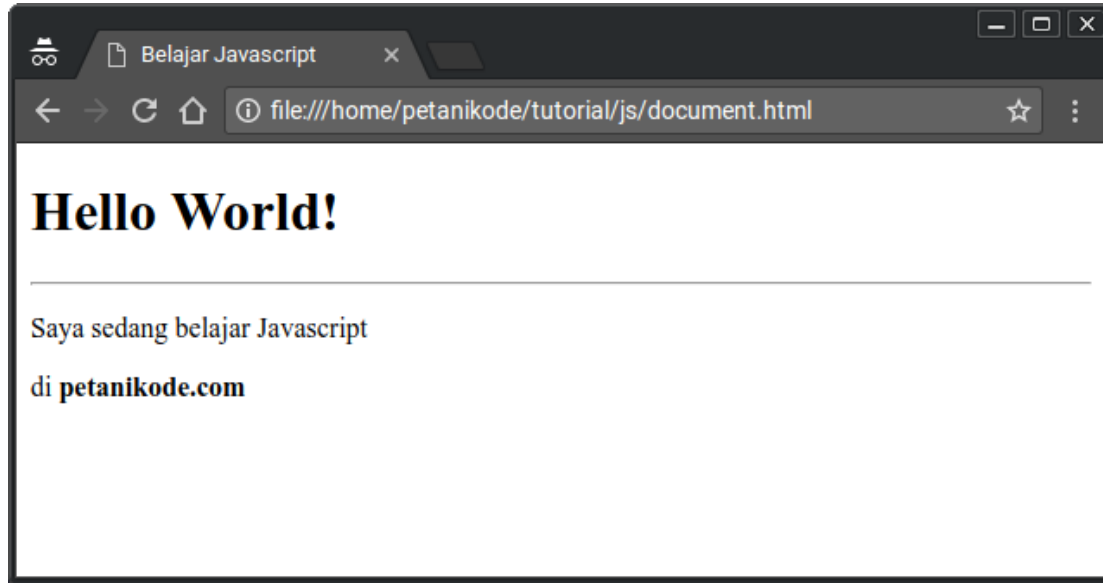
Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Belajar Javascript</title>
  <script>
    document.write("<h1>Hello World!</h1>");
    document.write("<hr>");
    document.write("<p>Saya sedang belajar Javascript</p>");
    document.write("di <b>petanikode.com</b>")
  </script>
</head>
<body>

</body>
</html>
```

Menampilkan Output Javascript

Hasil:



Selain fungsi `write()`, objek document juga menyediakan berbagai macam fungsi untuk manipulasi dokumen HTML.

Menggunakan **innerHTML**

innerHTML adalah sebuah atribut di dalam (objek) elemen HTML yang berisi string HTML.

Dengan innerHTML, kita dapat menampilkan output ke elemen yang lebih spesifik. Contoh:

```
<!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>
```

Menampilkan Output Javascript

Hasil:



Pengenalan

Variabel adalah sebuah nama yang mewakili sebuah nilai. Variabel bisa diisi dengan berbagai macam nilai seperti string (teks), number (angka), objek, array, dan sebagainya.

Cara Membuat Variabel di 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";
```

Pada contoh di atas, kita membuat variabel bernama title dengan nilai berupa teks (string): "Belajar Pemrograman Javascript".

Cara Membuat Variabel di Javascript



Pada contoh sebelumnya, kita menggunakan huruf besar atau kapital untuk nama variabel yang terdiri dari dua suku kata. Pada Javascript kita dianjurkan menggunakan camelCase dalam penamaan. Boleh-boleh saja menggunakan *snake case* atau *underscore* seperti ini:

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

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

Cara Membuat Variabel di Javascript



Perlu diketahui juga, selain kata kunci var kita juga bisa membuat variabel dengan kata kunci let atau tanpa awalan apapun.

Contoh:

```
// membuat variabel dengan kata kunci let  
let price = 15000;
```

```
// membuat variabel tanpa awalan apapun  
stock = 12
```

Cara Membuat Variabel di Javascript



Variable akan bernilai **undefined** bila tidak diisi
Contoh:

```
var x;
```

Maka variabel x akan bernilai **undefined**

Menampilkan isi Variabel

Untuk menampilkan isi variabel, kita bisa memanfaatkan fungsi-fungsi untuk menampilkan output seperti:

- Fungsi `console.log()` menampilkan output ke console javascript;
- Fungsi `document.write()` menampilkan output ke dokumen HTML;
- dan Fungsi `alert()` menampilkan output ke jendela dialog.

Contoh variable javascript dalam HTML

- Buat file bernama **belajar-variabel.html**, lalu isi dengan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Belajar Variabel dalam Javascript</title>
  <script>
    // membuat variabel
    var name = "Petani Kode";
    var visitorCount = 50322;
    var isActive = true;
    var url = "https://www.petanikode.com";

