

MODUL 4 - Javascript Web Programming



Disusun oleh :

Willy Rafael F. Silalahi_23083000168_4G

**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 Screenshoot
2. Output diisikan dengan Screenshoot
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
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 diinput diletakkan diantara tag pembuka <script> dan tag penutup </script>

Contoh Code:

A screenshot of a code editor with a dark background and light-colored text. The code is HTML with an embedded JavaScript script. The script calculates the volume of a rectangular prism (balok) based on length (P), width (L), and height (T). It uses document.write() to output the results for each dimension and the final volume calculation. The code is color-coded: HTML tags are in red, JavaScript code is in green, and comments are in yellow.

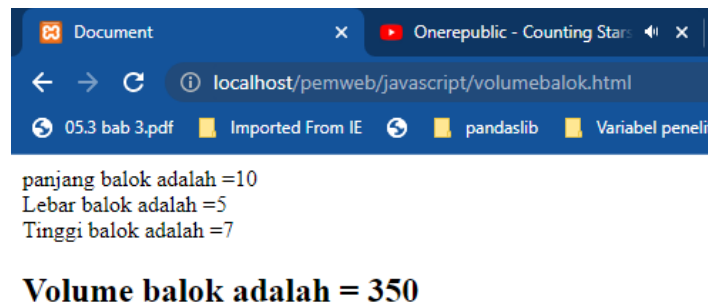
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script>
    var P = 10;
    var L = 5;
    var T = 7;
    volume = P * L * T;
    document.write("panjang balok adalah =" + P + "<br>")
    document.write("Lebar balok adalah   =" + L + "<br>")
    document.write("Tinggi balok adalah  =" + T + "<br>")
  </script>
</head>
<body>
  <h2>Volume balok adalah = <script>document.write(volume)</script></h2>
</body>
</html>
```

Gambar 4. 1. Internal JavaScript

MODUL 4

Javascript

Output :



Gambar 4. 2. Output Internal JavaScript

Buatlah contoh internal JavaScript :

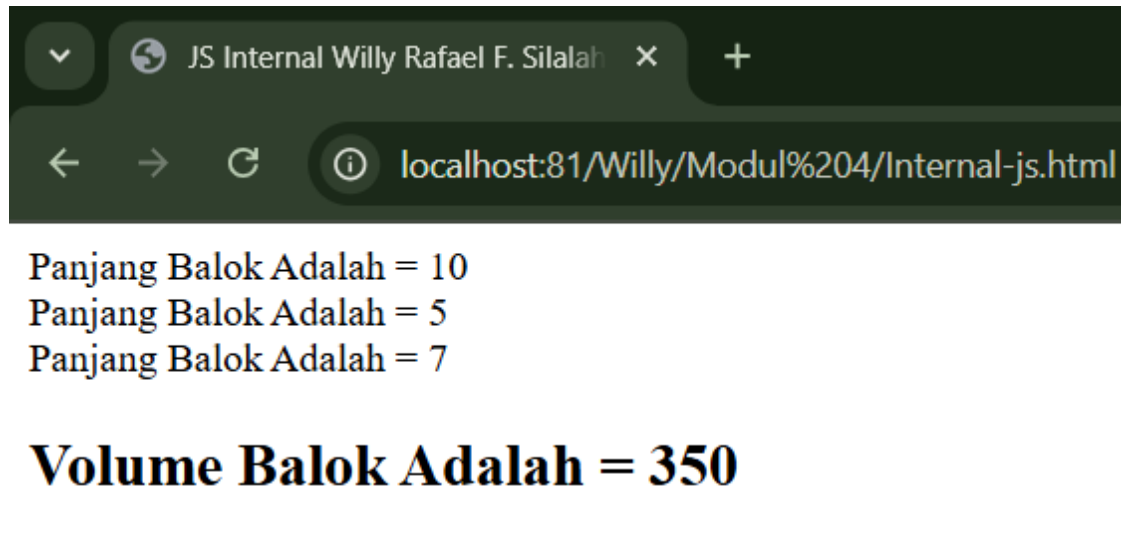
Code :

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JS Internal Willy Rafael F. Silalahi</title>
7   <script>
8     var P = 10;
9     var L = 5;
10    var T = 7;
11    volume = P * L * T;
12    document.write('Panjang Balok Adalah = ' + P + "<br>")
13    document.write('Panjang Balok Adalah = ' + L + "<br>")
14    document.write('Panjang Balok Adalah = ' + T + "<br>")
15  </script>
16 </head>
17 <body>
18   <h2>Volume Balok Adalah = <script>document.write(volume)</script></h2>
19 </body>
20 </html>
```

