

MODUL 4 - Javascript Web Programming



Disusun oleh :

Muhammad Rhosid Narendra 23083000095

**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:



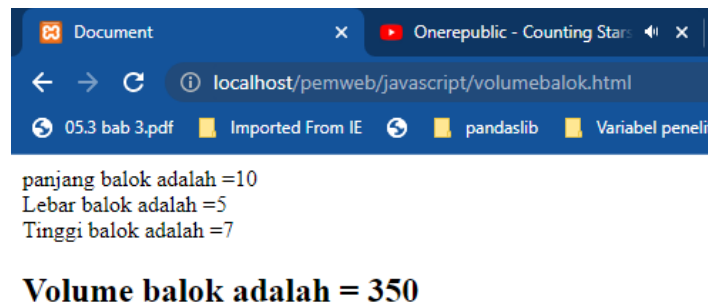
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script>
    var P = 10;
    var L = 5;
    var T = 7;
    volume = P * L * 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 :

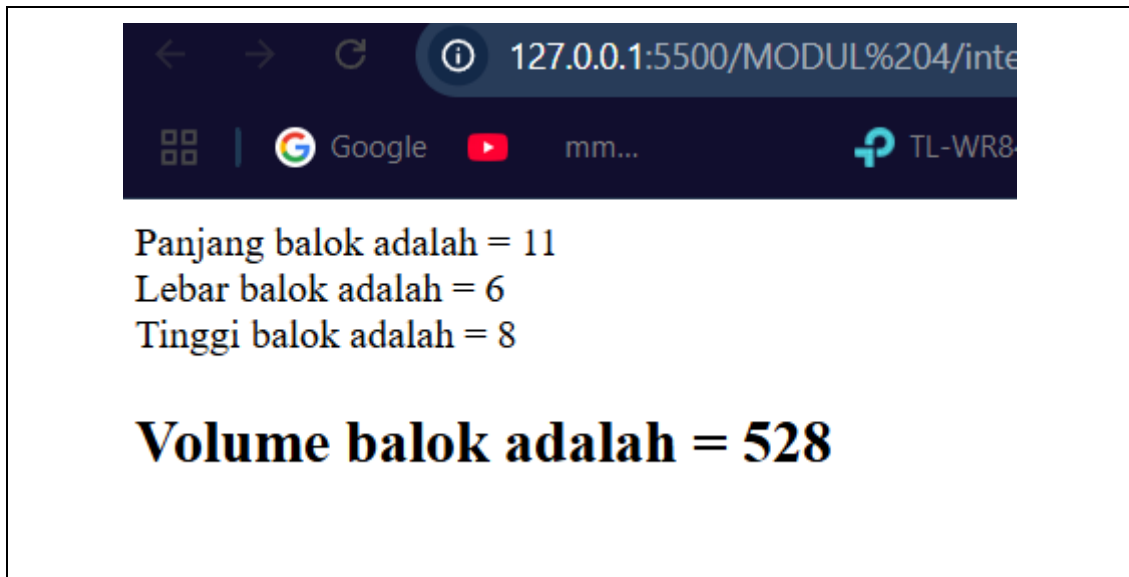
Code :

```
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>Volume balok</title>
7   <script>
8     var P = 11;
9     var L = 6;
10    var T = 8;
11    var volume = P * L * T;
12
13    document.write("Panjang balok adalah = " + P + "<br>");
14    document.write("Lebar balok adalah = " + L + "<br>");
15    document.write("Tinggi balok adalah = " + T + "<br>");
16  </script>
17 </head>
18 <body>
19   <h2>Volume balok adalah = <script>document.write(volume)</script></h2>
20 </body>
21 </html>
```

Output :

MODUL 4

Javascript



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 :

