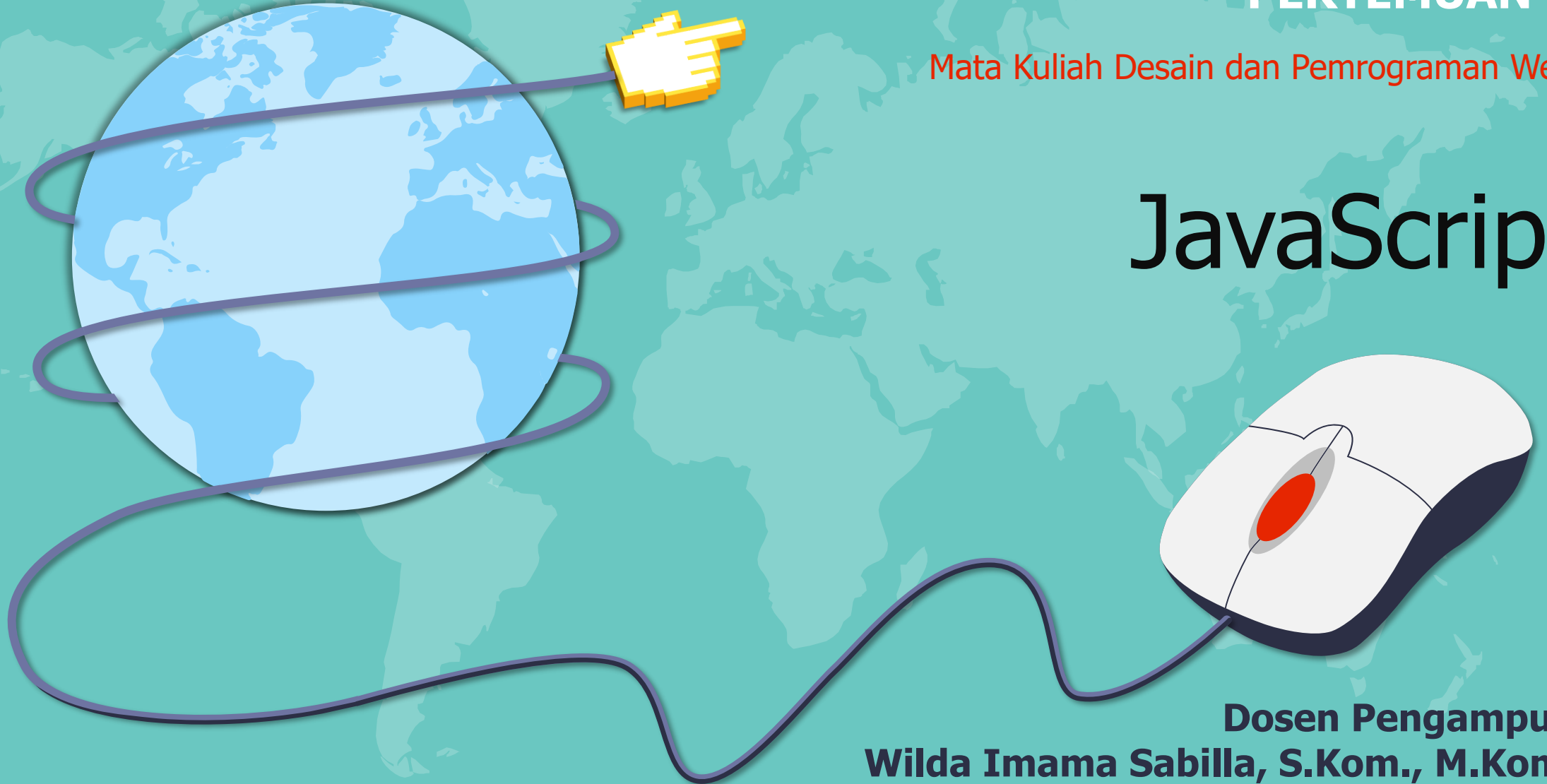


PERTEMUAN 3

Mata Kuliah Desain dan Pemrograman Web

JavaScript



Dosen Pengampu :
Wilda Imama Sabilla, S.Kom., M.Kom.