MODUL 4

Javascript

Output :



4.1.2 External Javascript <script src="">

Cara atau metode kedua untuk menginput kode JavaScript ke dalam halaman 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 :

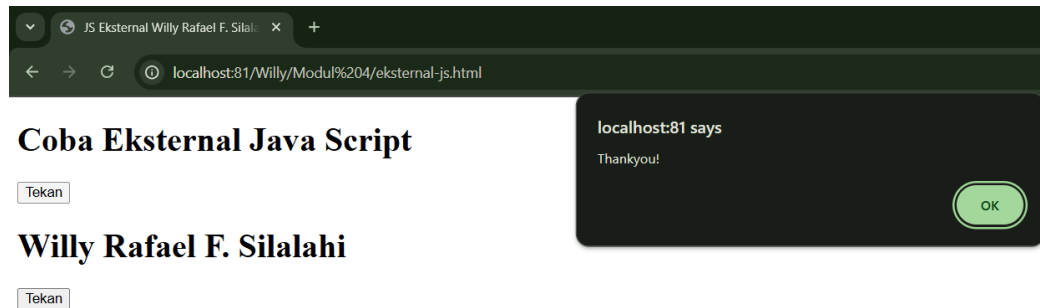
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>JS Eksternal Willy Rafael F. Silalahi</title>
7     <script src="Eksternal-js.js" defer></script>
8 </head>
9 <body>
10    <h1 id="text">Coba Eksternal Java Script</h1>
11    <button id="notifyButton">Tekan</button>
12    <br>
13    <h1 id="text2">Willy Rafael F. Silalahi</h1>
14    <button id="changeTextButton">Tekan</button>
15 </body>
16 </html>
```

```
1 document.getElementById("notifyButton").addEventListener("click", function() {
2     alert("Thankyou!");
3 });
4
5 document.getElementById("changeTextButton").addEventListener("click", function() {
6     const textElement = document.getElementById("text2");
7
8     if (textElement.textContent === "Willy Rafael F. Silalahi") {
9         textElement.textContent = "FTI Unmer Malang!";
10    } else {
11        textElement.textContent = "Willy Rafael F. Silalahi";
12    }
13 });
```

MODUL 4

Javascript

Output :



4.1.3 Inline JavaScript (Event Handler)

Cara ketiga untuk menjalankan JavaScript adalah dengan memanggilnya menggunakan Event Handler dari dalam tag HTML. secara sederhana, 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, di arahkanmouse, 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 terbagai 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 elemet 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 |
| onLoad | Event ini dibangkitkan ketika suatu objek selesai ditampilkan pada halaman web. Objek yang dimaksud pada keterangan ini adalah objek |

MODUL 4

Javascript

| | |
|----------|---|
| | 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 focusnya |
| onChange | Event ini dibangkitkan ketika sebuah element (dalam hal ini adalah element select, text dan textarea) telah diubah nilainya sebelum element tersebut kehilangan focusnya |
| onClick | Event ini dibangkitkan ketika terjadi aksi klik (dilakukan oleh pengguna) terhadap element, dimana element yang dimaksud adalah semua element dari form yang dapat diklik 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 Anda. |

MODUL 4

Javascript

| | |
|-------------|---|
| 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 (diblok) |
| 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 :

Code :

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JS Inline Willy Rafael F. Silalahi</title>
7 </head>
8 <body>
9   <h1>Contoh Inline JavaScript</h1>
10  <p id="demo">Tekan tombolnya rek</p>
11  <button onclick="(changeText())">Klik Disini</button>
12
13  <script>
14    function changeText(){
15      document.getElementById('demo').textContent = 'Selamat, Teks Telah Diubah Menggunakan Javascript!';
16      alert('Thankyou!')
17    }
18  </script>
19 </body>
20 </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 protocol http//: menjadi javascript.

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JS URL Willy Rafael F. Silalahi</title>
7 </head>
8 <body>
9   <h1>Ini adalah contoh Javascript menggunakan URL</h1>
10  <p id="message">Coba klik link dibawah ini unntuk membuktikan Javascript ini menggunakan URL!</p>
11  <a href="javascript:changeMessage()">Klik Disini</a>
12
13  <script>
14    function changeMessage() {
15      document.getElementById('message').textContent = "Selamat anda terkena link phising!";
16      alert('selamat anda terkena phising')
17    }
18  </script>
19 </body>
20 </html>
```

