

MODUL 4 - Javascript Web Programming



Disusun oleh :

Moh. Andris Saputra

23083000191

4H

**JURUSAN SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS MERDEKA
MALANG**

MODUL 4

Javascript

"

TUJUAN

1. Mahasiswa Memahami penggunaan Javascript
2. Mahasiswa Mampu mengaplikasikan dan membuat Javascript

CARA Pengerjaan

1. Setiap code diisikan dengan Screenshot
2. Output diisikan dengan Screenshot
3. Penjelasan dituliskan dalam text

OVERVIEW

Javascript merupakan bagian dari 3 teknologi penting yang harus dikuasai programmer web, yakni HTML untuk konten (isi dari website), CSS untuk tampilan (presentation), dan JavaScript untuk interaksi (behavior). Hampir seluruh website modern saat ini menggunakan JavaScript untuk membuat berbagai aplikasi yang dapat berinteraksi dengan user, seperti validasi form HTML, games, kalkulator, fitur chatting, dll.

Dalam mempelajari JavaScript, sebaiknya telah menguasai dasar-dasar HTML, dan bisa membuat halaman web sederhana menggunakan HTML. Pengetahuan tentang HTML, CSS dan PHP juga akan membantu.

MODUL 4

Javascript

"

DEFINISI

Javascript adalah bahasa pemrograman yang bisa membuat internet bekerja. Lebih lengkapnya, Javascript adalah bahasa pemrograman yang ringan, bersifat dinamis, dan dapat diarahkan untuk oriented programming maupun functional programming. Javascript biasanya digunakan untuk melakukan perubahan tampilan website, seperti ketika mengklik tombol, menentukan layout dan memberikan efek dinamik.

Javascript dikenal sebagai bahasa pemrograman yang cukup kompleks, namun juga sangat fleksibel. Banyak programmer yang sudah menyediakan tools yang berdiri di atas core Javascript yang bisa digunakan untuk belajar bahasa pemrograman.

Javascript Juga dapat dipakai sebagai core sebagai bahasa pemrograman, diantaranya untuk :

1. Application Programming Interfaces (APIs) dibangun pada web browser agar kamu bisa melakukan apapun dari dinamik dokumen HTML dan set CSS yang sudah kamu buat.
2. API pihak ketiga menyediakan akses bagi programmer untuk menghubungkan aplikasi mereka pada website atau aplikasi lain, contohnya seperti Facebook dan Twitter. Kamu pernah nggak login Soundcloud menggunakan akun Facebook? Nah, ini salah satu contoh pihak ketiga.
3. Frameworks pihak ketiga dan libraries dapat digabungkan pada HTML sehingga memungkinkan programmer membangun website atau membuat aplikasi dengan cepat.

KEUNTUNGAN

Keuntungan dari bahasa pemrograman javascript adalah :

1. Sederhana
2. Cepat
3. Fleksibel

MODUL 4

Javascript

"

4. Kaya akan Fitur

MODUL 4

Javascript

"

CODE PADA JAVASCRIPT

JavaScript termasuk jenis bahasa script, yang digunakan di dalam file HTML. Untuk menginput, atau memasukkan kode JavaScript ke dalam HTML, JavaScript menyediakan 4 alternatif, yaitu :

4.1 Penggunaan Javascript

4.1.1 Internal JavaScript <script>

Untuk menggunakan internal JavaScript yang digunakan adalah tag <script>. Cara pertama untuk menginput kode JavaScript ke dalam halaman HTML adalah dengan menggunakan tag <script> secara internal. Internal disini berarti bahwa kode JavaScript ditulis pada halaman yang sama dengan HTML, atau di dalam satu file HTML.

Cara ini merupakan cara yang paling sering digunakan, jika kode JavaScript tidak begitu panjang, dan hanya digunakan di 1 halaman saja. Kode JavaScript yang akan di input diletakkan diantara tag pembuka <script> dan tag penutup </script>

Contoh Code:

MODUL 4

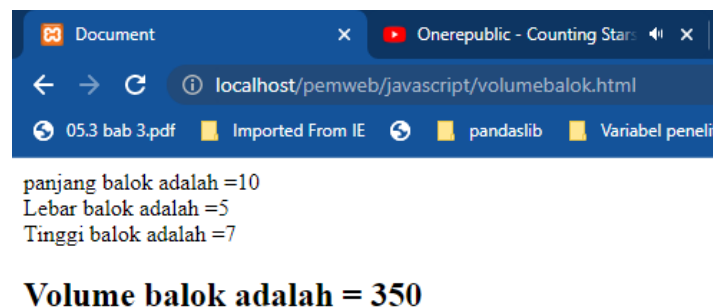
Javascript

"