TOPIK PEMBELAJARAN

- 🌐 Pengenalan konsep dasar Javascript
- ☑ Tipe Data, Operator, Fungsi
- 🌐 Javascript di dalam HTML

CAPAIAN PEMBELAJARAN

- 📌 Menjelaskan konsep dasar CSS
- 📌 Menerapkan tipe data, operator, dan fungsi pada JavaScript
- 📌 Menerapkan JavaScript pada HTML

JavaScript

- **JavaScript** adalah bahasa pemrograman web yang bersifat *Client Side Programming Language* → pemrosesannya dilakukan oleh *client* → *web browser* (**Google Chrome, Mozilla Firefox, dll**).
- **JavaScript** pada awal perkembangannya berfungsi untuk membuat interaksi antara user dengan situs web menjadi lebih cepat tanpa harus menunggu pemrosesan di *web server*. Sebelum *javascript*, setiap interaksi dari user harus diproses oleh *web server*.
- Awalnya JavaScript berguna untuk *validasi form*, sekarang digunakan juga untuk:
 - animasi untuk mempercantik halaman web
 - fitur chatting
 - games, dsb
- Seiring perkembangannya, JavaScript juga bisa digunakan di sisi server dengan bantuan berbagai macam framework seperti Node.js, dan React.js.
- Selain itu JavaScript juga dapat digunakan untuk mengedit DOM (Document Object Model).

Menggunakan Javascript

1. Menuliskan Tag dengan diawali `<script type="text/javascript" >` dan diakhir dengan `</script>`
2. Menuliskan Tag dengan diawali `<script language="javascript" >` dan diakhir dengan `</script>`. Atribut ini digunakan dengan tujuan untuk menentukan versi, contoh `<script language="javascript1.2">`
3. Menuliskan Tag dengan diawali `<script language="javascript" type="text/javascript"` > dan diakhir dengan `</script>`

Menggunakan Javascript

Embed (Kode Javascript ditempel langsung pada HTML)

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```

HEAD

```
<!DOCTYPE html>
<html>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</body>
</html>
```

BODY

Menggunakan Javascript

Inline (Kode Javascript ditulis pada atribut HTML)

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

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

Menggunakan Javascript

Eksternal (Kode Javascript ditulis dengan terpisah dengan file HTML)

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>
```

JS

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

Display pada JavaScript

- Menulis pada elemen HTML → innerHTML
- Menulis pada output HTML → document.write()
- Menulis pada kotak peringatan (alert box) → window.alert()
- Menulis pada *browser console* → console.log()

```
1 <html>
2 <head>
3   <title>Belajar Javascript</title>
4 </head>
5 <body>
6 <script>
7   document.write("<h2>Welcome to Javascript</h2>");
8   console.log("Hello World!");
9 </script>
10 </body>
11 </html>
```