MODUL 4

Javascript

Output :



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 :

Code :



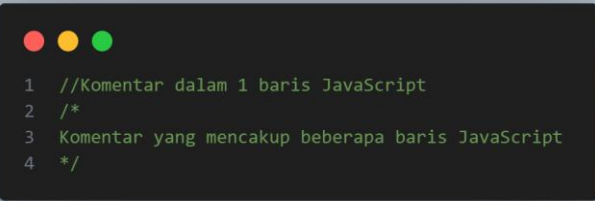
```
1 <script>
2   var name = "Argi";
3   var name = "Willy Rafael";
4
5   function showMessage() {
6       alert("This is showMessage function");
7   }
8
9   function showMessage() {
10      alert("This is showMessage function");
11  }
12 </script>
```

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 :



```
1 //Komentar dalam 1 baris JavaScript
2 /*
3 Komentar yang mencakup beberapa baris JavaScript
4 */
```

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 :

```
1 <script>
2 // Deklarasi variabel dengan var
3 var name = "Willy Rafael F. Silalahi";
4 var age = 21;
5 var isStudent = true;
6
7 // Deklarasi dengan let
8 let city = "Malang";
9 let country = "Indonesia";
10
11 // Deklarasi variasi dengan const
12 const pi = 3.14;
13 </script>
```

2. Cara Membuat Variabel JavaScript

Cara membuat variabel di dalam JavaScript di bedakan 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 :



```
1 <script>
2   // Deklarasi variabel dengan var
3   var a; // Deklarasi variabel a
4   var b; // Deklarasi variabel b
5   var c; // Deklarasi variabel c
6   var d; // Deklarasi variabel d
7   var e; // Deklarasi variabel e
8   var f; // Deklarasi variabel f
9   var g; // Deklarasi variabel g
10
11  // Deklarasi variabel tidak dengan var
12  a; // Deklarasi variabel a
13  b; // Deklarasi variabel b
14  c; // Deklarasi variabel c
15  d; // Deklarasi variabel d
16  e; // Deklarasi variabel e
17  f; // Deklarasi variabel f
18  g; // Deklarasi variabel g
19 </script>
```


MODUL 4

Javascript

Pembuatan variabel tanpa menggunakan keyword var memang lebih cepat, akan 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. Pembanding
5. Logika
6. String