MODUL 4

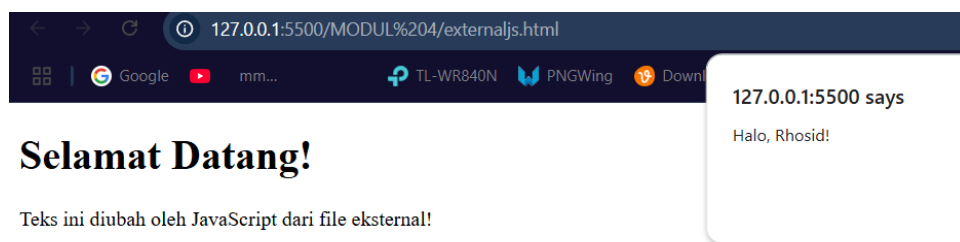
Javascript

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>external Rhosid</title>
7 </head>
8 <body>
9   <h1>Selamat Datang!</h1>
10  <p id="demo">Ini adalah paragraf contoh.</p>
11
12  <script src="externaljs.js"></script>
13 </body>
14 </html>
```

```
1 document.getElementById("demo").innerHTML = "Teks ini diubah oleh JavaScript dari file eksternal!";
2 console.log("Halo dari file script.js!");
3 function sapa(nama) {
4   alert("Halo, " + nama + "!");
5 }
6
7 setTimeout(function() {
8   sapa("Rhosid");
9 }, 3000);
```

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

MODUL 4

Javascript

adalah pemanggilan kode javascript ketika „sesuatu“ terjadi dalam tag HTML.

Sesuatu disini maksudnya ketika sebuah element dalam HTML di klik, di klik kanan, di arahkan 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 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 element 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 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

MODUL 4

Javascript

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.
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 .

MODUL 4

Javascript

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>Inline Rhosid</title>
7 </head>
8 <body>
9   <h1 onmouseover="this.style.color='red'">Arahkan Mouse ke sini!</h1>
10  <p onclick="this.textContent='Teks ini berubah!'">Klik paragraf ini!</p>
11  <button onclick="alert('Tombol ini diklik!')">Click Here!</button>
12 </body>
13 </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 metoda cara memasukkan JavaScript lainnya adalah:

1. Menyederhanakan halaman HTML dengan memindahkan seluruh kode 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.

MODUL 4

Javascript

Buatlah contoh JavaScript menggunakan URL:

Code :

```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1" />
6   <title>Contoh JavaScript Eksternal</title>
7 </head>
8 <body>
9   <h1>Selamat Datang di Halaman Utama</h1>
10  <p id="pesan">Pesan awal...</p>
11
12  <script src="url.js"></script>
13 </body>
14 </html>
```

```
1 document.addEventListener('DOMContentLoaded', function() {
2   console.log("File script.js berhasil dimuat dan dijalankan!");
3   const pesanElement = document.getElementById('pesan');
4   if (pesanElement) {
5     pesanElement.textContent = "Halo dari file JavaScript eksternal!";
6   }
7 });
```

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 // Deklarasi beberapa variabel dengan perbedaan huruf kapital
2 var unname = "Muhammad";
3 var Unname = "Rhosid";
4 var UNname = "Narendra";
5
6 // Menampilkan nilai dari masing-masing variabel
7 console.log(unname); // Output: Muhammad
8 console.log(Unname); // Output: Rhosid
9 console.log(UNname); // Output: Narendra
10
11 // Mencoba mengakses variabel dengan penulisan yang berbeda (akan error)
12 // console.log(unNam); // Uncaught ReferenceError: unNam is not defined
13
14 // Contoh penggunaan keyword yang benar dan salah
15 var angka = 5;
16 while (angka > 0) { // Penulisan keyword 'while' yang benar
17     console.log("Angka saat ini: " + angka);
18     angka--;
19 }
20
21 // Contoh penulisan keyword 'While' yang salah (akan error)
22 // var nomor = 3;
23 // While (nomor > 0) {
24 //     console.log("Nomor saat ini: " + nomor);
25 //     nomor--;
26 // }
27
28 // Contoh fungsi dengan perbedaan huruf kapital
29 function sapaUnname(unname) {
30     console.log("Halo, " + unname + "!");
31 }
32
33 function SapaUnname(unname) {
34     console.warn("Fungsi ini memiliki nama yang mirip tapi berbeda karena case.");
35 }
36
37 sapaUnname("Rhosid"); // Output: Halo, Rhosid!
38 SapaUnname("Narendra"); // Output: Peringatan di konsol: Fungsi ini memiliki nama yang mirip tapi berbeda karena case.
39
40 // // Memanggil fungsi dengan penulisan berbeda yang tidak dideklarasikan
41 // Sapaunname("Cid"); // akan error jika tidak dideklarasikan
```