```
<script>
document.getElementById("demo").innerHTML
= 5 + 6;
</script>
```

INNER HTML

Menampilkan ke halaman HTML

Menampilkan ke console

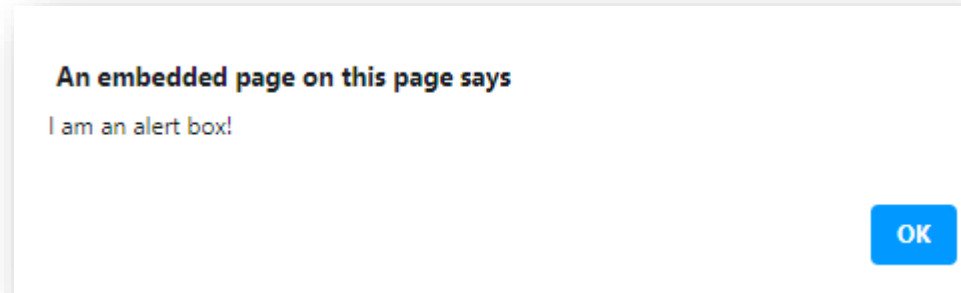
Jendela Dialog

Jendela dialog merupakan jendela yang digunakan untuk berinteraksi dengan pengguna.

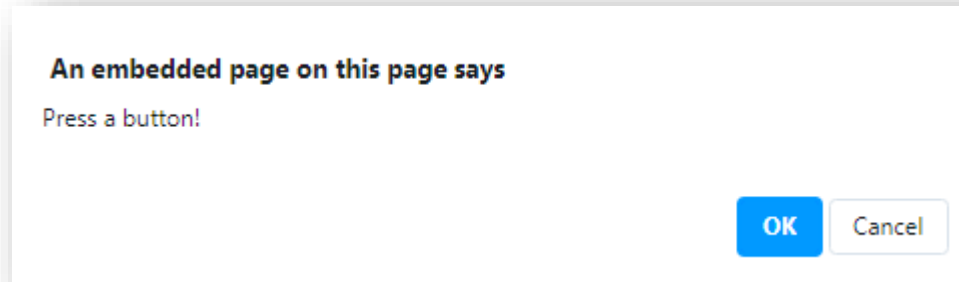
Ada tiga macam jendela dialog pada Javascript:

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

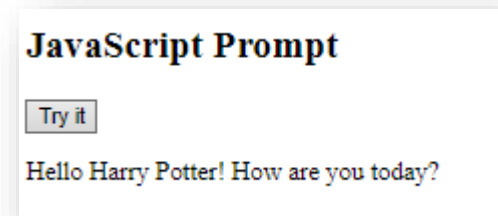
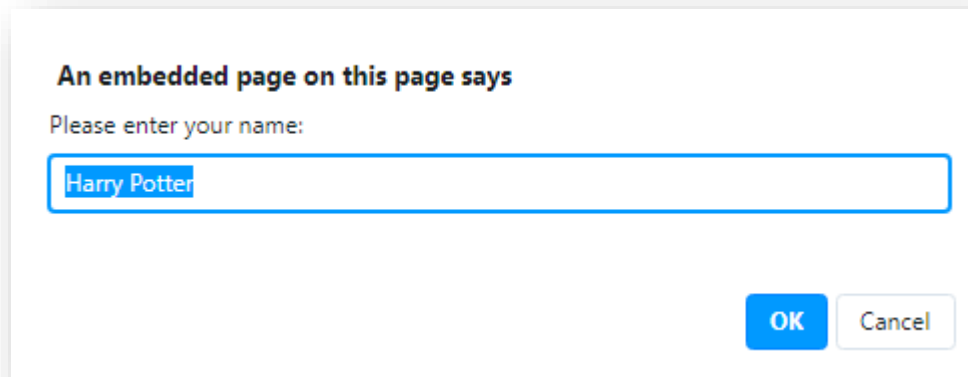
Jendela Dialog



ALERT



CONFIRM



PROMPT



Variabel

- Cara membuat variabel yang umum digunakan di javascript adalah menggunakan kata kunci var lalu diikuti dengan nama variabel dan nilainya.
- Penamaan variabel **tidak boleh** menggunakan angka di depannya.
`// salah var 123nama = "Polinema";`
- Penamaan variabel **boleh** menggunakan awal underscore.
- Penamaan variabel **dianjurkan** menggunakan camelCase apabila terdiri dari dua suku kata.
`var fullName = "Polinema";`
- Penamaan variabel **dianjurkan** menggunakan bahasa inggris

```
var title = "Belajar Pemrograman Javascript";  
var person = "John Doe", carName = "Volvo", price = 200;  
var x = "John" + " " + "Doe";
```





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
- 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*.



Operator

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

```
var x = 5;  
var y = 2;  
var z = x * y;
```

```
var txt1 = "John";  
var txt2 = "Doe";  
var txt3 = txt1 + " " + txt2;
```




Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

Fungsi

- Fungsi adalah sub-program yang bisa digunakan kembali baik di dalam program itu sendiri, maupun di program yang lain. Fungsi JavaScript akan dieksekusi ketika dipanggil (*invokes*)

nama fungsi



```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

- Dimulai dengan *keyword* **function**
- Parameter berada di dalam tanda kurung ()
- Kode berada di dalam kurung kurawal { }