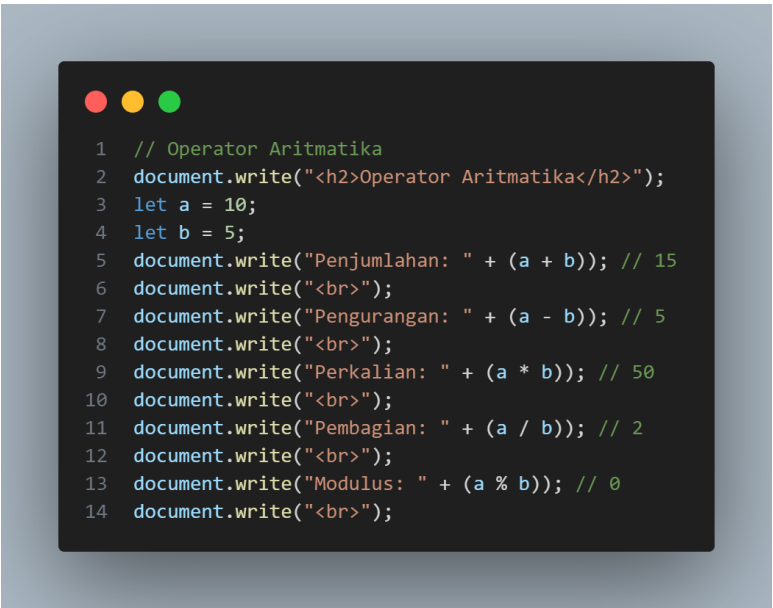
Jelaskan dan praktikan dengan mengcoding ke enam operator di atas, dengan menscreenshoot code anda serta hasil output pada tampilan browser. Letakan screenshoot anda pada tabel berikut :

Buatlah Operator dalam JavaScript :

1. Aritmatika

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

Code :



```
1 // Operator Aritmatika
2 document.write("<h2>Operator Aritmatika</h2>");
3 let a = 10;
4 let b = 5;
5 document.write("Penjumlahan: " + (a + b)); // 15
6 document.write("<br>");
7 document.write("Pengurangan: " + (a - b)); // 5
8 document.write("<br>");
9 document.write("Perkalian: " + (a * b)); // 50
10 document.write("<br>");
11 document.write("Pembagian: " + (a / b)); // 2
12 document.write("<br>");
13 document.write("Modulus: " + (a % b)); // 0
14 document.write("<br>");
```

Output :

Operator Aritmatika

Penjumlahan: 15

Pengurangan: 5

Perkalian: 50

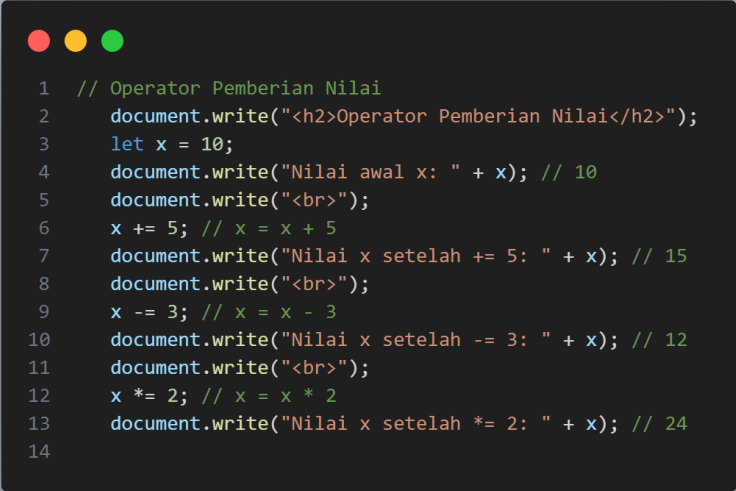
Pembagian: 2

Modulus: 0

2. Assign

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

Code:



```
1 // Operator Pemberian Nilai
2 document.write("<h2>Operator Pemberian Nilai</h2>");
3 let x = 10;
4 document.write("Nilai awal x: " + x); // 10
5 document.write("<br>");
6 x += 5; // x = x + 5
7 document.write("Nilai x setelah += 5: " + x); // 15
8 document.write("<br>");
9 x -= 3; // x = x - 3
10 document.write("Nilai x setelah -= 3: " + x); // 12
11 document.write("<br>");
12 x *= 2; // x = x * 2
13 document.write("Nilai x setelah *= 2: " + x); // 24
14
```

Output:

Operator Pemberian Nilai

Nilai awal x: 10

Nilai x setelah += 5: 15

Nilai x setelah -= 3: 12

Nilai x setelah *= 2: 24

3. Bitwise

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

Code :



```
1 //Operator Pemanipulasian bit (bitwise)
2 document.write("<h2>Operator Pemanipulasian bit (bitwise)</h2>");
3 let y = 5; // 0101
4 let z = 3; // 0011
5 document.write("y: " + y + " (0101)");
6 document.write("<br>");
7 document.write("z: " + z + " (0011)");
8 document.write("<br>");
9 document.write("AND: " + (y & z)); // 1 (0001)
10 document.write("<br>");
11 document.write("OR: " + (y | z)); // 7 (0111)
12 document.write("<br>");
13 document.write("XOR: " + (y ^ z)); // 6 (0110)
14 document.write("<br>");
15 document.write("NOT: " + (~y)); // -6 (1010)
```

Output:

Operator Pemanipulasian bit (bitwise)

y: 5 (0101)

z: 3 (0011)

AND: 1

OR: 7

XOR: 6

NOT: -6

4. Pembandingan

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

Code

```
1 //Operator Pembandingan
2 document.write("<h2>Operator Pembandingan</h2>");
3 let a1 = 10;
4 let b1 = 5;
5 document.write("a: " + a1);
6 document.write("<br>");
7 document.write("b: " + b1);
8 document.write("<br>");
9 document.write("a == b: " + (a1 == b1)); // false
10 document.write("<br>");
11 document.write("a != b: " + (a1 != b1)); // true
12 document.write("<br>");
13 document.write("a > b: " + (a1 > b1)); // true
14 document.write("<br>");
15 document.write("a < b: " + (a1 < b1)); // false
16 document.write("<br>");
17 document.write("a >= b: " + (a1 >= b1)); // true
18 document.write("<br>");
19 document.write("a <= b: " + (a1 <= b1)); // false
20 document.write("<br>");
21 document.write("a === b: " + (a1 === b1)); // false
```

Output


Operator Pemanding

```
a: 10  
b: 5  
a == b: false  
a != b: true  
a > b: true  
a < b: false  
a >= b: true  
a <= b: false  
a === b: false
```

5. Logika

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

Code



```
1 //Operator Logika  
2 document.write("<h2>Operator Logika</h2>");  
3 let x1 = true;  
4 let y1 = false;  
5 document.write("x: " + x1);  
6 document.write("<br>");  
7 document.write("y: " + y1);  
8 document.write("<br>");  
9 document.write("x && y: " + (x1 && y1)); // false  
10 document.write("<br>");  
11 document.write("x || y: " + (x1 || y1)); // true  
12 document.write("<br>");  
13 document.write("!x: " + (!x1)); // false  
14 document.write("<br>");  
15 document.write("!y: " + (!y1)); // true  
16 document.write("<br>");
```

Output

Operator Logika

```
x: true  
y: false  
x && y: false  
x || y: true  
!x: false  
!y: true
```

6. String

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

Code

```
1 //Operator String  
2 document.write("<h2>Operator String</h2>");  
3 let str1 = "Hello";  
4 let str2 = "World";  
5 document.write("str1: " + str1);  
6 document.write("<br>");  
7 document.write("str2: " + str2);  
8 document.write("<br>");  
9 document.write("str1 + str2: " + (str1 + " " + str2)); // Hello World  
10 document.write("<br>");  
11 document.write("str1.length: " + str1.length); // 5  
12 document.write("<br>");  
13 document.write("str2.length: " + str2.length); // 5  
14 document.write("<br>");
```

Output

Operator String

```
str1: Hello  
str2: World  
str1 + str2: Hello World  
str1.length: 5  
str2.length: 5
```

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 di deklarasikan akan bertipe apa, maka jika sebuah variabel diberikan nilai angka, maka variabel tersebut telah menjadi variabel dengan tipe angka.