Gambar 4. 1. Internal JavaScript

Output :



Gambar 4. 2. Output Internal JavaScript

Buatlah contoh internal JavaScript :

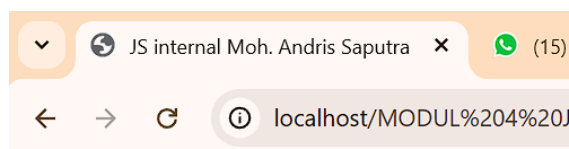
MODUL 4

Javascript

Code :

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>JS internal Moh. Andris Saputra</title>
8
9    <script>
10     var P = 10;
11     var L = 5;
12     var T = 7;
13     volume = P * L * T;
14     document.write('panjang balok adalah = ' + P + '<br>');
15     document.write('lebar balok adalah = ' + L + '<br>');
16     document.write('tinggi balok adalah = ' + T + '<br>');
17
18     window.onload = function () {
19       document.querySelector('#volume_result').innerHTML = volume + '<br>';
20     }
21   </script>
22 </head>
23
24 <body>
25   <h2>Volume balok adalah = <span id="volume_result"></span></h2>
26 </body>
27
28 </html>
```

Output :



panjang balok adalah = 10
lebar balok adalah = 5
tinggi balok adalah = 7

Volume balok adalah = 350

4.1.2 External Javascript<script scr="">

Cara atau metode kedua untuk menginput kode JavaScript ke dalam halaman

MODUL 4

Javascript

HTML adalah dengan memindahkan kode JavaScript ke dalam sebuah file terpisah, lalu “memanggilnya” dari HTML. Cara ini sangat disarankan karena akan memberikan banyak keuntungan dan fleksibilitas dalam membuat program JavaScript.

Sebuah file JavaScript disimpan dalam ekstensi .js, seperti:

1. sample.js
2. login.js,
3. logout.js.

Dari halaman HTML, kita memanggilnya menggunakan tag <script> dengan atribut src. Atribut src berisi letak dari file .js yang anda buat.

Buatlah contoh external JavaScript :

Code :

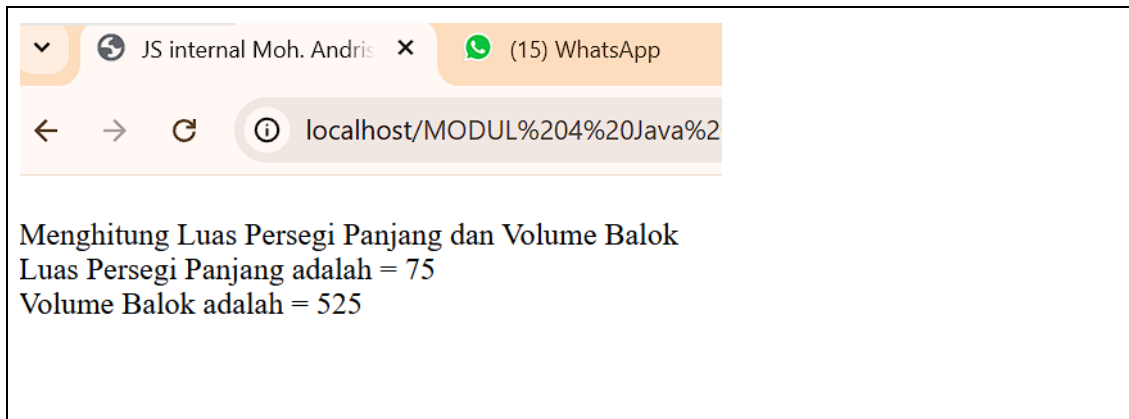
```
MODUL 4 Java script > external-js.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>JS internal Moh. Andris Saputra</title>
8  </head>
9
10 <body>
11   <h1 id="screen_size"></h1>
12   <script src="external-js.js"></script>
13 </body>
14
15 </html>