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 // Ini adalah komentar satu baris
2 var nama = "Rhosid"; // Menyimpan nama ke dalam variabel
3
4 /*
5 Ini adalah komentar beberapa baris.
6 Komentar ini dapat mencakup lebih dari satu baris.
7 Misalnya, kita bisa menjelaskan fungsi atau logika yang lebih kompleks di sini.
8 */
```

4.2.2 Variabel dalam JavaScript

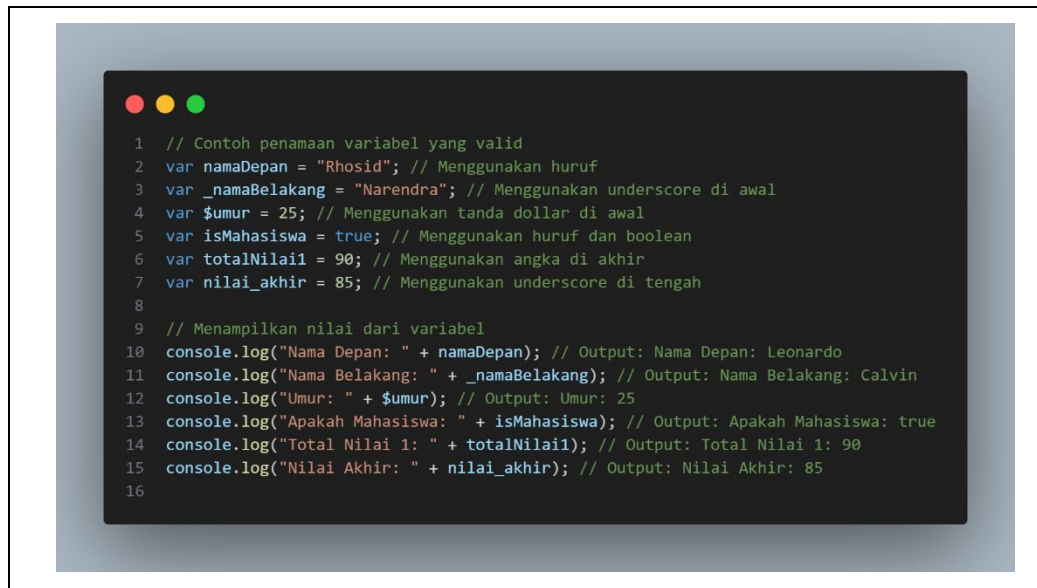
Dalam bahasa pemrograman, variabel adalah “**penampung**” sebuah nilai. Tergantung dengan “**nilai**” dari variabel tersebut, sebuah variabel di dalam JavaScript dapat bertipe **Angka (Number)**, **String**, **Boolean**, atau yang lainnya.

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 :



2. Cara Membuat Variabel JavaScript

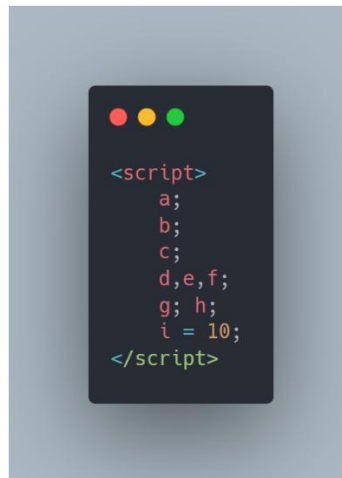
Cara membuat variabel di dalam JavaScript di bedakan menjadi 2, yakni dengan menggunakan keyword **var**, dan **tanpa var**.

contoh penulisan menggunakan kata kunci var :



Gambar 4. 3. Menggunakan var

contoh penulisan menggunakan tanpa kata kunci var :

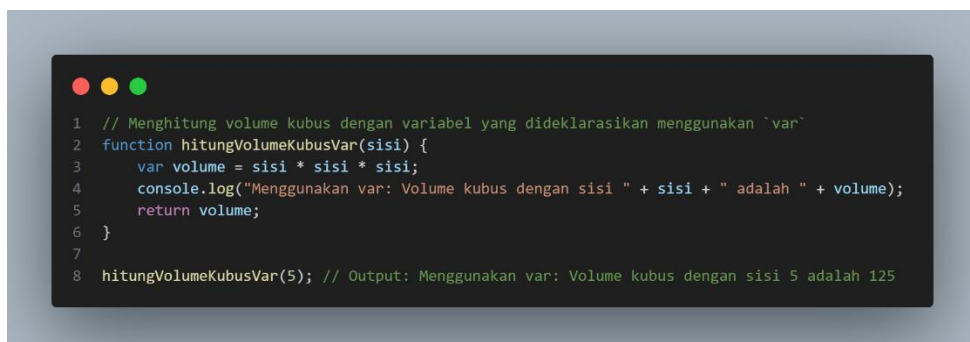


Gambar 4. 4. Tidak menggunakan var

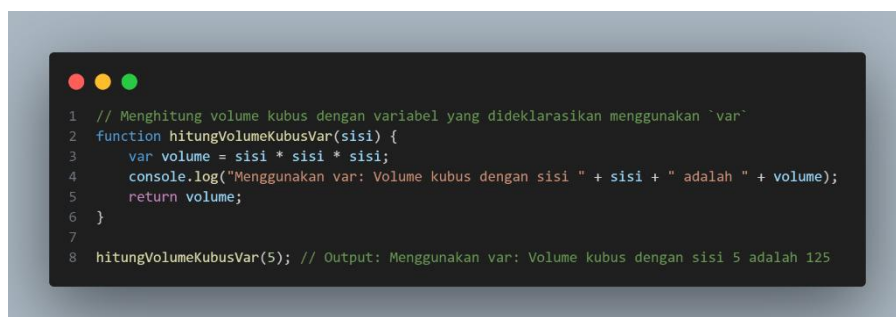
Buatlah contoh penamaan Variabel pada JavaScript :

Code :

var