Buatlah contoh Tipe data angka pada JavaScript :

Code :



```
1 <script>
2     // Tipe data angka dalam JavaScript
3     var integer = 42; // Bilangan bulat
4     var floatNumber = 3.14; // Bilangan pecahan
5     var negativeNumber = -7; // Bilangan negatif
6
7     // Menampilkan tipe data angka
8     document.write("Integer: " + integer);
9     document.write("<br>");
10    document.write("Float Number: " + floatNumber);
11    document.write("<br>");
12    document.write("Negative Number: " + negativeNumber);
13 </script>
```

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 :

```
1  <script>
2      // String dengan menggunakan double quotes
3      var str1 = "Hello, World!";
4      document.write(str1); // Output: Hello, World!
5
6      document.write("<br>");
7
8      // String dengan menggunakan single quotes
9      var str2 = 'Hello, World!';
10     document.write(str2); // Output: Hello, World!
11
12     document.write("<br>");
13
14     // String dengan menggunakan backticks (template literals)
15     var str3 = `Hello, World!`;
16     document.write(str3); // Output: Hello, World!
17
18     document.write("<br>");
19
20     // Menggunakan string interpolation
21     var name = "Willy Rafael F. Silalahi";
22     var greeting = `Hello, ${name}!`;
23     document.write(greeting); // Output: Hello, Willy Rafael F. Silalahi!
24 </script>
```

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 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 :

```
1  <script>
2    // Boolean dengan menggunakan true dan false
3    var isTrue = true; // Boolean true
4    var isFalse = false; // Boolean false
5
6    document.write("isTrue: " + isTrue); // Output: isTrue: true
7    document.write("<br>");
8    document.write("isFalse: " + isFalse); // Output: isFalse: false
9
10   document.write("<br>");
11
12   // Boolean dengan menggunakan operator perbandingan
13   var a = 5;
14   var b = 10;
15   var isEqual = (a == b); // false
16   var isNotEqual = (a != b); // true
17
18   document.write("isEqual: " + isEqual); // Output: isEqual: false
19   document.write("<br>");
20   document.write("isNotEqual: " + isNotEqual); // Output: isNotEqual: true
21 </script>
```

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.

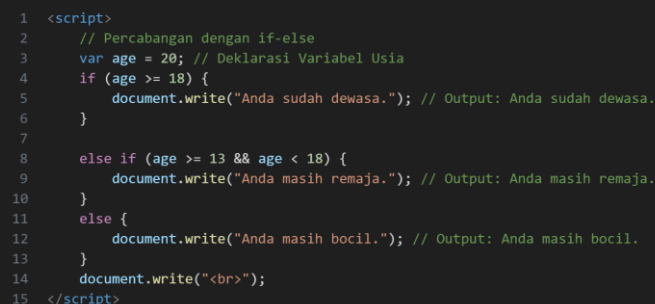


```
if (kondisi)
{
  //pernyataan1 dieksekusi
  //bila kondisi terpenuhi
}
else
{
  //pernyataan2 dieksekusi
  //bila kondisi tidak terpenuhi
}
```

Gambar 4. 5. Contoh if.. else

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

Code :