MODUL 4 Java script > JS external-js.js > ...
1  document.write("Menghitung Luas Persegi Panjang dan Volume Balok <br>")
2  var P = 15;
3  var L = 5;
4  var T = 7;
5  Luas = P*L;
6  Volume = P*L*T;
7  document.write("Luas Persegi Panjang adalah = "+ Luas + "<br>")
8  document.write("Volume Balok adalah = "+ Volume + "<br>")
```

Output :

MODUL 4

Javascript



4.1.3 Inline JavaScript (Event Handler)

Cara ketiga untuk menjalankan JavaScript adalah dengan memanggilnya menggunakan Event Handler dari dalam tag HTML. secara sederhananya, event handler adalah pemanggilan kode javascript ketika „sesuatu“ terjadi dalam tag HTML.

Sesuatu disini maksudnya ketika sebuah element dalam HTML di klik, di klik kanan, diarahkan mouse, dan lain-lain. Event handler di dalam JavaScript ditulis dengan penambahan kata on. Sehingga jika sebuah tombol di-klik, maka disebut sebagai onclick, jika mouse berada diatas element disebut sebagai onmouseover, dan lain-lain.

Event yang disediakan JavaScript terbagi menjadi 2 (dua) jenis, yaitu event sistem dan event pengguna. Event sistem adalah event yang selalu ada dan disediakan oleh JavaScript serta sangat terkait dengan sistem dimana halaman web ditampilkan. Sedangkan event pengguna adalah event yang hanya tersedia relatif terhadap konteks elemen halaman web apa yang diacu oleh pengguna.

Berikut dibawah ini adalah event system :

Tabel 4. 1. Event System dalam JavaScript

Event	Keterangan
onError	Event ini dibangkitkan ketika terjadi kesalahan. Event ini dimiliki oleh objek Window dan Image

MODUL 4

Javascript

onLoad	Event ini dibangkitkan ketika suatu objek selesai ditampilkan pada halaman web. Objek yang dimaksud pada keterangan ini adalah objek Window, Frame dan Image
onUnload	Event ini dibangkitkan ketika suatu objek telah dikeluarkan dari ruang memori, atau telah selesai digunakan. Objek ini adalah objek Window, Frame dan Image

Tabel 4. 2. List event pengguna beserta keterangannya

Event	Keterangan
onAbort	Event yang dimiliki oleh objek image ini akan dibangkitkan ketika. Sebuah gambar dihentikan proses pemunculannya yang dimungkinkan akibat penekanan tombol stop pada browser
onBlur	Event ini dibangkitkan ketika sebuah element (dalam hal ini adalah element window, frame, select, text, dan textarea) kehilangan fokusnya
onChange	Event ini dibangkitkan ketika sebuah element (dalam hal ini adalah element select, text dan textarea) telah diubah nilainya sebelum element tersebut kehilangan fokusnya
onClick	Event ini dibangkitkan ketika terjadi aksi klik (dilakukan oleh pengguna) terhadap element, dimana element yang dimaksud adalah semua elemen dari form yang dapat di klik seperti element button (tombol). Aksi klik yang dimaksud adalah adanya penekanan tombol klik kiri mouse terhadap suatu element
onContextMenu	Event ini terbangkitkan ketika terjadi aksi klik kanan dari tombol mouse terhadap suatu element
onDbClick	Hampir sama dengan event onClick, namun akan dibangkitkan ketika terjadi aksi klik ganda terhadap suatu element
onFocus	Event ini adalah kebalikan dari event onBlur, dimana akan dibangkitkan ketika suatu element dikenai focus. Element-element yang dimaksud adalah element window, frame, select, text dan textarea
onHelp	Event ini dibangkitkan ketika terjadi penekanan terhadap tombol F1, yang umumnya digunakan berbagai aplikasi sebagai tombol shortcut untuk menampilkan fasilitas bantuan (help) . Anda dapat menggunakan event ini untuk mengalihkan fitur help milik browser menjadi fitur help yang Anda buat sendiri untuk halaman web

MODUL 4

Javascript

	Anda.
onKeyDown	Event ini dibangkitkan ketika terjadi penekanan ke bawah tombol keyboard
onKeyPress	Event ini dibangkitkan ketika terjadi aksi penekanan tombol keyboard. Aksi penekanan disini adalah sebuah aksi lengkap penekanan tombol keyboard, mulai dari memencet tombol keyboard tersebut hingga melepaskannya
onKeyUp	Event ini dibangkitkan ketika tombol keyboard yang awalnya ditekan lalu dilepas, ketika tombol dilepas event ini bangkit .
onMouseDown	Event ini dibangkitkan, ketika tombol mouse ditekan ke bawah
onMouseMove	Event ini dibangkitkan ketika mouse digerakkan (panah mouse berubah posisinya)
onMouseout	Event ini dibangkitkan ketika panah mouse keluar dari daerah lingkup suatu element, yaitu element hyperlink dan element area.
onMouseover	Event ini dibangkitkan ketika panah mouse berada diatas element , yaitu element hyperlink dan element area .
onReset	Event yang dimiliki secara khusus oleh objek form ini akan dibangkitkan ketika terjadi penekanan terhadap tombol reset yang dimiliki form yang bersangkutan
onResize	Event ini dibangkitkan ketika jendela browser diubah ukurannya, baik diperbesar maupun diperkecil
onSelect	Event ini dibangkitkan ketika terjadi pemilihan teks pada lement text dan textarea dengan cara memberinya highlight (di block)
onStop	Event ini dibangkitkan ketika pengguna melakukan penekanan terhadap tombol stop dari browsernya
onSubmit	Event ini dibangkitkan setelah terjadi penekanan tombol submit yang dimiliki sebuah form

Buatlah contoh Inline JavaScript :

MODUL 4

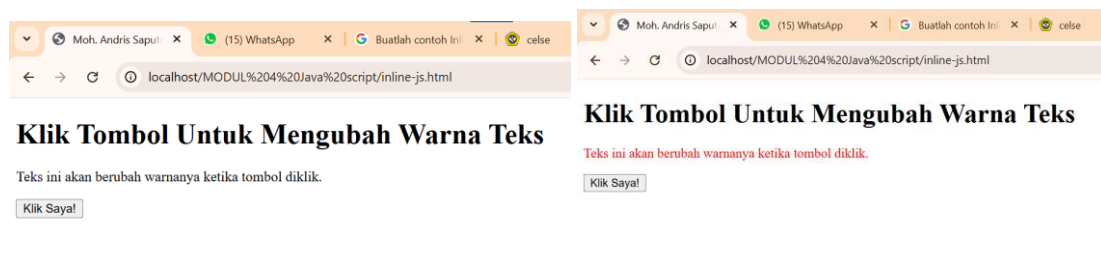
Javascript

Code :

**di halaman berikutnya*