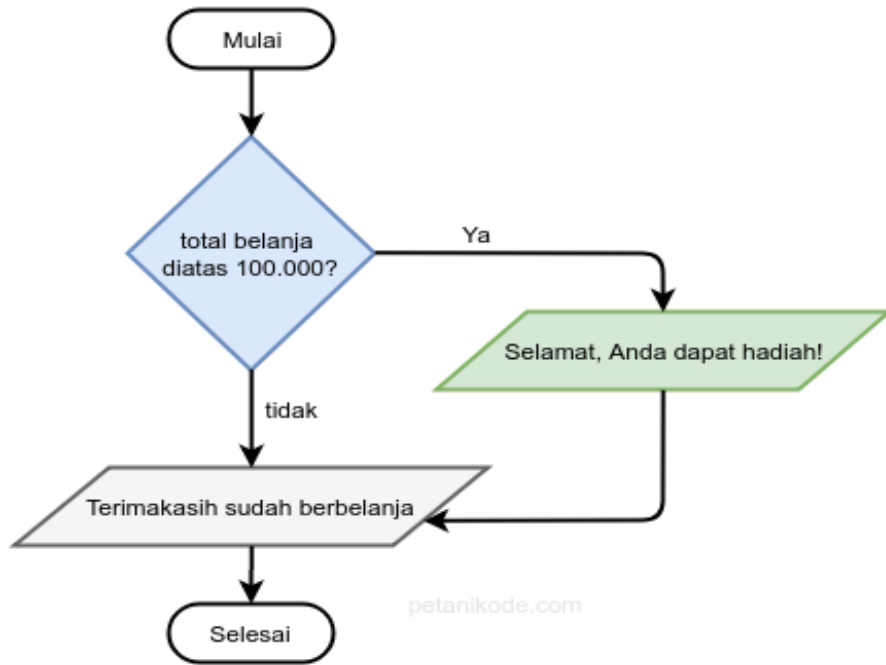
Percabangan

Digunakan untuk menjalankan program ketika ditentukan beberapa kondisi.

- Gunakan **if** untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan adalah true
- Gunakan **else** untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sama adalah false
- Gunakan **else if** untuk menentukan kondisi baru yang akan diuji, jika kondisi pertama adalah false
- Gunakan **switch** untuk menentukan banyak blok kode alternatif yang akan dieksekusi

Percabangan IF

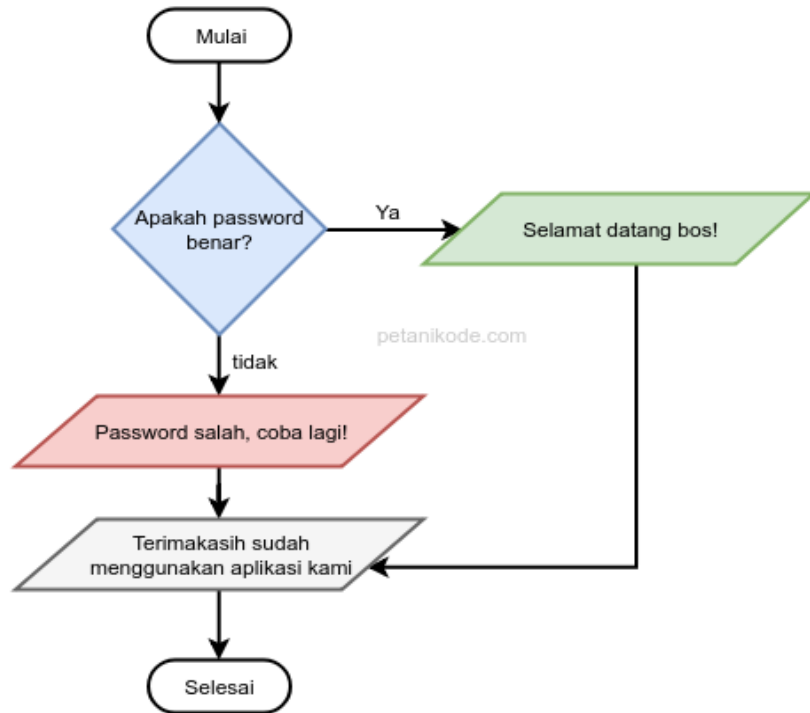
- Percabangan *if* merupakan percabangan yang hanya memiliki **satu blok pilihan** saat kondisi bernilai benar



```
if (condition) {  
    // block of code to be executed if the condition is true  
}
```