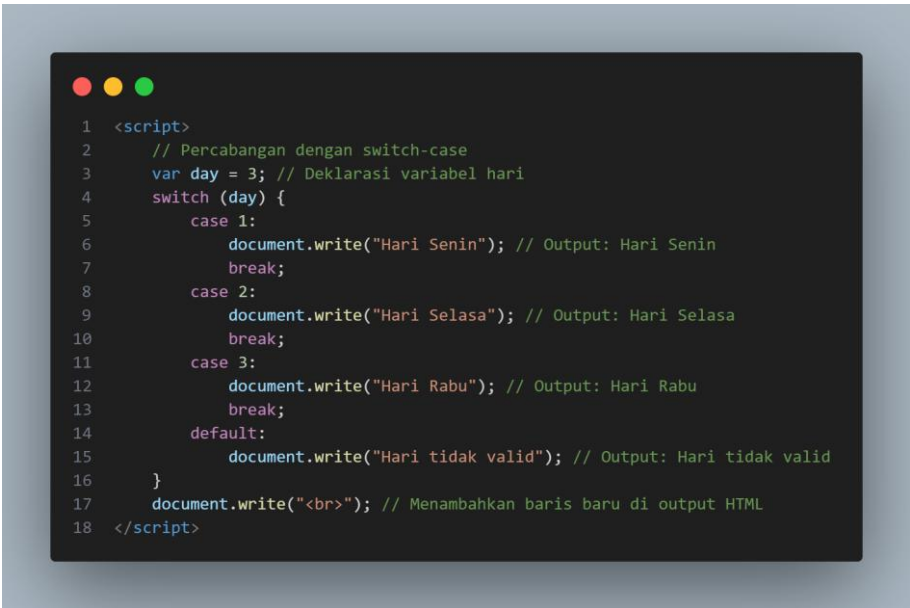
```
1 <script>
2   // Percabangan dengan if-else
3   var age = 20; // Deklarasi Variabel Usia
4   if (age >= 18) {
5     document.write("Anda sudah dewasa."); // Output: Anda sudah dewasa.
6   }
7
8   else if (age >= 13 && age < 18) {
9     document.write("Anda masih remaja."); // Output: Anda masih remaja.
10  }
11  else {
12    document.write("Anda masih bocil."); // Output: Anda masih bocil.
13  }
14  document.write("<br>");
15 </script>
```

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:

Code :

A screenshot of a code editor with a dark background and light-colored text. The code is a JavaScript switch-case statement. It starts with a script tag, followed by a comment about switch-case branching. A variable 'day' is declared and set to 3. A switch statement follows, with three cases: case 1 (output 'Hari Senin'), case 2 (output 'Hari Selasa'), and case 3 (output 'Hari Rabu'). A default case outputs 'Hari tidak valid'. After the switch, a document.write call adds a new line to the HTML output. The code ends with a closing script tag.

```
1 <script>
2 // Percabangan dengan switch-case
3 var day = 3; // Deklarasi variabel hari
4 switch (day) {
5     case 1:
6         document.write("Hari Senin"); // Output: Hari Senin
7         break;
8     case 2:
9         document.write("Hari Selasa"); // Output: Hari Selasa
10        break;
11    case 3:
12        document.write("Hari Rabu"); // Output: Hari Rabu
13        break;
14    default:
15        document.write("Hari tidak valid"); // Output: Hari tidak valid
16    }
17    document.write("<br>"); // Menambahkan baris baru di output HTML
18 </script>
```

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:

Code :



```
1 <script>
2 // Perulangan for untuk mencetak angka 1 sampai 5
3 for (var i = 1; i <= 5; i++) {
4     document.write("Angka: " + i + "<br>");
5 }
6
7 document.write("<br>");
8
9 // Perulangan for untuk mencetak angka genap dari 2 sampai 10
10 for (var j = 2; j <= 10; j += 2) {
11     document.write("Angka Genap: " + j + "<br>");
12 }
13 </script>
```

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 For dalam JavaScript:

Code :

```
1 <script>
2 // Perulangan while untuk mencetak angka 1 sampai 5
3 var i = 1;
4 while (i <= 5) {
5     document.write("Angka: " + i + "<br>");
6     i++;
7 }
8
9 document.write("<br>");
10
11 // Perulangan while untuk mencetak angka genap dari 2 sampai 10
12 var j = 2;
13 while (j <= 10) {
14     document.write("Angka Genap: " + j + "<br>");
15     j += 2;
16 }
17 </script>
```

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 For dalam JavaScript:

Code :

```
1 <script>
2 // Perulangan do-while untuk mencetak angka 1 sampai 5
3 var i = 1;
4 do {
5     document.write("Angka: " + i + "<br>");
6     i++;
7 } while (i <= 5);
8
9 document.write("<br>");
10
11 // Perulangan do-while untuk mencetak angka genap dari 2 sampai 10
12 var j = 2;
13 do {
14     document.write("Angka Genap: " + j + "<br>");
15     j += 2;
16 } while (j <= 10);
17 </script>
```

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