```
MODUL 4 Java script > <> inline-js.html > html > head > script
1  <!DOCTYPE html>
2  <html lang="id">
3
4  <head>
5      <meta charset="UTF-8">
6      <title>Moh. Andris Saputra</title>
7      <script>
8          function UbahWarnaTeks() {
9              var elemenP = document.getElementById('paragraf')
10             elemenP.style.color = elemenP.style.color === 'red' ? 'black' : 'red';
11         }
12     </script>
13 </head>
14 <body>
15     <h1>Klik Tombol Untuk Mengubah Warna Teks</h1>
16     <p id="paragraf">Teks ini akan berubah warnanya ketika tombol diklik.</p>
17     <button onclick="UbahWarnaTeks()">Klik Saya!</button>
18 </body>
19 </html>
```

Output :



4.1.4 Menggunakan URL (href="javascript:")

Cara terakhir (dan juga paling jarang digunakan saat ini) adalah dengan menyisipkan JavaScript ke dalam alamat href dari tag HTML. Cara ini disebut juga dengan protocol javascript. Disebut demikian, karena kita mengganti alamat link dari yang biasa menggunakan protokol http//: menjadi javascript.

Dari ke-4 cara menginput kode JavaScript, memisahkan kode JavaScript ke dalam sebuah file tersendiri (menggunakan metode `<script src="">`) adalah yang paling disarankan. Beberapa keuntungan menggunakan metoda `<script src>` bila dibandingkan dengan metode cara memasukkan JavaScript lainnya adalah:

1. Menyederhanakan halaman HTML dengan memindahkan seluruh kode

MODUL 4

Javascript

"

JavaScript, sehingga halaman HTML hanya berisi konten saja.

2. Sebuah file JavaScript external bisa digunakan untuk beberapa halaman HTML, sehingga jika diperlukan perubahan, kita hanya perlu mengedit sebuah file daripada mengubah secara satu persatu halaman HTML tempat JavaScript ditulis secara internal.
3. Jika file JavaScript external digunakan oleh beberapa halaman, file tersebut hanya perlu didownload oleh web browser pada saat pertama kali saja. Pada saat loading halaman lainnya, web browser cukup mengambilnya dari browser cache, sehingga mempercepat loading halaman.

Buatlah contoh JavaScript menggunakan URL:

Code :

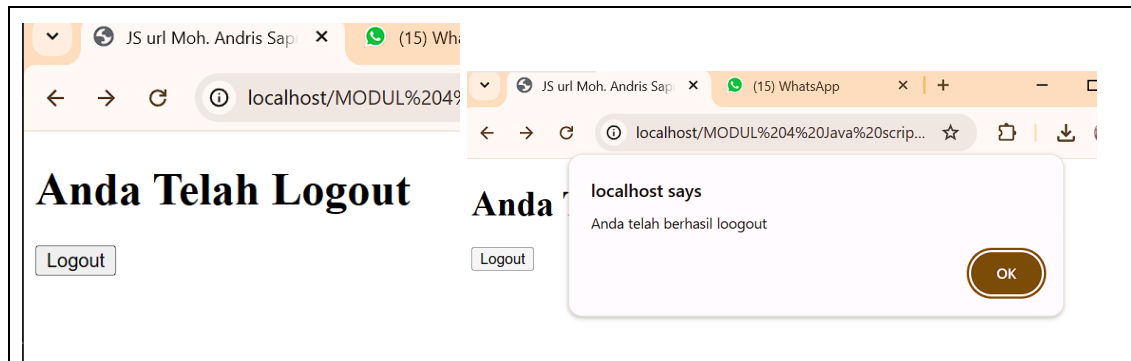
```
MODUL 4 Java script > JS url.js > prosesLogout
1  function prosesLogout() {
2      localStorage.removeItem('userSessionToken');
3      alert('Anda telah berhasil loogout');
4      window.location.href = 'login.html';
5  }