Non var



Pembuatan variabel tanpa menggunakan keyword var memang lebih cepat, akan tetapi **tidak disarankan**.

MODUL 4

Javascript

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 menscreenshoot code anda serta hasil output pada tampilan browser. Letakan screenshoot anda pada tabel berikut :

Buatlah Operator dalam JavaScript :

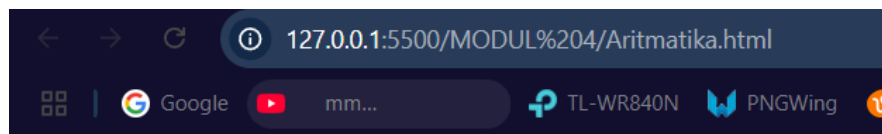
1. Aritmatika

Penjelasan: Operator aritmatika digunakan untuk operasi matematika seperti penambahan (+), pengurangan (-), perkalian (*), pembagian (/), dan sisa bagi (%).

Code :

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator Aritmatika Rhosid</title>
5 </head>
6 <body>
7   <h2>Operator Aritmatika</h2>
8   <script>
9     let a = 10;
10    let b = 3;
11
12    let tambah = a + b;
13    let kurang = a - b;
14    let kali = a * b;
15    let bagi = a / b;
16    let modulo = a % b;
17
18    document.write("Penjumlahan 10 + 3 = " + tambah + "<br>");
19    document.write("Pengurangan 10 - 3 = " + kurang + "<br>");
20    document.write("Perkalian 10 * 3 = " + kali + "<br>");
21    document.write("Pembagian 10 / 3 = " + bagi + "<br>");
22    document.write("Sisa bagi 10 % 3 = " + modulo + "<br>");
23  </script>
24 </body>
25 </html>
26
```