Percabangan IF ELSE

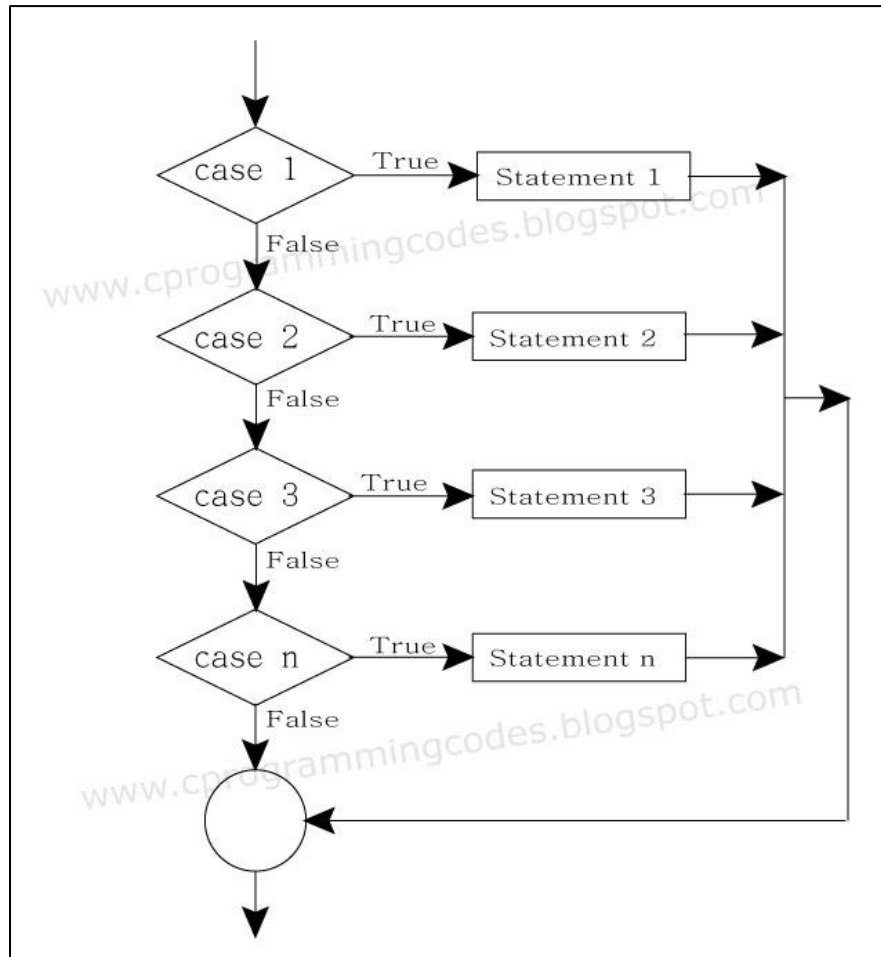
- Percabangan *if/else* merupakan percabangan yang memiliki **dua blok pilihan**. Pilihan pertama untuk kondisi **benar**, dan pilihan kedua untuk kondisi **salah** (*else*)



```
if (condition) {  
    // block of code to be executed if the condition is true  
} else {  
    // block of code to be executed if the condition is false  
}
```

Percabangan SWITCH CASE

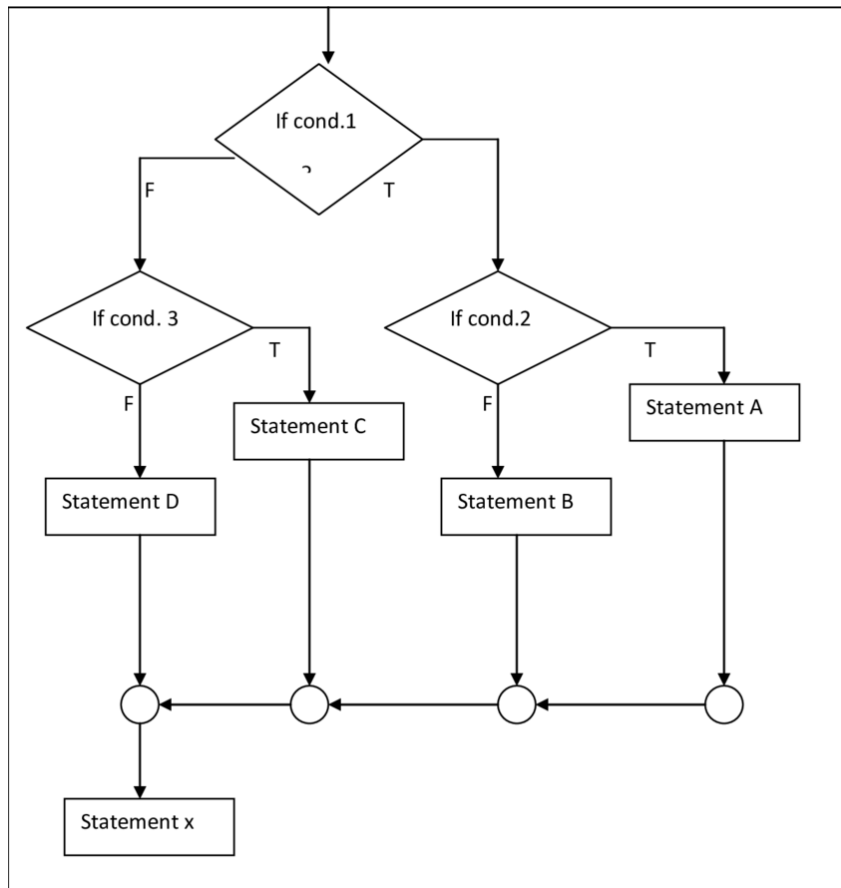
- Switch case digunakan ketika terdapat banyak pilihan



```
switch(expression) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block  
}
```