MODUL 4 Java script > url.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>JS url Moh. Andris Saputra</title>
8      <script src="url.js"></script>
9  </head>
10 <body>
11     <h1>Anda Telah Logout</h1>
12     <button onclick="prosesLogout()">Logout</button>
13 </body>
14 </html>
```

Output :

MODUL 4

Javascript



4.2 Tipe Data dan Operasi Dasar JavaScript

Pada pertemuan ini kita akan membahas jenis tipe data yang dapat diproses dalam JavaScript. Variabel yang digunakan untuk menyimpan data atau informasi yang akan diproses, serta operasi dasar JavaScript yang digunakan untuk melakukan pemrosesan data.

4.2.1 Aturan Dasar Penulisan Kode Program JavaScript

1. Perbedaan Penulisan Huruf (Case Sensitivity)

Di dalam **JavaScript**, penulisan huruf besar dan huruf kecil dibedakan, atau dalam istilah pemrograman bersifat Case Sensitif. Hal ini berarti penulisan variabel, keyword, maupun nama fungsi di dalam JavaScript harus konsisten. Variabel **name**, **Name**, **NaMe** dan **NAME** merupakan 4 variabel berbeda. Sedangkan untuk penulisan *keyword* **while**, harus ditulis dengan 'while', bukan 'While' atau 'WHILE'.

Namun karena HTML sendiri tidak bersifat case-sensitif, kadang hal ini bisa mendatangkan permasalahan. Contohnya, jika menggunakan event handler, di dalam HTML kadang bisa ditulis: **onclick**, **onClick**, atau **OnClick**. Ketiga penulisan ini dibolehkan di dalam HTML. Akan tetapi untuk menghindari permasalahan, sebaiknya anda membuat kesepakatan untuk menggunakan **huruf kecil untuk semua penulisan keyword dan variabel di dalam JavaScript**.

Buatlah contoh Case Sensitivity pada JavaScript :

MODUL 4

Javascript

Code :

```
MODUL 4 Java script > JS casesensitivity.js > ...
1  var kota = 'Jakarta'
2  var kota = 'Surabaya'
3  console.log(kota);
4  console.log(kota);
5
6  function salam () {
7      console.log('Halo dari fungsi salam!');
8  }
9  function salam () {
10     console.log('Halo dari fungsi salam!');
11 }
12 salam();
13 salam();
14
15 var mobil = {
16     merk: 'Toyota',
17     merk: 'Honda'
18 };
19 console.log(mobil.merk);
20 console.log(mobil.merk);
```

2. Cara Penulisan Komentar dalam JavaScript

JavaScript mendukung 2 jenis cara penulisan komentar, yakni menggunakan karakter // untuk komentar dalam 1 baris, dan karakter pembuka komentar /* dan penutup */ untuk komentar yang mencakup beberapa baris.

Buatlah contoh komentar pada JavaScript :

Code :

```
MODUL 4 Java script > JS coment.js
1  // contoh comment
2  //
3  /*contoh komen
4  */
```

MODUL 4

Javascript

4.2.2 Variabel dalam JavaScript

Dalam bahasa pemrograman, variabel adalah “**penampung**” sebuah nilai. Tergantung dengan “**nilai**” dari variabel tersebut, sebuah variabel di dalam JavaScript dapat bertipe **Angka (Number)**, **String**, **Boolean**, atau yang lainnya.

1. Aturan Penamaan Variabel JavaScript

Karakter pertama harus diawali dengan **huruf**, **underscore (_)** atau **tanda dollar (\$)**, Karakter kedua dan seterusnya bisa ditambahkan dengan **huruf**, **angka**, **underscore (_)** atau **tanda dollar (\$)**.

Buatlah contoh penamaan Variabel pada JavaScript :

Code :

```
MODUL 4 Java script > JS penamaan.js > ...
1 let nama;           //diawali huruf
2 let _alamat         //diawal underscore
3 let $telepon;       //diawali dollar sign
4 let namaLengkap;    //camelCase
5 let umur2;          //mengandung angka di belakang
6 let _data$Sementara3; //kombinasi underscore, dollar, dan angka
```

2. Cara Membuat Variabel JavaScript

Cara membuat variabel di dalam JavaScript dibedakan menjadi 2, yakni dengan menggunakan keyword **var**, dan **tanpa var**.

contoh penulisan menggunakan kata kunci var :

MODUL 4

Javascript

"