Output :



Operator Aritmatika

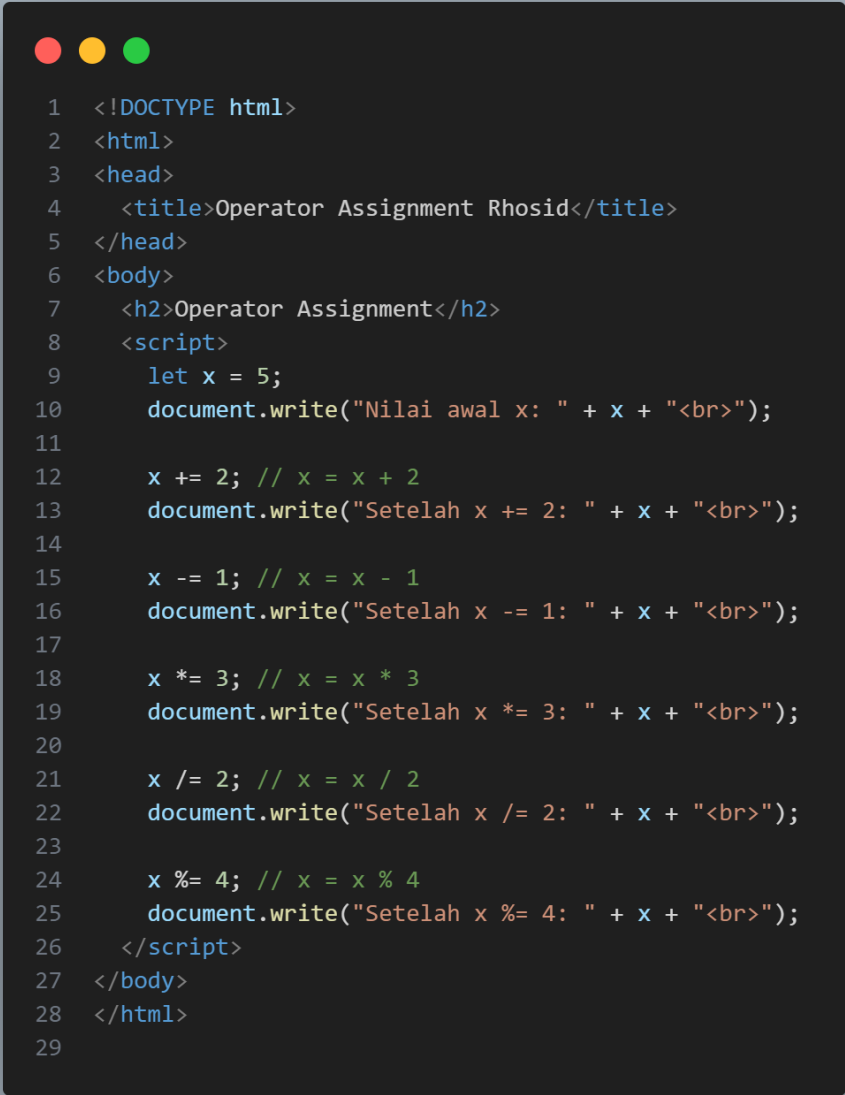
Penjumlahan $10 + 3 = 13$
Pengurangan $10 - 3 = 7$
Perkalian $10 * 3 = 30$
Pembagian $10 / 3 = 3.3333333333333335$
Sisa bagi $10 \% 3 = 1$

2. Assignment

Penjelasan: Operator assignment digunakan untuk menetapkan nilai ke variabel. Operator gabungan seperti $+=$, $-=$, $*=$, dan lainnya memperbarui nilai variabel secara

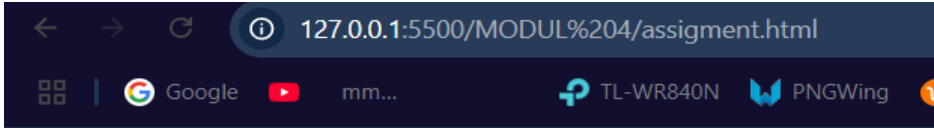
langsung.

