



Universitas Kristen Duta Wacana

Fakultas Teknologi Informasi

Prodi Informatika

Jln. Dr. Wahidin Sudiro Husodo No 5-25

Daerah Istimewa Yogyakarta 55224

Mata Kuliah	: Pemrograman Website
Kode Mata Kuliah	: TI0243
Periode	: Semester Genap 2020/2021
Dosen Pengampu	: Danny Sebastian S.Kom., M.M., M.T. : Maria Nila Anggia Rini S.T., M.Tl. : Agata Filiana S.Kom., M.Cs.

BAB 10 JAVASCRIPT (1)

A. CPL Pertemuan

- Menggunakan javascript untuk membuat interaktivitas halaman website

B. Alat & Bahan

Kegiatan praktikum pada bagian ini membutuhkan perangkat keras (*hardware*) atau komputer dengan spesifikasi minimum:

- Processor : Intel Core i3
- Memory : 2 GB
- Hard disk : kapasitas kosong penyimpanan sebesar 3 GB

Selain perangkat keras, kegiatan praktikum bagian ini juga membutuhkan perangkat lunak (*software*):

- Sistem Operasi (Windows, Linux, atau MacOSX)
- Text editor (Atom, Visual Studio Code, atau editor lain)
- Browser (Google Chrome, Opera, atau Mozilla Firefox)
- Xampp (Web Server Apache & Database MariaDB)

C. Javascript Dasar

JavaScript adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. Client Side Programming Language adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada *web browser* seperti Google Chrome dan Mozilla Firefox. Untuk menjalankan JavaScript, kita hanya membutuhkan aplikasi *text editor*, dan *web browser*.

JavaScript memiliki fitur *loosely typed* yaitu tidak perlu mendefinisikan tipe dari variabel, misal: int, string, dll; semua variabel didefinisikan menggunakan 'var') dan berorientasi objek. JavaScript bersifat

case sensitive dan *sequential* yang artinya kode akan dibaca dari atas ke bawah. Format file dari JavaScript adalah .js

C.1. Menggunakan Javascript

Ada 3 cara menggunakan Javascript

C.1.1. Secara langsung pada tag <head>

Kode JavaScript yang secara langsung diletakkan pada tag <head> diawali dengan tag <script> dengan `type="text/javascript"`. Berikut ini adalah contohnya:

```
<html>
<head>
  <title>Test JavaScript</title>
  <script type="text/javascript">
    document.write("Hello world!");
  </script>
</head>
<body>

</body>
</html>
```

C.1.2. File terpisah yang berekstensi .js

Kode JavaScript juga bisa diletakkan di file tersendiri yang berekstensi .js. Untuk memanggil kode JavaScript yang terdapat di file sendiri, di bagian awal <head> harus ditentukan dahulu nama file .js yang dimaksud menggunakan contoh kode seperti berikut (misal: `src="coba.js"`):

```
<html>
<head>
  <title>Test JavaScript</title>
  <script type="text/javascript" src="coba.js">
  </script>
</head>
<body>

</body>
</html>
```

Berikut ini adalah contoh isi dari "coba.js" yang ada di file tersendiri:

```
document.write("Hello world!");
```

C.1.3. Secara langsung pada elemen

Script ini dieksekusi ketika halaman di-load sampai di bagian <body>. Ketika menempatkan script pada tag <body> berarti antara isi dan JavaScript dijadikan satu bagian.

```
<html>
<head>
  <title>Test JavaScript</title>
</head>
<body>
  <script type="text/javascript">
    document.write("Hello world!");
  </script>
</body>
</html>
```

C.2. Javascript Syntax

C.2.1. Comment

Ada 2 cara menuliskan comment pada javascript, yaitu menggunakan syntax // untuk comment 1 baris, dan /* */ untuk komentar multiple baris.

```
//ini fungsi
function myFunction(){
}

function myFunction(){
  /*baris kode ini akan
  digunakan untuk mengolah
  data anda */
}
```

C.2.2. Variable

Di dalam JavaScript terdapat 4 cara untuk mendefinisikan sebuah variabel yaitu menggunakan keyword var, let, dan const. Berikut perbedaan dari masing masing keyword tersebut:

1. Var

Keyword Var ini digunakan diseluruh JavaScript dari tahun 1995 sampai 2015. Oleh karena itu jika kamu menjalankan kode JavaScript pada browser lama, kamu harus menggunakan var. Ketika kamu membuat variabel di dalam sebuah blok { } menggunakan keyword var, kamu tetap akan bisa mengaksesnya dari luar blok { } tersebut. Hal tersebut disebabkan karena var mempunyai global scope.

```
var x = 5;
var y = "lima";
```

2. Let

Let mulai ditambahkan di JavaScript pada tahun 2015. Salah satu hal yang membedakan let dengan var adalah variabel yang didefinisikan menggunakan let tidak bisa didefinisikan ulang.

```
let a = true;
let b = 0.5;
```

Variabel yang dibuat menggunakan let tidak dapat didefinisikan ulang (Redeclared)

```
let username = "bejo";
let username = "asep";
```

Variabel yang dibuat menggunakan var dapat didefinisikan ulang

```
var x = 0.55;
var x = 1.33;
```

3. Const

Const digunakan untuk membuat sebuah variabel yang tidak akan diubah nilainya (konstanta). Biasanya const digunakan untuk mendefinisikan sebuah nilai tetapan seperti phi, bilangan Avogadro, kecepatan cahaya, dll.

```
const phi = 3.14;
const g = 9.888;
```

	Scope	Redeclared	Reassign
var	Global	ya	ya
let	Block	tidak	ya
const	Global	ya	tidak

Penamaan Variable pada javascript memiliki beberapa aturan:

- Nama variabel harus diawali dengan sebuah huruf
- Nama variable case sensitive
- Keyword tertentu tidak dapat digunakan untuk membuat variable. Contohnya return, break, case, catch, dll.

```
//error
let return = 4;
var catch = "tangkap";
const break = false;
```

- Nama variable dapat juga diawali dengan \$ dan _

```
//error
const _phi = 3.14;
var $string = "inisetring";
let _angka = 1000;
```

- Nama variable dapat memuat karakter alphanumeric

```
let satu1 = 11;
var dua25lima = true;
const avo99dro = "6.02214 x 10^23";
```

C.2.3. Tipe data Javascript

- String
String digunakan untuk menyimpan data teks, contoh penggunaan:
Single quotes: 'Hello'
Double quotes: "Hello"
Backticks: `Hello`

```
const nama = 'vito';
const nama2 = "farel";
const result = `The names are ${nama} and ${nama2}`;
```

Untuk *single* dan juga *double quotes* secara praktek penggunaannya sama, tetapi untuk *backticks* secara umum digunakan ketika kita ingin memasukkan sebuah variabel atau *expression* ke dalam sebuah *string*. Hal tersebut digunakan dengan cara menggunakan notasi `${}` seperti pada contoh baris kode diatas.

- Number
Number digunakan untuk merepresentasikan bilangan *integer* dan juga *floating number* (desimal dan juga eksponensial), contohnya:

```
const angka1 = 3;
const angka2 = 3.433;
const angka3 = 3e5 // 3 * 10^5
```

- BigInt

Di dalam JavaScript tipe data Number hanya bisa merepresentasikan data kurang dari $(2^{53}-1)$ dan lebih dari $-(2^{53}-1)$, maka dari itu JavaScript menyediakan BigInt jika ingin menggunakan angka yang lebih besar dari itu, contoh penggunaannya:

```
// BigInt value
const value1 = 900719925124740998n;

// menambhakna dua big integers
const result1 = value1 + 1n;
console.log(result1); // "900719925124740999n"

const value2 = 900719925124740998n;

// Error! BitInt dan number tidak bisa ditambahkan
const result2 = value2 + 1;
console.log(result2);
```

```
900719925124740999n
Uncaught TypeError: Cannot mix BigInt and other types
```

output:

- Boolean

Tipe data boolean merepresentasikan entitas logikal yaitu true dan false, contohnya:

```
const data = true;
const salah = false;
```

- Undefined

Tipe data undefined merepresentasikan data yang tidak terdapat *value* atau isinya. Jika sebuah variabel dideklarasikan tetapi variabel tersebut tidak diisi dengan suatu nilai apapun, maka variabel tersebut bertipe data undefined, contohnya:

```
let names;
console.log(names); // undefined
```