```
<script>
  var a;
  var b;
  var c;
  var d,e,f;
  var g; var h;
  var i = 10;
</script>
```

Gambar 4. 3. Menggunakan var

contoh penulisan menggunakan tanpa kata kunci var :

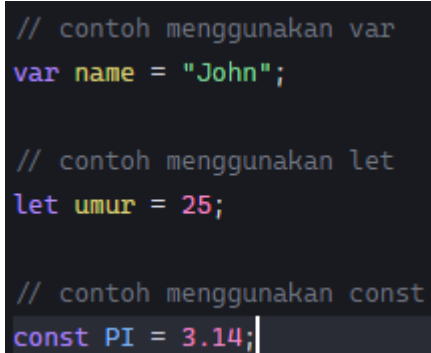


```
<script>
  a;
  b;
  c;
  d,e,f;
  g; h;
  i = 10;
</script>
```

Gambar 4. 4. Tidak menggunakan var

Buatlah contoh penamaan Variabel pada JavaScript :

Code :



```
// contoh menggunakan var
var name = "John";

// contoh menggunakan let
let umur = 25;

// contoh menggunakan const
const PI = 3.14;
```

Pembuatan variabel tanpa menggunakan keyword var memang lebih cepat, akan

MODUL 4

Javascript

tetapi tidak disarankan.

4.2.3 Operator dalam JavaScript

Operator pada JavaScript terbagi menjadi enam, yaitu :

1. Aritmatika
2. Pemberian nilai (Assign)
3. Pemanipulasian bit (bitwise)
4. Pembandingan
5. Logika
6. String

Jelaskan dan praktikan dengan mengcoding ke enam operator di atas, dengan meng-screenshot code anda serta hasil output pada tampilan browser. Letakan screenshot anda pada tabel berikut :

Buatlah Operator dalam JavaScript :

1. Aritmatika

Penjelasan : Digunakan untuk operasi matematika dasar: **+**, **-**, *****, **/**, **%**, ******.

Code :

MODUL 4

Javascript

```
MODUL 4 Java script > JS aritmatika.js > ...
```

```
1 let a = 10, b = 4;  
2 console.log("Penjumlahan:", a + b); //14  
3 console.log("Pengurangan:", a - b); //6  
4 console.log("Perkalian:", a * b); //40  
5 console.log("Pembagian:", a / b); //2.5  
6 console.log("Pangkat:", a ** b); //10000
```

Output :

```
C:\Program Files\nodejs\node.exe .\MODUL 4 Java script\aritmatika.js  
Penjumlahan: 14  
Pengurangan: 6  
Perkalian: 40  
Pembagian: 2.5  
Pangkat: 10000
```

2. Assign

Penjelasan : Digunakan untuk memberikan nilai =, +=, -=, *=, /=, %=.

Code :

```
MODUL 4 Java script > JS assign.js > ...  
1 let x = 6;  
2 x +=3 // Sama dengan x = x + 3  
3 console.log("Hasil x += 3:", x); //9
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
C:\Program Files\nodejs\node.exe .\MODUL 4 Java script\assign.js  
Hasil x += 3: 9
```

3. Bitwise

Penjelasan : Melakukan operasi pada representasi bit: &, |, ^, ~, <<, >>.

Code :

MODUL 4

Javascript

"

```
MODUL 4 Java script > JS bitwish.js > ...
1 let c = 6; //0101
2 let d = 3; //0011
3 console.log("Bitwise AND", c & d); //1
4 console.log("Bitwise OR", c | d); //7
5 console.log("Bitwise XOR", c ^ d); //6
6 console.log("Bitwise NOT", ~c); //7
```

Output :

```
C:\Program Files\nodejs\node.exe .\MODUL 4 Java script\bitwish.js
Bitwise AND 2
Bitwise OR 7
Bitwise XOR 5
Bitwise NOT -7
```

4. Pembandingan

Penjelasan : Digunakan untuk membandingkan nilai: `==`, `!=`, `===`, `!==`, `>`, `<`, `>=`, `<=`.

Code :

```
MODUL 4 Java script > JS pembandingan.js > ...
1 let m = 10, n = "10";
2 console.log("m == n:", m == n); //true
3 console.log("m == n:", m === n); //false
4 console.log("m > 5:", m > 5); //true
5 console.log("m <= 10 :", m <= 10 ); //true
```

Output :