Code:



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator Assignment Rhosid</title>
5 </head>
6 <body>
7   <h2>Operator Assignment</h2>
8   <script>
9     let x = 5;
10    document.write("Nilai awal x: " + x + "<br>");
11
12    x += 2; // x = x + 2
13    document.write("Setelah x += 2: " + x + "<br>");
14
15    x -= 1; // x = x - 1
16    document.write("Setelah x -= 1: " + x + "<br>");
17
18    x *= 3; // x = x * 3
19    document.write("Setelah x *= 3: " + x + "<br>");
20
21    x /= 2; // x = x / 2
22    document.write("Setelah x /= 2: " + x + "<br>");
23
24    x %= 4; // x = x % 4
25    document.write("Setelah x %= 4: " + x + "<br>");
26  </script>
27 </body>
28 </html>
29
```

Output



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:5500/MODUL%204/assignment.html`. The browser's toolbar includes icons for Google, YouTube, and other search engines. The page content is as follows:

Operator Assignment

Nilai awal x: 5
Setelah `x += 2`: 7
Setelah `x -= 1`: 6
Setelah `x *= 3`: 18
Setelah `x /= 2`: 9
Setelah `x %= 4`: 1

3. Bitwise

Penjelasan: Bitwise bekerja pada representasi biner dari angka. Misalnya `&` (AND), `|` (OR), `^` (XOR), `~` (NOT), `<<` (shift kiri), `>>` (shift kanan).

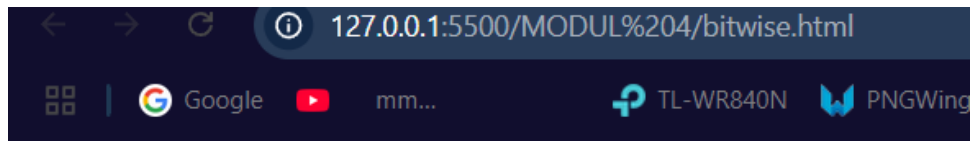
Code:

MODUL 4

Javascript

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Operator Bitwise</title>
5  </head>
6  <body>
7    <h2>Operator Bitwise</h2>
8    <script>
9      let a = 5;    // 0101
10     let b = 3;    // 0011
11
12     document.write("a = 5 (0101)<br>");
13     document.write("b = 3 (0011)<br><br>");
14
15     document.write("a & b = " + (a & b) + " (AND)<br>");
16     document.write("a | b = " + (a | b) + " (OR)<br>");
17     document.write("a ^ b = " + (a ^ b) + " (XOR)<br>");
18     document.write("~a = " + (~a) + " (NOT)<br>");
19     document.write("a << 1 = " + (a << 1) + " (Shift Kiri)<br>");
20     document.write("a >> 1 = " + (a >> 1) + " (Shift Kanan)<br>");
21   </script>
22 </body>
23 </html>
24
```

Output



Operator Bitwise

a = 5 (0101)

b = 3 (0011)

a & b = 1 (AND)

a | b = 7 (OR)

a ^ b = 6 (XOR)

~a = -6 (NOT)

a << 1 = 10 (Shift Kiri)

a >> 1 = 2 (Shift Kanan)

4. Pembanding

Penjelasan: Operator pembanding digunakan untuk membandingkan dua nilai dan mengembalikan true atau false. Contoh: ==, ===, !=, >, <, >=, <=.