Percabangan NESTED IF

- Kita juga dapat membuat blok percabangan di dalam percabangan. Ini disebut percabangan bersarang atau *nested if*.



```
if ( test condition 1)
{
    //If the test condition 1 is TRUE then these it will check for test condition 2
    if ( test condition 2)
    {
        //If the test condition 2 is TRUE then these statements will be executed
        Test condition 2 True statements;
    }

    else
    {
        //If the c test condition 2 is FALSE then these statements will be executed
        Test condition 2 False statements;
    }

    else
    {
        //If the test condition 1 is FALSE then these statements will be executed
        Test condition 1 False statements;
    }
}
```

Perulangan

- Perulangan akan membantu kita mengeksekusi kode yang berulang-ulang, berapapun yang kita mau. Ada lima macam bentuk perulangan di Javascript. Secara umum, perulangan ini dibagi dua yaitu *counted loop* dan *uncounted loop*.
- Perbedaanya:
 - **Counted Loop** merupakan perulangan yang jelas dan sudah tentu banyak perulangannya.
 - **Uncounted Loop**, merupakan perulangan yang tidak jelas berapa kali ia harus mengulang.
- Perulangan yang termasuk dalam *Counted Loop*:
 - Perulangan For
 - Perulangan Foreach
 - Perulangan Repeat
- Perulangan yang termasuk dalam *Uncounted Loop*:
 - Perulangan While
 - Perulangan Do/While

Perulangan

Perulangan

For (counted loop)	VS	While (uncounted loop)
		
Push up 10x		Push up Sampai bosan
<pre>for(i=0; i<10; i++){ pushUp(); }</pre>		<pre>while(not bosan){ pushUp(); }</pre>

Petanikode.com

   @petanikode

Perulangan FOR

- Perulangan for merupakan perulangan yang termasuk dalam counted loop, karena sudah jelas berapa kali ia akan mengulang

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

Statement 1 dijalankan sekali sebelum eksekusi blok kode.

Statement 2 mendefinisikan kondisi dijalkannya kode blok.