Selain itu dapat dilakukan secara eksplisit:

```
let name2 = undefined;
console.log(name2); // undefined
```

- Null

Pada JavaScript nilai null merepresentasikan nilai kosong atau *unknown value*, contohnya:

```
const tidakTahu = null;
```

- Symbol

Tipe data symbol dapat disebut sebagai *symbol value*. Tipe data symbol merupakan nilai *immutable primitive* yang unik, contohnya:

```
const valuee1 = Symbol('hello');
const valuee2 = Symbol('hello');
```

Meskipun valuee1 dan valuee2 keduanya berisi 'halo', keduanya berbeda karena bertipe Simbol.

- Array

Sama seperti bahasa pemrograman lain, javascript juga memiliki tipe data array.

```
var cars = [
  "Saab",
  "Volvo",
  "BMW"
];

var cars = [];
cars[0] = "Saab";
cars[1] = "Volvo";
cars[2] = "BMW";

var buah = ["Banana", "Orange", "Apple", "Mango"];
var jumlah = fruits.length;

<p id="demo"></p>
<script>
var buah = ["Banana", "Orange", "Apple", "Mango"];
var jumlah = fruits.length;

let text = "<ul>";
for (let i = 0; i < fLen; i++) {
  text += "<li>" + fruits[i] + "</li>";
}
text += "</ul>";
document.getElementById("demo").innerHTML = text;
</script>
```

- Object

Tipe data object merupakan tipe data yang berisi pasangan *key:value*, jika pada python maka hampir mirip dengan dictionary, dan jika pada PHP mirip dengan associative array, contoh penggunaannya:

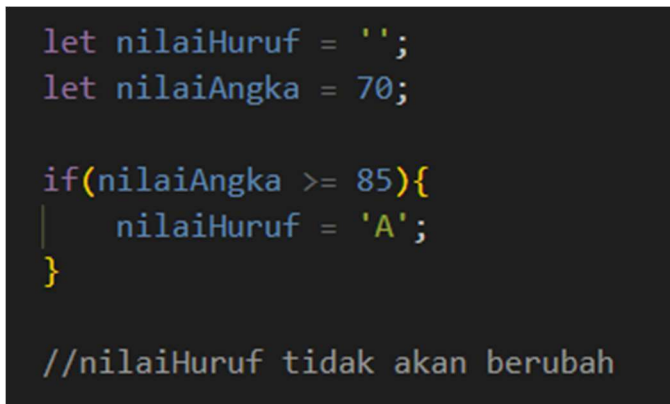
```
const student = {
  firstName: 'farel',
  lastName: null,
  class: 10
};
```

C.2.4. Percabangan

Percabangan pada JavaScript secara umum ada 4 bentuk, yaitu: if, else, else if, switch. Namun yang akan dibahas pada modul kali ini hanya if, else dan else if. Berikut contoh bentuk conditional if:

```
if(<kondisi>){  
    <lakukan ini>  
    <lakukan ini>  
    ...  
}
```

Sebagai contoh, apabila mahasiswa mendapat nilai ≥ 85 maka nilai hurufnya akan bernilai A. Implementasinya sebagai berikut:



```
let nilaiHuruf = '';  
let nilaiAngka = 70;  
  
if(nilaiAngka >= 85){  
    nilaiHuruf = 'A';  
}  
  
//nilaiHuruf tidak akan berubah
```

Bentuk conditional else merupakan percabangan yang memiliki dua alternatif langkah yang harus dijalankan berdasarkan kondisi tertentu. Secara umum bentuknya adalah sebagai berikut:

```
if(<kondisi>){  
    <lakukan ini>  
    <lakukan ini>  
    ...  
} else {  
    <lakukan itu>  
    <lakukan itu>  
    ...  
}
```

Sebagai contoh, apabila suhu ruangan < 26 maka akan dianggap dingin, sedangkan selain itu dianggap panas. Implementasinya sebagai berikut:


```

let kategoriSuhu = "unidentified";
let suhuKamar = 30;

if(suhuKamar < 26){
  kategoriSuhu = "dingin";
} else {
  kategoriSuhu = "puanas";
}

//kategoriSuhu akan bernilai "puanas"