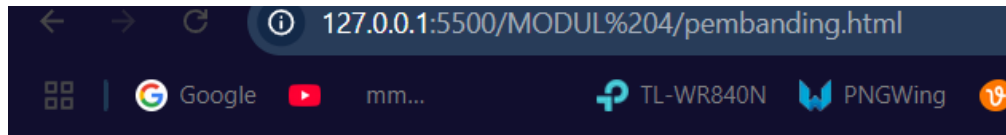
Code

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Operator Pembanding</title>
5  </head>
6  <body>
7    <h2>Operator Pembanding</h2>
8    <script>
9      let a = 5;
10     let b = '5';
11     let c = 7;
12
13     document.write("a == b: " + (a == b) + "<br>"); // true, karena nilainya sama
14     document.write("a === b: " + (a === b) + "<br>"); // false, karena tipe data berbeda
15     document.write("a != c: " + (a != c) + "<br>"); // true
16     document.write("a > c: " + (a > c) + "<br>"); // false
17     document.write("a < c: " + (a < c) + "<br>"); // true
18     document.write("a >= 5: " + (a >= 5) + "<br>"); // true
19     document.write("c <= 10: " + (c <= 10) + "<br>"); // true
20   </script>
21 </body>
22 </html>
23

```

Output



Operator Pembanding

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

5. Logika

Penjelasan: Digunakan untuk menggabungkan ekspresi boolean. Contoh: && (dan), || (atau), ! (tidak).

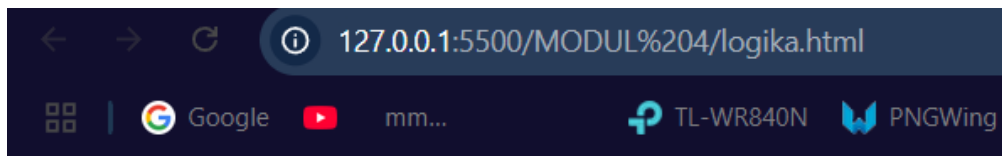
Code

MODUL 4

Javascript

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator Logika Rhosid</title>
5 </head>
6 <body>
7   <h2>Operator Logika</h2>
8   <script>
9     let umur = 20;
10    let punyaKTP = true;
11
12    let bisaVote = umur >= 17 && punyaKTP;
13    let tidakPunyaKTP = !punyaKTP;
14
15    document.write("Bisa memilih (umur >=17 dan punya KTP): " + bisaVote + "<br>");
16    document.write("Tidak punya KTP: " + tidakPunyaKTP + "<br>");
17
18    let cukupUmurAtauKTP = umur >= 17 || punyaKTP;
19    document.write("Cukup umur atau punya KTP: " + cukupUmurAtauKTP + "<br>");
20  </script>
21 </body>
22 </html>
23
```

Output



Operator Logika

Bisa memilih (umur >=17 dan punya KTP): true

Tidak punya KTP: false

Cukup umur atau punya KTP: true

6. String

Penjelasan: Operator + dapat digunakan untuk menggabungkan (concatenate) dua atau lebih string.