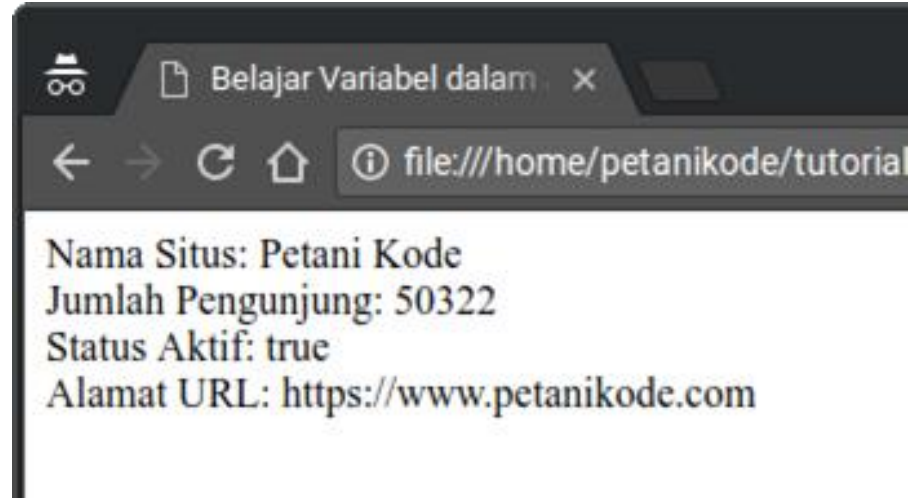
    // menampilkan variabel ke jendela dialog (alert)
    alert("Selamat datang di " + name);

    // 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>
```

Contoh variable javascript dalam HTML

- Buka dengan browser dan perhatikan hasilnya.
- Arti symbol plus (+) adalah menggabungkan, bukan menjumlahkan.



Mengisi Ulang Variabel

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

Contoh:

```
// mula-mula kita buat variabel dengan isi seperti ini  
var age = 18;  
  
// lalu kita isi ulang  
age = 21;
```

Menghapus Variabel

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

Contoh:

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

Mengenal Tipe Data

Tipe data adalah jenis-jenis data yang bisa kita simpan di dalam variabel.

Ada beberapa tipe data dalam pemrograman Javascript:

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

Mengenal Tipe Data

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

Mengenal Tipe Data

Contoh:

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

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

Pada contoh di atas variabel:

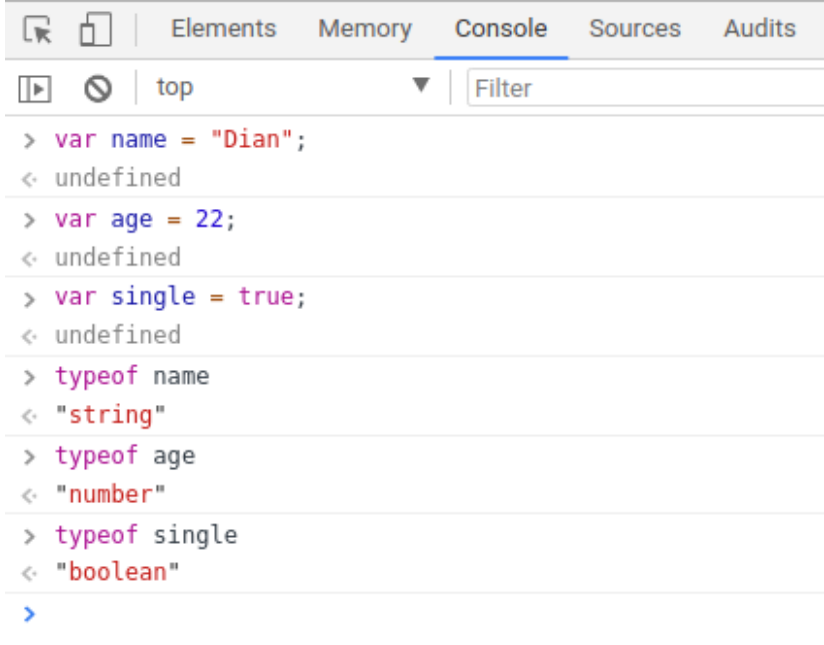
- name bertipe data String;
- age bertipe data integer;
- dan single bertipe data boolean.

Mengenal Tipe Data

Hal ini bisa juga kita cek dengan kata kunci **typeof**.
Contoh:

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

Hasil:



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

Aturan Penulisan Nama Variabel di Javascript

Ada beberapa aturan penulisan variabel dalam Javascript:

- Penamaan variabel tidak boleh menggunakan angka di depannya.