```

Bentuk conditional else if digunakan jika kemungkinan langkah yang harus dijalankan berikutnya lebih dari dua. Bentuknya secara umum adalah sebagai berikut:

```

if(<kondisi>){
  <lakukan ini>
  <lakukan ini>
  ...
} else if {
  <lakukan itu>
  <lakukan itu>
  ...
} else if {
  <lakukan sini>
  <lakukan sini>
  ...
}
...
else {
  <lakukan situ>
  <lakukan situ>
  ...
}

```

Sebagai contoh, tulisan dengan ukuran font yang bernilai kurang sama dengan 11 akan dianggap kecil, bernilai 12-14 dianggap normal, dan lebih dari 14 dianggap besar. Perhatikan implementasi berikut:

```

let ukuranFont = 12;
let kategori = "unknown";

if(ukuranFont <= 11) {
    kategori = "kecil";
} else if(ukuranFont >= 12 && ukuranFont <= 14) {
    kategori = "normal";
} else {
    kategori = "besar";
}
//kategori akan bernilai "normal"

```

C.2.5. Perulangan

Perulangan for pada JavaScript memiliki sintaks:

```

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

```

Pada statement pertama akan dieksekusi satu kali saja sebelum eksekusi pada baris kode tersebut. Pada statement kedua mendefinisikan kondisi untuk mengeksekusi baris kode, dan pada statement ketiga akan dieksekusi selalu setelah baris kode telah dieksekusi. Contoh penggunaannya:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <p id="demo"></p>

  <script>
    let text = "";
    for (let i = 0; i < 5; i++) {
      text += "The number is " + i + "<br>";
    }
    document.getElementById("demo").innerHTML = text;
  </script>
</body>
</html>

```

Hasilnya adalah

```

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4

```

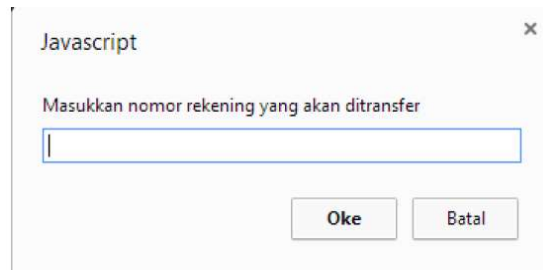
Pada contoh diatas perulangan tersebut menggunakan tiga statement. Pada statement pertama, perulangan tersebut mendefinisikan suatu variabel sebelum perulangan itu dimulai, variabel tersebut adalah `i = 0`. Lalu pada statement kedua, perulangan tersebut mendefinisikan suatu kondisi ketika perulangan tersebut berjalan yaitu `i < 5`, yang berarti agar perulangan itu berjalan maka syaratnya adalah isi dari variabel `i` kurang dari 5. Lalu pada statement ketiga, perulangan tersebut akan menambahkan nilai dari `i` sebanyak 1 (`i++`) setiap kali baris perulangan tersebut telah dieksekusi.

C.3. Javascript Dialogue Box

C.3.1. Prompt

Perintah prompt digunakan untuk memungkinkan pengguna memasukkan sesuatu dan melakukan sesuatu terhadap info yang dimasukkan tersebut.

Syntax: `prompt("<text yang ingin dituliskan>");`



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

  <button onclick="myFunction()">Try it</button>

  <p id="demo"></p>

  <script>
    function myFunction() {
      let person = prompt("Masukkan Nama", "Vito Kurniadi");
      if (person != null) {
        document.getElementById("demo").innerHTML =
          "Halo " + person + "! apakah sudah membuat web hari ini?";
      }
    }
  </script>
</body>
</html>
```

Pada contoh di atas terlihat bahwa terdapat fungsi `myFunction` yang di dalamnya akan memanggil fungsi `prompt`. Fungsi `myFunction` akan membuat variabel bernama `person` yang akan diisi dengan fungsi `prompt`, parameter pertama fungsi `prompt` tersebut berisi dengan “Masukkan Nama”, dan parameter keduanya berisi dengan “Vito Kurniadi”. Lalu terdapat sebuah kondisi jika variabel `person` tidak null (pengguna tidak memencet *cancel*) maka akan mengisi elemen HTML yang memiliki `id="demo"` dengan “Halo “ + `person` + “! apakah sudah membuat web hari ini?” dimana variabel `person` tersebut akan berisi nilai inputan yang telah diisi pengguna. Hasil halaman web jika pengguna mengisi dan memencet *OK*:



Halo Vito Kurniadi! apakah sudah membuat web hari ini?