Statement 3 dieksekusi setiap selesai kode blok dijalankan

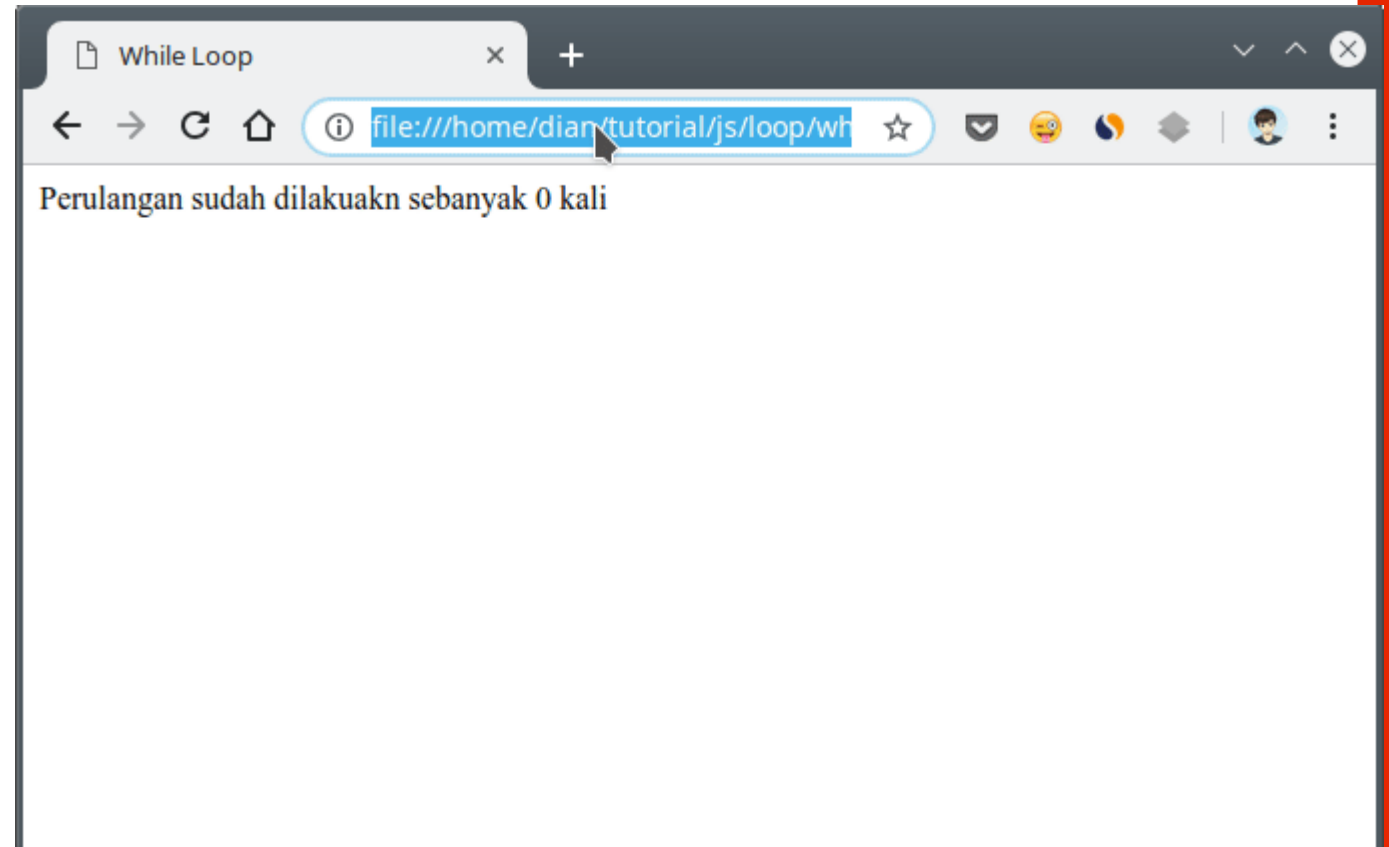
```
for (i = 0; i < 5; i++) {  
    text += "The number is " + i + "<br>";  
}
```

Perulangan WHILE – DO/WHILE

- Perulangan while merupakan perulangan yang termasuk dalam perulangan uncounted loop. Perulangan while juga dapat menjadi perulangan yang counted loop dengan memberikan counter di dalamnya.

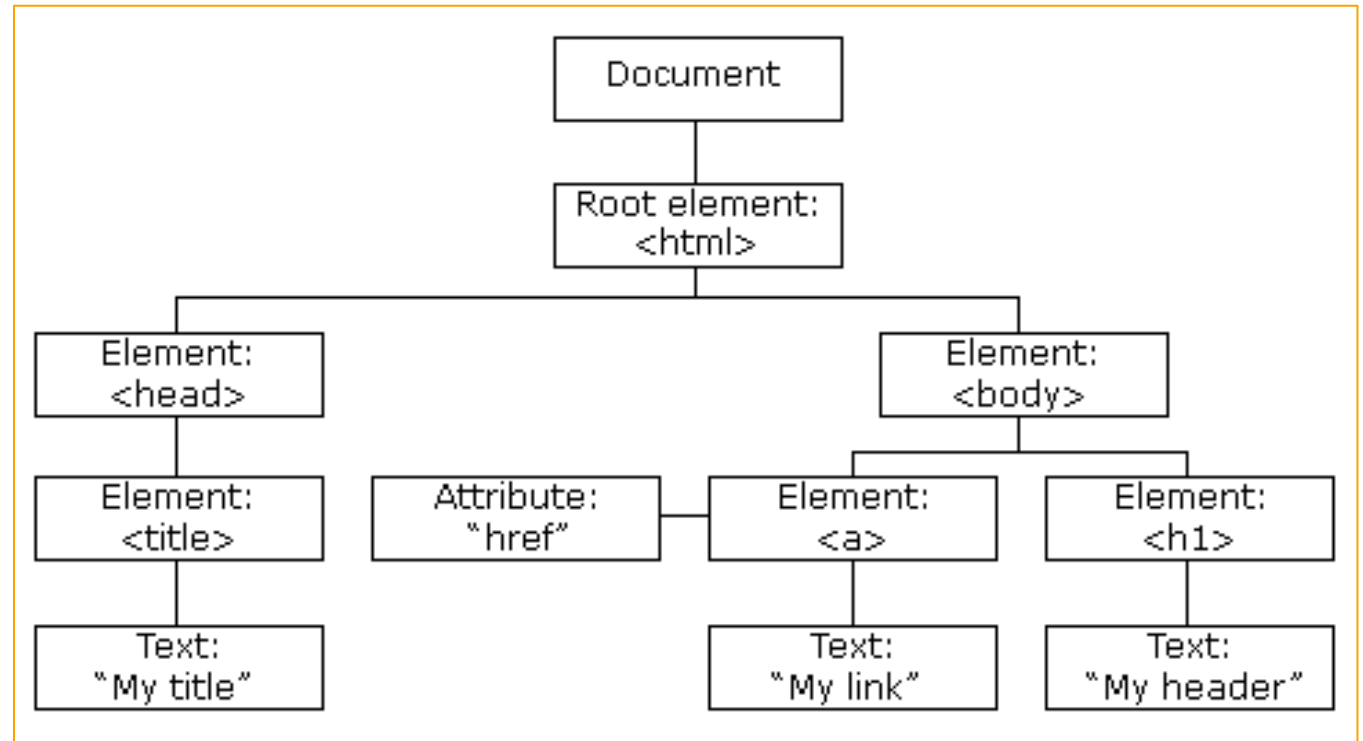
```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