```
var 123nama = "Petanikode"
```

- 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 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()`;

1. Dialog Alert

Dialog `alert()` biasanya digunakan untuk menampilkan sebuah pesan peringatan atau informasi. Fungsi `alert()` berada dalam objek `window`. Kita bisa menggunakannya seperti ini:

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

Atau seperti ini:

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

Karena objek `window` berisifat global, kita boleh tidak menuliskannya.

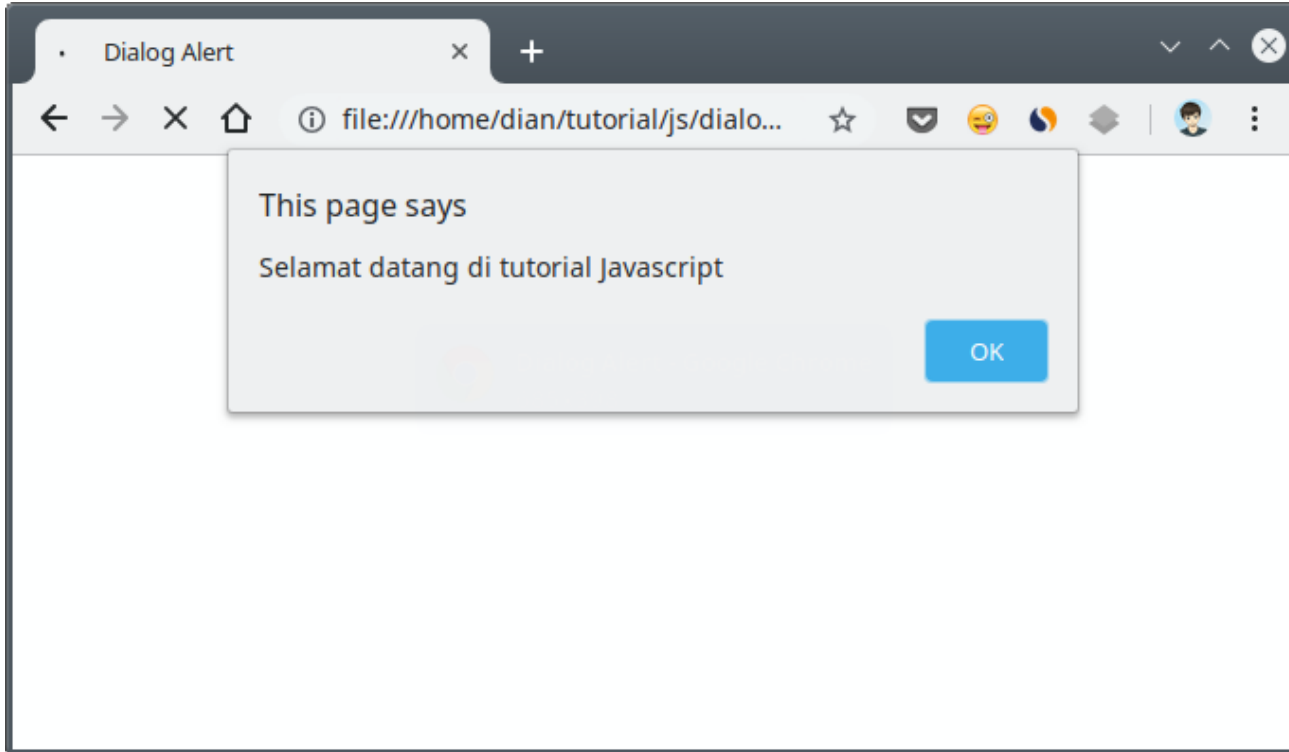
Dialog Alert

Dialog **alert()** tidak akan mengembalikan nilai apa-apa saat dieksekusi. Contoh:

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

Output Dialog Alert

Hasil:



Dialog Alert

Dialog alert() memiliki satu parameter yang harus diberikan, yaitu: teks yang akan ditampilkan pada dialog.

Pada contoh sebelumnya, kita memberikan teks "Selamat datang di tutorial Javascript". Dan kita bisa menampilkan dialog alert() pada event tertentu, misalnya saat sebuah tombol diklik.

Ini bisa kita lakukan dengan menambahkan fungsi dialog pada *event listener*. Pada HTML, kita bisa masukan fungsi alert() pada atribut `onClick` agar nanti ditampilkan saat sebuah elemen diklik.

Dialog Alert

Contoh:

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

2. Dialog Confirm

Dialog `confirm()` digunakan untuk melakukan konfirmasi dalam melakukan tindakan tertentu.

Dialog confirm dapat dibuat dengan fungsi `confirm()`. Contoh:

```
confirm("Apakah anda yakin akan menghapus?");
```

Dialog Confirm

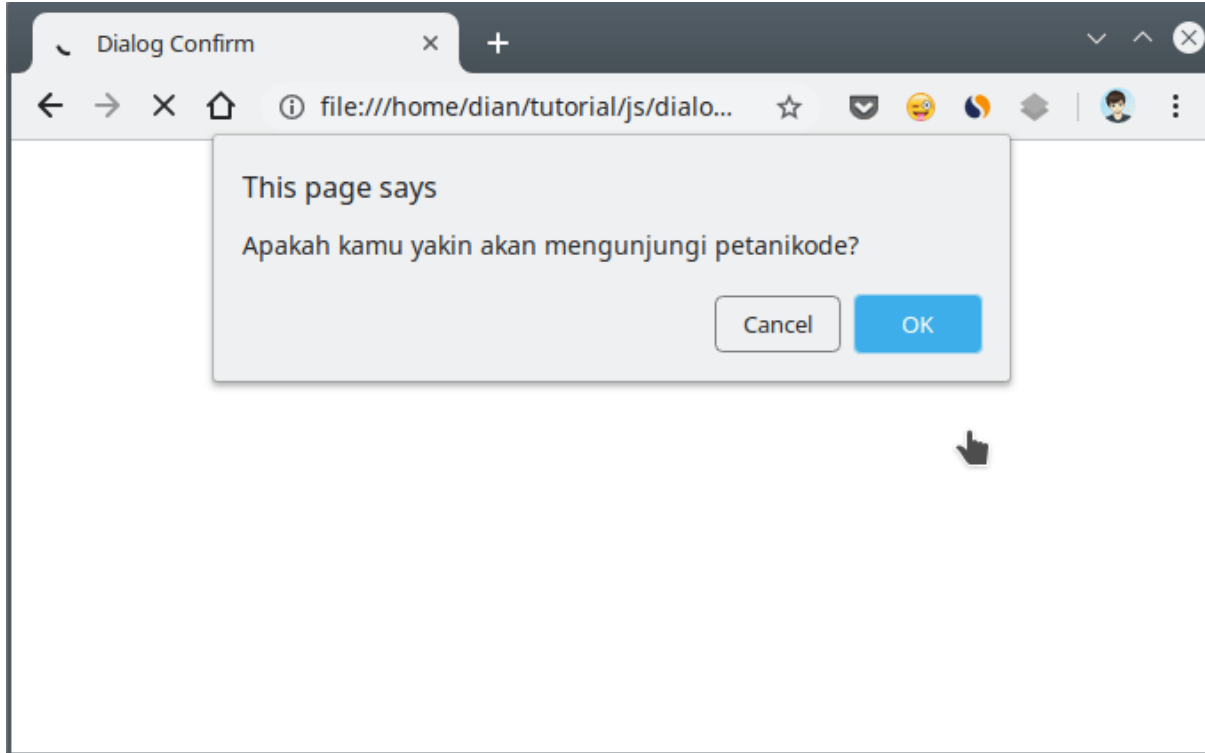
Dialog **confirm()** akan mengembalikan nilai true apabila kita memilih tombol OK dan akan mengembalikan nilai false apabila kita memilih Cancel. Nilai kembalian ini dapat kita tampung dalam variabel untuk diproses. Contoh:

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

      if (yakin) {
        window.location = "https://www.petanikode.com";
      } else {
        document.write("Baiklah, tetap di sini saja ya :)");
      }
    </script>
  </body>
</html>
```