```
Filter (e.g. text, !exclude, \escape)
C:\Program Files\nodejs\node.exe
m == n: true
m == n: false
m > 5: true
m <= 10 : true
```

5. Logika

Penjelasan : Digunakan untuk logika boolean: `&&`, `||`, `!`.

Code :

```
MODUL 4 Java script > JS Logika.js > ...
1 let isLoggedIn = true;
2 let isAdmin = false;
3 console.log("&&:", isLoggedIn && isAdmin); //false
4 console.log("||:", isLoggedIn || isAdmin); //true
5 console.log("!", !isLoggedIn); //false
```

MODUL 4

Javascript

"

Output :

```
C:\Program Files\nodejs
&&: false
||: true
!: false
```

6. String

Penjelasan : Digunakan untuk menggabungkan string dengan **+** atau **+=**.

Code :

```
MODUL 4 Java script > JS string.js > ...
1 let firstName = "moh.";
2 let lastName = "Andris";
3 console.log("Nama Lengkap:", firstName + " " + lastName); //Moh. Andris
4
```

Output :

```
C:\Program Files\nodejs\node.
Nama Lengkap: moh. Andris
```

4.2.4 Jenis dan Tipe Data Dalam JavaScript

Tipe data dalam javascript diantaranya adalah tipe data **angka**, **tipe data text (string)**, **dan tipe data boolean**. Selain itu dalam JavaScript ada tipe data objek. Contoh tipe data objek adalah **tipe data tanggal (date)**, **array**, **dan fungsi**.

1. Tipe data Angka

JavaScript tidak membedakan tipe data angka (number) antara angka bulat dengan angka desimal, atau tidak membedakan antara bilangan integer dengan float. Seluruh tipe data angka di dalam JavaScript disimpan dalam bentuk desimal (float). Karena di dalam JavaScript sebuah variabel tidak perlu dideklarasikan akan bertipe apa, maka jika sebuah variabel diberikan nilai angka,

MODUL 4

Javascript

"

maka variabel tersebut telah menjadi variabel dengan tipe angka.

Buatlah contoh Tipe data angka pada JavaScript :

Code :

```
MODUL 4 Java script > JS typedata.js > ...  
1 let angkaBulat = 100;           //bilangan bulat(integer)  
2 let angkaDesimal = 3.15;        //bilangan desimal(float)
```

2. Tipe Data String

Tipe data String di dalam JavaScript adalah tipe data yang terdiri dari kumpulan karakter yang berurutan. Atau di dalam penggunaan sehari-hari string adalah tipe data yang menampung nilai **text** atau **kalimat**. Untuk membuat sebuah tipe data string, kita hanya tinggal menambahkan tanda kutip (**bahasa inggris: 'quotes'**) pada awal dan akhir dari text. JavaScript mendukung penggunaan tanda kutip satu (") maupun tanda kutip ganda (""). Didalam sumber bahasa inggris sering disebut sebagai **single quote** dan **double quote**.

Buatlah contoh Tipe data String pada JavaScript :

Code :

```
MODUL 4 Java script > JS typedatastring.js > ...  
1 let namaDepan = 'Moh.';  
2 let namaTengah = 'Andris';  
3 let namaBelakang = 'Saputra';
```

3. Tipe data Boolean

Tipe data **Boolean** adalah tipe data yang hanya mempunyai dua nilai, yakni benar (True) atau salah (False). Tipe data boolean sering digunakan untuk membuat alur logika program. Struktur logika seperti if, else, while, dan do while, membutuhkan nilai boolean sebagai **"pengontrol" alur program**. Tipe data

MODUL 4

Javascript

"

boolean juga merupakan hasil yang didapat dari operasi perbandingan.

Misalkan apakah **variabel a** sama **dengan b**, atau **apakah a** lebih besar **dari b**.

Untuk membuat tipe data boolean di dalam JavaScript, kita cukup memberikan nilai true, atau false ke dalam sebuah variabel. Berikut adalah contoh pembuatan tipe data boolean di dalam JavaScript:

Buatlah contoh Tipe data Boolean pada JavaScript :

Code :