```
do {  
    text += "The number is " + i;  
    i++;  
}  
while (i < 10);
```



Pengenalan DOM

- DOM atau Document Object Model, adalah struktur data yang digunakan oleh browser untuk mewakili dokumen HTML atau XML yang sedang diakses.
- DOM adalah representasi dari halaman web dalam bentuk struktur pohon yang terdiri dari node.
- Setiap bagian dari halaman web, seperti elemen HTML, atribut, dan teks, diwakili sebagai node di dalam pohon ini.
- DOM memungkinkan JavaScript untuk berinteraksi dengan dan memanipulasi halaman web secara dinamis.
- Pengembang dapat mengakses dan mengubah konten, struktur, dan gaya halaman web secara dinamis tanpa perlu memuat ulang halaman.



Struktur DOM

```
<!DOCTYPE html>
<html>
  <head>
    <title>Halaman Contoh</title>
  </head>
  <body>
    <h1>Selamat Datang</h1>
    <p>Ini adalah paragraf contoh.</p>
  </body>
</html>
```



```
Document
├── html
│   ├── head
│   │   └── title
│   │       └── "Halaman Contoh"
│   └── body
│       ├── h1
│       │   └── "Selamat Datang"
│       └── p
│           └── "Ini adalah paragraf contoh."
```

Interaksi dengan DOM menggunakan JavaScript

Dengan DOM objek, JavaScript mendapatkan semua yang dibutuhkan untuk membuat HTML dinamis:

- JavaScript dapat mengubah semua elemen HTML
- JavaScript dapat mengubah semua atribut HTML
- JavaScript dapat mengubah semua styles CSS
- JavaScript dapat menghapus elemen dan atribut HTML yang ada
- JavaScript dapat menambahkan elemen dan atribut HTML baru
- JavaScript dapat bereaksi terhadap semua peristiwa HTML yang ada di halaman
- JavaScript dapat membuat events baru di HTML

Mengakses Elemen

Method	Keterangan
<code>document.getElementById(id)</code>	Menemukan elemen demi elemen id
<code>document.getElementsByTagName(name)</code>	Menemukan elemen berdasarkan nama tag
<code>document.getElementsByClassName(name)</code>	Menemukan elemen berdasarkan nama kelas

Contoh Penggunaan DOM

Mengubah konten

```
let header = document.getElementById("header");  
header.innerHTML = "Judul Baru";
```

Mengubah gaya

```
let header = document.getElementById("header");  
header.style.color = "blue";
```

Menambahkan atau menghapus elemen

```
let newParagraph =  
document.createElement("p"); newParagraph.textContent =  
"Paragraf baru"; document.body.appendChild(newParagraph);
```

Contoh Penggunaan DOM

Manipulasi DOM dengan event

```
<button id="myButton">Klik Saya</button>
<p id="myParagraph">Ini adalah paragraf.</p>

<script>
  let button = document.getElementById("myButton");
  let paragraph = document.getElementById("myParagraph");

  button.addEventListener("click", function() {
    paragraph.textContent = "Tombol telah diklik!";
  });
</script>
```

Contoh Event

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

The list is much longer: [W3Schools JavaScript Reference HTML DOM Events](#).



Terima Kasih