Output Dialog Confirm

Hasil:



3. Dialog Prompt

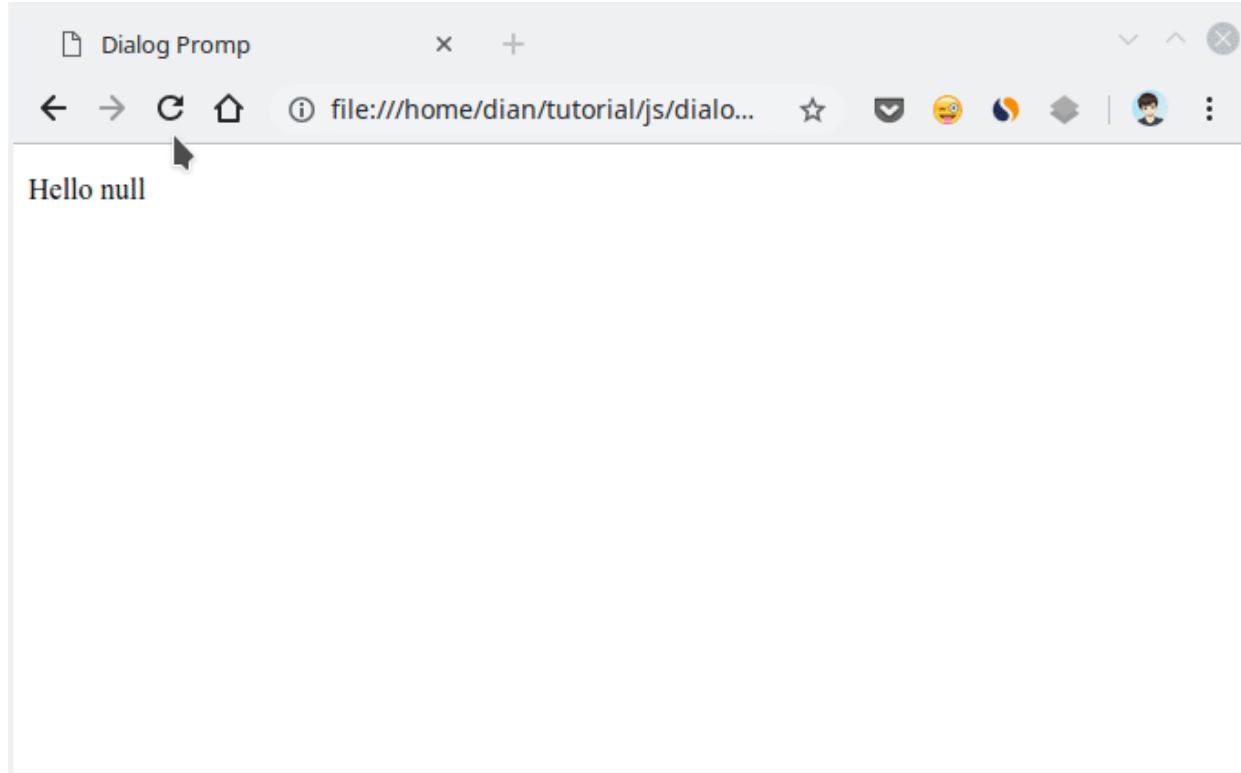
Dialog **prompt()** berfungsi untuk mengambil sebuah inputan dari pengguna.

Dialog **prompt()** akan mengembalikan sebuah nilai string dari apa yang diinputkan oleh pengguna.

```
<!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>
```

Output Dialog Prompt

Hasil:



Dialog Prompt

Dialog `prompt()` memiliki beberapa parameter yang harus diberikan:

1. Teks yang akan ditampilkan pada form;
2. Nilai default untuk field input.

Pada contoh di atas, kita memberikan nilai default-nya berupa string kosong dengan tanda petik "".