Jika memencet *cancel*:

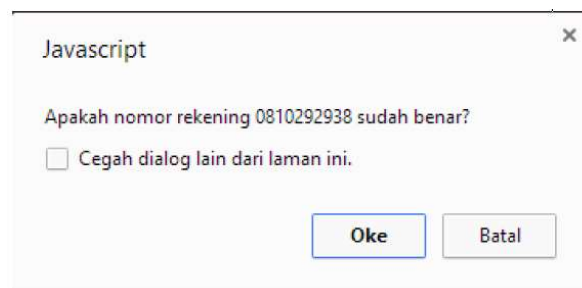


Maka tidak akan muncul apapun.

C.3.2. *Confirm*

Perintah `confirm` digunakan untuk melakukan konfirmasi pilihan pengguna terhadap dua pilihan.

Syntax: `confirm("<text yang ingin dituliskan>");`



Berikut contoh penggunaan `confirm`. Terlihat bahwa kita membuat fungsi bernama `myFunction` yang berisi variabel `text` dan juga memanggil fungsi `confirm` dengan parameternya adalah variabel `text`, pada `myFunction` terdapat percabangan jika fungsi `confirm(text)` mengembalikan *true* (dipencet *OK*) maka

variabel text akan berubah menjadi "Kamu telah memencet OK!", dan jika dipencet cancel maka variabel text akan menjadi "Kamu telah memencet cancel!". Lalu pada fungsi myFunction juga akan mengubah isi dari elemen HTML yang memiliki id="demo" dengan variabel text.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

  <button onclick="myFunction()">Try it</button>

  <p id="demo"></p>

  <script>
    function myFunction() {
      let text = "Pencet tombolnya!\n OK ataupun Cancel.";
      if (confirm(text) == true) {
        text = "Kamu telah memencet OK!";
      } else {
        text = "Kmau telah memencet cancel!";
      }
      document.getElementById("demo").innerHTML = text;
    }
  </script>
</body>
</html>
```

Jika memencet *OK* maka pada halaman web akan menjadi:

Kamu telah memencet OK!

Jika memencet *cancel* maka halaman web akan menjadi:

Try it

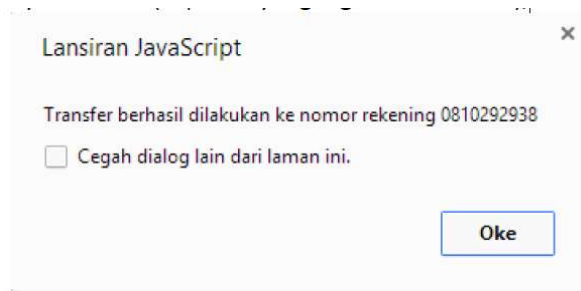
Kmau telah memencet cancel!

Fungsi myFunction akan dipanggil ketika kita memencet elemen button.

C.3.3. Alert

Alert adalah perintah untuk menampilkan pesan. (*Pop up message*). Contoh penggunaan dapat dilihat dari contoh yang sebelumnya.

Syntax: `alert("<pesan yang ingin dituliskan>");`



C.3.4. Contoh Penggunaan Prompt, Confirm, Alert

```
<!DOCTYPE html>
<html>
<head>
  <title>Test JavaScript - Interactive Dialog Boxes</title>
  <script type="text/javascript">
    function transfer(){
      var rek = prompt("Masukkan nomor rekening yang akan ditransfer");
      var cek = confirm("Apakah nomor rekening "+ rek +" sudah benar?");
      if(cek){
        alert("Transfer berhasil dilakukan ke nomor rekening "+rek);
      }
      else{
        alert("Transfer tidak dilakukan");
      }
    }
  </script>
</head>
<body>
  <button onClick="transfer()">Transfer</button>
</body>
</html>
```