```
MODUL 4 Java script > JS tipedataboolean.js > ...  
1  let sudahLogin = true;  
2  let isAdmin = false;  
3  let perbandingan = 10 > 5; //true
```

4.3 Percabangan dan Perulangan

Pada dasarnya dalam JavaScript terdapat dua macam pernyataan percabangan yaitu **if..else** dan **switch**.

4.3.1 Percabangan

1. If .. else

Pernyataan ini digunakan untuk menguji sebuah kondisi dan kemudian mengeksekusi pernyataan tertentu bila kondisi tersebut terpenuhi, dan mengeksekusi pernyataan lain bila kondisi tersebut tidak terpenuhi.

MODUL 4

Javascript

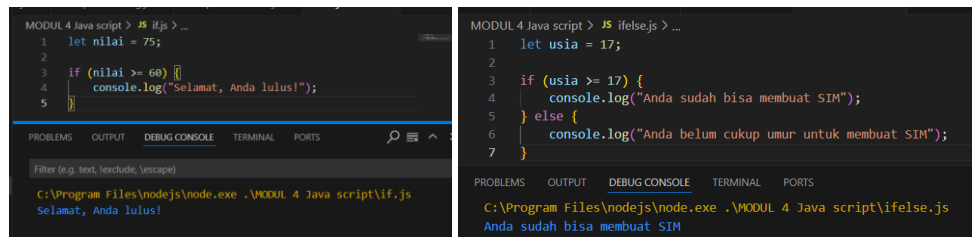
"



Gambar 4. 5. Contoh if.. else

Buatlah contoh percabangan if, if else dan else dalam JavaScript:

Code :



2. Switch

Selain menggunakan if..else, percabangan juga dapat ditangani dengan perintah switch. Dengan kata lain pernyataan switch digunakan untuk menyederhanakan pernyataan if..else yang terlalu banyak.

Buatlah contoh percabangan Switch dalam JavaScript:

MODUL 4

Javascript

"

Code :

```
MODUL 4 Java script > JS percabangaswitch.js > ...
1  let hari = "Rabu";
2  let aktivitas;
3
4  switch (hari) {
5      case "Senin":
6          aktivitas = "Meeting mingguan";
7          break;
8      case "Selasa":
9          aktivitas = "Pelatihan karyawan";
10         break;
11     case "Rabu":
12         aktivitas = "Work from home";
13         break;
14     case "Kamis":
15         aktivitas = "Presentasi proyek";
16         break;
17     case "Jumat":
18         aktivitas = "Evaluasi mingguan";
19         break;
20     default:
21         aktivitas = "Libur akhir pekan";
22 }
23
24 console.log(`Hari ${hari}, aktivitas: ${aktivitas}`);
25 // Output: "Hari Rabu, aktivitas: Work from home"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

C:\Program Files\nodejs\node.exe .\MODUL 4 Java script\percabangaswitch.js
Hari Rabu, aktivitas: Work from home

4.3.2 Perulangan

Untuk mengulang kejadian beberapa kali maka kita membutuhkan proses perulangan. Pada JavaScript dikenal beberapa metode/cara perulangan.

1. Perulangan For

Digunakan untuk mengeksekusi pernyataan-pernyataan beberapa kali. Perulangan For paling sering dipakai, jika anda sudah tahu akhir dari perulangan tersebut. . Perintah for mengulang suatu loop sampai kondisi menghasilkan evaluasi true atau loop keluar dengan perintah break.

Buatlah contoh pengulangan menggunakan For dalam JavaScript:

MODUL 4

Javascript

Code :

```
MODUL 4 Java script > JS perulangan.js > ...
1  // Mencetak angka 1 sampai 5
2  for (let i = 1; i <= 5; i++) {
3      console.log(i);
4  }
```

2. Perulangan While

Perulangan lain yang dapat digunakan adalah dengan menggunakan perintah While. Perintah while digunakan untuk perulangan yang tidak diketahui berapa kali proses perulangannya. Perintah while terus mengulangi loop selama kondisi memiliki nilai true.

Buatlah contoh pengulangan menggunakan while dalam JavaScript:

Code :

```
MODUL 4 Java script > JS perulangan while.js > ...
1  // Mencetak angka 1 sampai 5
2  let i = 1;
3  while (i <= 5) {
4      console.log(i);
5      i++;
6  }
7
```

3. Perulangan Do While

Perulangan ini hampir sama seperti while, digunakan apabila kita belum tahu berapa kali perulangan harus dilakukan. Bedanya pernyataan do..while pengujiannya dilakukan di akhir pernyataan.

Buatlah contoh pengulangan menggunakan Do while dalam JavaScript:

MODUL 4

Javascript

Code :

```
MODUL 4 Java script > JS perulangan do while.js > ...  
1  // Mencetak angka 1 sampai 5  
2  let i = 1;  
3  do {  
4      console.log(i);  
5      i++;  
6  } while (i <= 5);  
7  |
```

MODUL 4

Javascript

"

Tugas

1. Buatlah perhitungan sebuah bentuk bangun menggunakan Javascript (kubus, tabung, persegi panjang dll)
2. Dengan tambahan HTML dan CSS akan dapat nilai tambah
3. Memiliki input akan dapat nilai tambah
4. Kumpulkan pada GC