Waktu yang Tepat Menggunakan Alert, Confirm, dan Prompt

Berdasarkan contoh-contoh sebelumnya bisa diketahui kapan waktu yang tepat untuk menggunakan `alert()`, `confirm()` dan `prompt()`.

- Saat kita hanya ingin menampilkan informasi saja, maka gunakan `alert()`.
- Saat kita ingin jawaban konfirmasi dari pengguna, maka gunakan `confirm()`.
- apabila kita ingin mengambil data teks dari pengguna, maka gunakan `prompt()`.

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

1. Operator Aritmatika

Operator aritmatika merupakan operator untuk melakukan operasi aritmatika seperti penjumlahan, pengurangan, pembagian, perkalian, dsb.

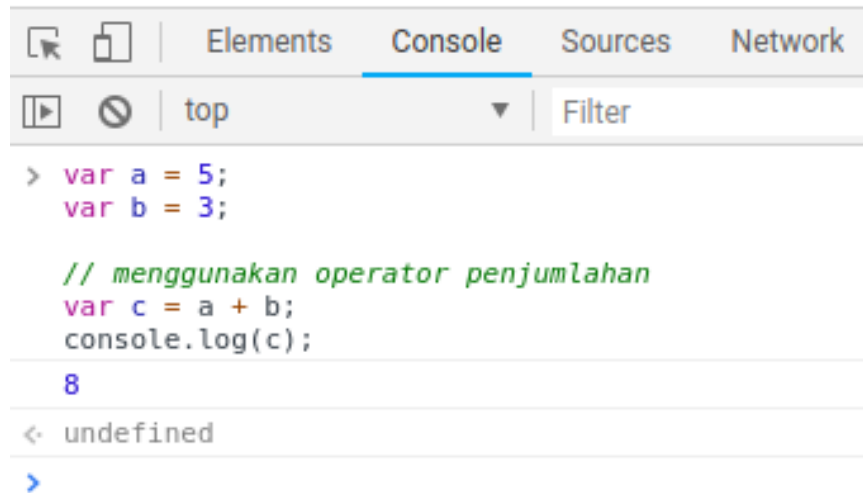
Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pemangkatan	**
Pembagian	/
Sisa Bagi	%

Operator Aritmatika

Contoh:

```
var a = 5;  
var b = 3;  
  
// menggunakan operator penjumlahan  
var c = a + b;  
console.log(c);
```

Hasil:



```
<img alt="back" data-bbox="500 285 525 315"/> <img alt="copy" data-bbox="535 285 560 315"/> | Elements Console Sources Network  
<img alt="run" data-bbox="500 350 525 380"/> <img alt="stop" data-bbox="535 350 560 380"/> | top Filter  
> var a = 5;  
   var b = 3;  
  
   // menggunakan operator penjumlahan  
   var c = a + b;  
   console.log(c);  
  
8  
< undefined  
>
```

Operator Penggabungan Teks

Pada Javascript, apabila kita akan melakukan operasi terhadap tipe data string atau teks menggunakan penjumlahan (+), maka yang akan terjadi adalah penggabungan; Bukan penjumlahan.

Contoh:

```
var a = "10" + "2";
```

Maka Hasilnya adalah "102" bukan 12 karena kedua angka tersebut merupakan string.

2. Operator Penugasan

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

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

Operator Penugasan

Contoh:

```
var jumlahView = 12;  
  
// menggunakan operator penugasan penjumlahan  
// untuk menambah nilai  
jumlahView += 1;
```

Hasilnya adalah variable jumlahView akan bertambah 1 atau bisa ditulis seperti ini:

```
jumlahView = jumlahView + 1;
```

Operator Penugasan

Khusus untuk operator penugasan yang dijumlahkan dan dikurangi dengan satu, bisa disingkat dengan ++ dan -- untuk pengurangan.

Contoh:

```
var a = 2;  
a++;
```

Maka variable a akan menjadi 3.

Operator Penugasan

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

Operator Penugasan

Contoh:

```
<!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 menjumlahkan dengan 5
    score += 5;
    document.write("score = "+ score + "<br/>");

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

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

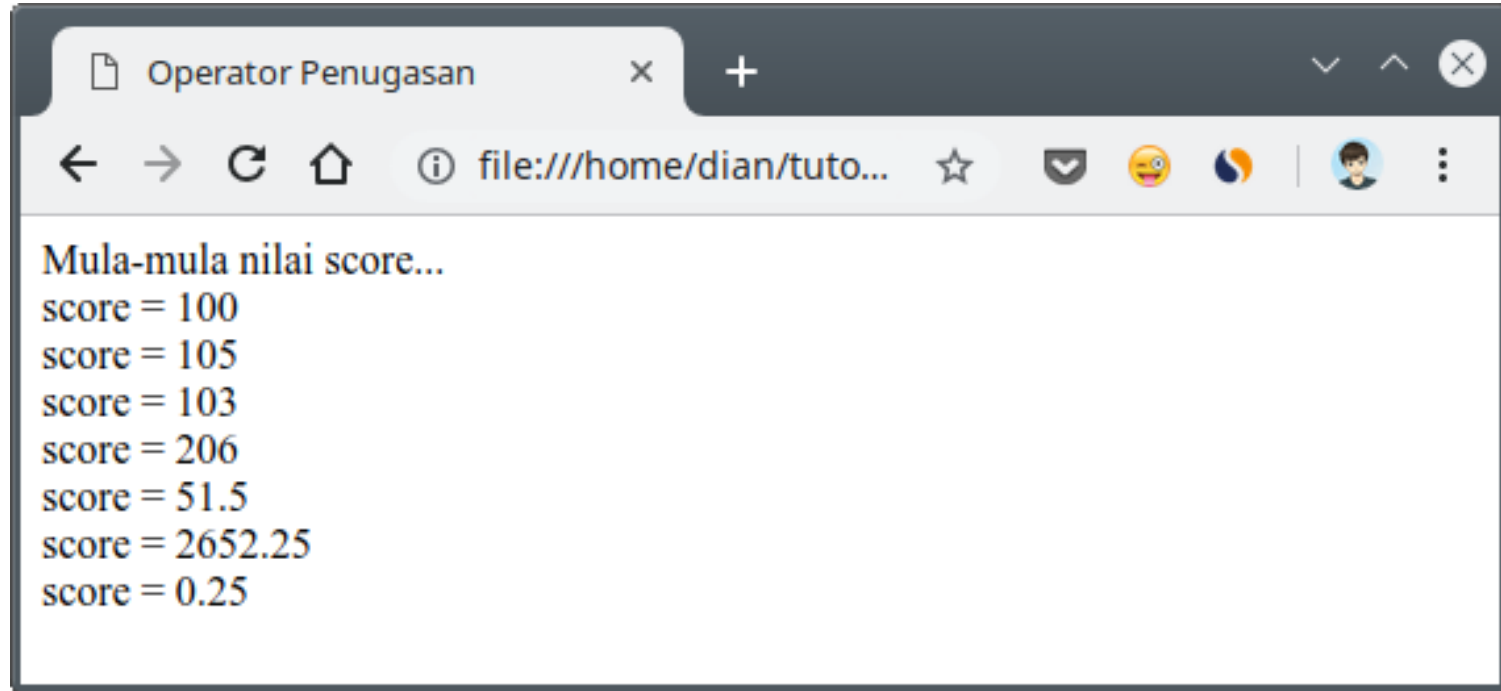
// pengisian dan pemangkatan dengan 2
score **= 2;
document.write("score = "+ score + "<br/>");

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

</html>
```

Output

Hasil:



The screenshot shows a web browser window with a single tab titled "Operator Penugasan". The address bar displays the file path "file:///home/dian/tuto...". The main content area of the browser shows the following text:

```
Mula-mula nilai score...  
score = 100  
score = 105  
score = 103  
score = 206  
score = 51.5  
score = 2652.25  
score = 0.25
```

3. Opearator Perbandingan

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

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

Contoh

```
<!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/>`);

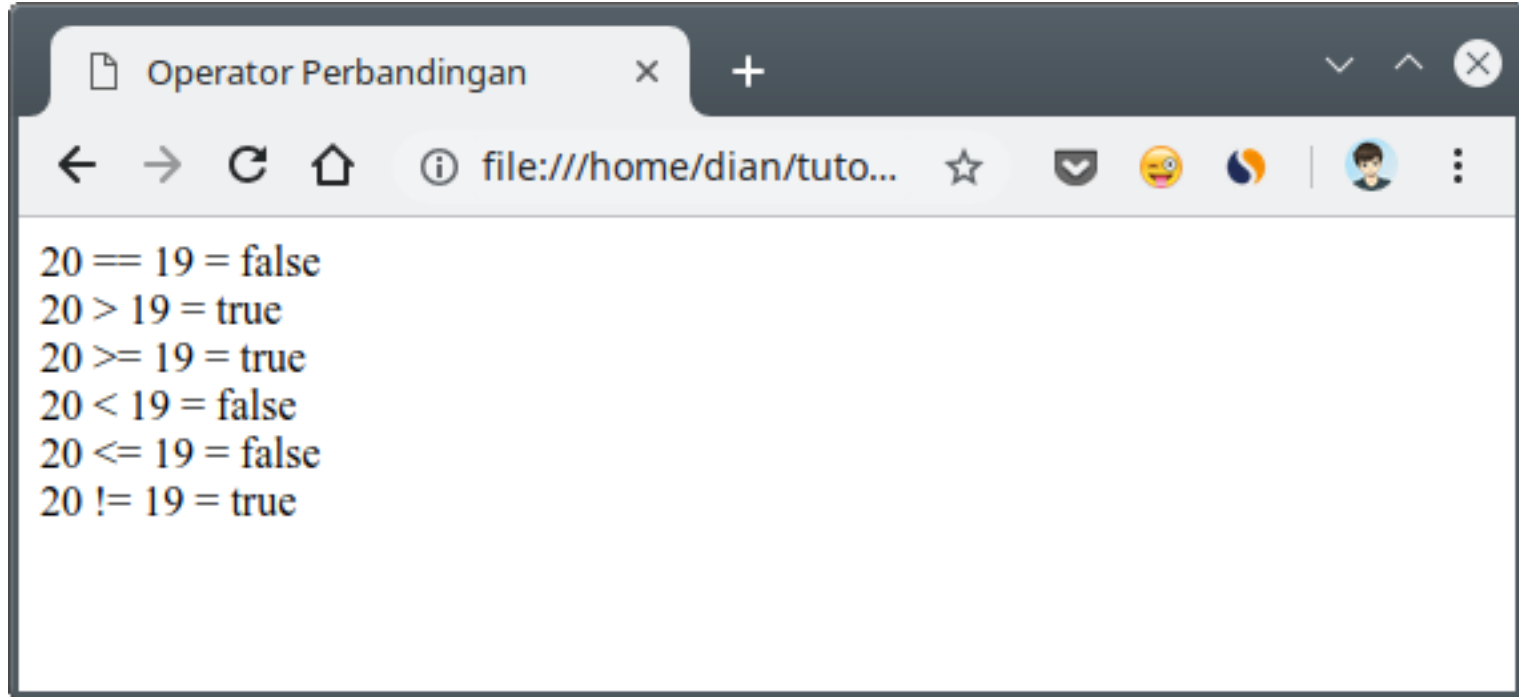
// lebih kecil samadengan
var hasil = aku <= kamu;
document.write(`${aku} <= ${kamu} = ${hasil}<br/>`);

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

</html>
```

Output

Hasil:



The screenshot shows a web browser window with a single tab titled "Operator Perbandingan". The address bar displays the file path "file:///home/dian/tuto...". The main content area lists the results of various comparison operations on the number 20:

```
20 == 19 = false
20 > 19 = true
20 >= 19 = true
20 < 19 = false
20 <= 19 = false
20 != 19 = true
```

Perbandingan

Menggunakan `===` akan membandingkan dengan tipe data.

Contoh:

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

4. Operartor Logika

Operator logika digunakan untuk melakukan operasi terhadap dua nilai boolean.

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

Contoh

```
<!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/>`);

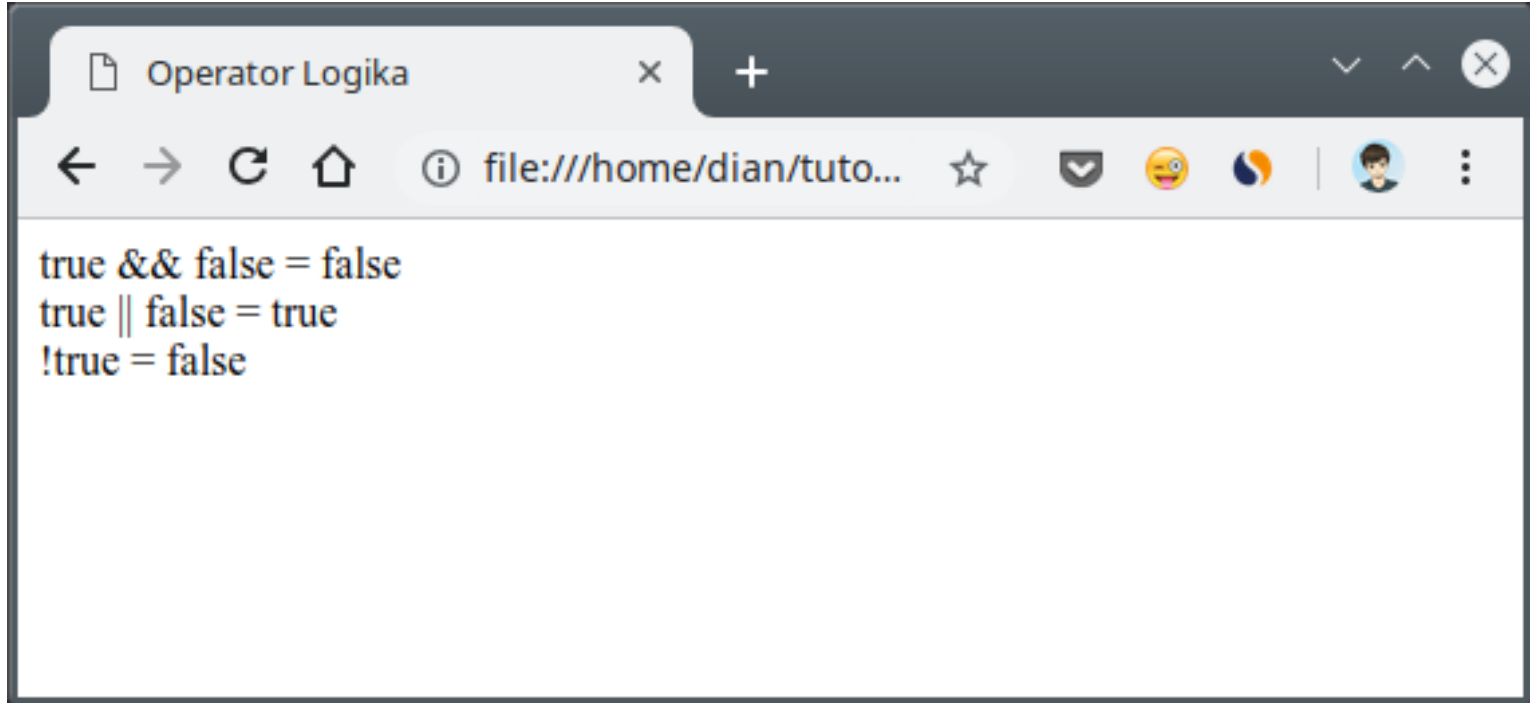
// operator ! (not)
var hasil = !benar
document.write(`${!benar} = ${hasil}<br/>`);

</script>
</body>

</html>
```

Output

Hasil:



A screenshot of a web browser window. The tab is titled "Operator Logika". The address bar shows the file path "file:///home/dian/tuto...". The main content area displays three lines of text: "true && false = false", "true || false = true", and "!true = false".

```
true && false = false  
true || false = true  
!true = false
```

5. Operator Bitwise

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

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

Operator Bitwise

Operator ini berlaku untuk tipe data int, long, short, char, dan byte. Operator ini akan menghitung dari bit-ke-bit.

Misalnya, kita punya variabel $a = 60$ dan $b = 13$.

Bila dibuat dalam bentuk biner, akan menjadi seperti ini:

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

Operator Bitwise

Kemudian, dilakukan operasi bitwise

AND

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

XOR

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

OR

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

NOT

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

Contoh

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

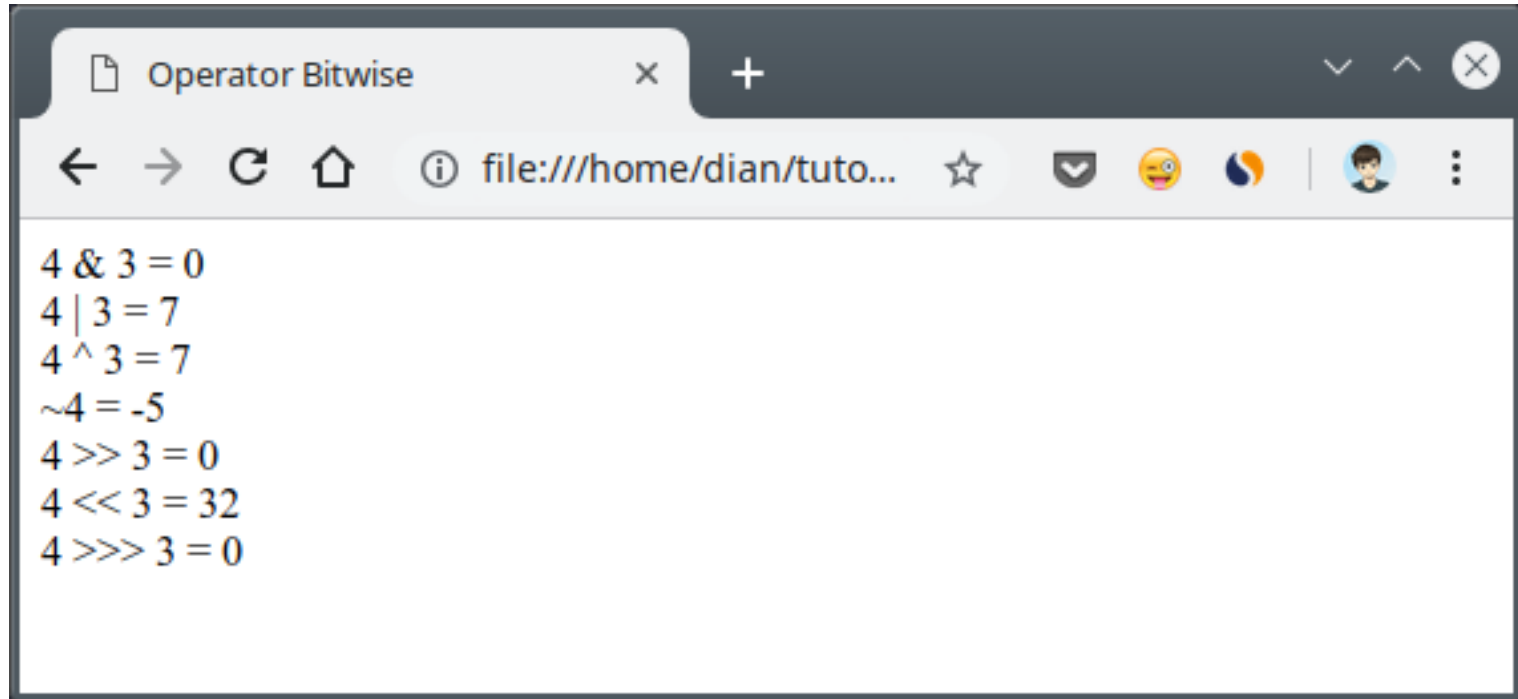
    // operator bitwise right shift (unsigned) >>>
    var hasil = x >>> y;
    document.write(`${x} >>> ${y} = ${hasil}<br/>`);

  </script>
</body>

</html>
```

Output

Hasil:



A screenshot of a web browser window with a single tab titled "Operator Bitwise". The address bar shows the file path "file:///home/dian/tuto...". The browser interface includes back, forward, refresh, and home buttons, as well as a search bar and several extension icons. The main content area displays the following bitwise operations and their results:

```
4 & 3 = 0
4 | 3 = 7
4 ^ 3 = 7
~4 = -5
4 >> 3 = 0
4 << 3 = 32
4 >>> 3 = 0
```

5. Operator Ternary

Operator ternary merupakan operator yang terdiri dari tiga bagian. bagian kiri, tengah, dan 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 berikut:

kondisi ? "benar" : "salah"

Perhatikan! kondisi dapat kita isi 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.

Contoh

```
<!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>
```

Kesimpulan Pertemuan

1. Kesimpulan materi 1
2. Kesimpulan materi 2
3. Kesimpulan materi 3
4. dst

Referensi

1. Referensi 1
2. Referensi 2
3. Referensi 3
4. dst

Tim Pengajar

1. Nama 1
2. Nama 2
3. Nama 3
4. dst



TERIMA KASIH