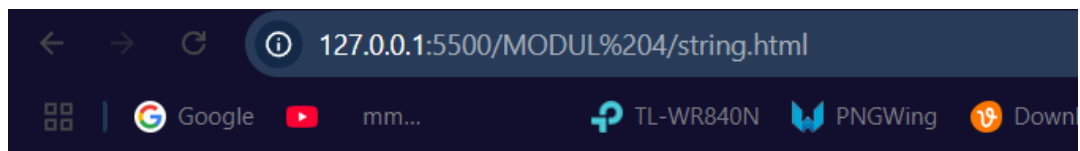
Code

MODUL 4

Javascript

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator String Rhosid</title>
5 </head>
6 <body>
7   <h2>Operator String</h2>
8   <script>
9     let namaDepan = "Rhosid";
10    let namaBelakang = "Narendra";
11
12    let namaLengkap = namaDepan + " " + namaBelakang;
13    document.write("Nama lengkap: " + namaLengkap + "<br>");
14
15    let salam = "Halo, " + namaLengkap + "!";
16    document.write("Salam: " + salam + "<br>");
17  </script>
18 </body>
19 </html>
20
```

Output



Operator String

Nama lengkap: Rhosid Narendra
Salam: Halo, Rhosid Narendra!

4.2.4 Jenis dan Tipe Data Dalam JavaScript

Tipe data dalam javascript diantaranya adalah tipe data **angka**, **tipe data text**

MODUL 4

Javascript

(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 // Contoh tipe data angka dalam JavaScript
2
3 let angkaBulat = 42;           // Bilangan bulat
4 let angkaDesimal = 3.14;      // Bilangan desimal (float)
5 let hasilPerhitungan = angkaBulat + angkaDesimal; // Operasi penjumlahan
6
7 console.log("Angka Bulat:", angkaBulat);           // Output: 42
8 console.log("Angka Desimal:", angkaDesimal);       // Output: 3.14
9 console.log("Hasil Penjumlahan:", hasilPerhitungan); // Output: 45.14
10
```

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

MODUL 4

Javascript

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 // Contoh tipe data String dalam JavaScript
2
3 let namaDepan = 'Rhosid';           // Menggunakan tanda kutip satu (single quote)
4 let namaBelakang = "Narendra";      // Menggunakan tanda kutip ganda (double quote)
5 let kalimat = "Halo, nama saya " + namaDepan + " " + namaBelakang + "."; // Menggabungkan string
6
7 console.log(namaDepan);             // Output: Rhosid
8 console.log(namaBelakang);          // Output: Narendra
9 console.log(kalimat);               // Output: Halo, nama saya Rina Saputri.
10
```

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 // Contoh tipe data Boolean dalam JavaScript
2
3 let benar = true;           // Nilai boolean true
4 let salah = false;         // Nilai boolean false
5
6 // Contoh boolean dari hasil perbandingan
7 let a = 10;
8 let b = 5;
9
10 let hasil1 = a > b;         // true, karena 10 lebih besar dari 5
11 let hasil2 = a === b;      // false, karena 10 tidak sama dengan 5
12
13 // Contoh penggunaan dalam struktur logika
14 if (hasil1) {
15     console.log("a lebih besar dari b");
16 } else {
17     console.log("a tidak lebih besar dari b");
18 }
19
20 console.log("Nilai benar:", benar);    // Output: true
21 console.log("Nilai salah:", salah);    // Output: false
22 console.log("Apakah a > b?:", hasil1); // Output: true
23 console.log("Apakah a === b?:", hasil2); // Output: false
24
```

MODUL 4

Javascript

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  // Contoh penggunaan percabangan if, if...else, dan else
2
3  let nilai = 75;
4
5  // Menggunakan if
6  if (nilai >= 90) {
7      console.log("Nilai kamu A");
8  }
9
10 // Menggunakan if...else
11 if (nilai >= 70) {
12     console.log("Kamu lulus");
13 } else {
14     console.log("Kamu tidak lulus");
15 }
16
17 // Menggunakan if...else if...else
18 if (nilai >= 90) {
19     console.log("Predikat: Sangat Baik");
20 } else if (nilai >= 80) {
21     console.log("Predikat: Baik");
22 } else if (nilai >= 70) {
23     console.log("Predikat: Cukup");
24 } else {
25     console.log("Predikat: Kurang");
26 }
27
```

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 :

```
1  // Contoh percabangan switch
2
3  let hari = "Senin";
4
5  switch (hari) {
6      case "Senin":
7          console.log("Hari ini kamu harus semangat kerja!");
8          break;
9      case "Selasa":
10         console.log("Tetap semangat, minggu masih panjang.");
11         break;
12      case "Rabu":
13         console.log("Sudah di tengah minggu, tetap fokus!");
14         break;
15      case "Kamis":
16         console.log("Hampir sampai di akhir minggu!");
17         break;
18      case "Jumat":
19         console.log("Hari terakhir kerja, semangat!");
20         break;
21      case "Sabtu":
22      case "Minggu":
23         console.log("Waktunya istirahat, selamat libur!");
24         break;
25      default:
26         console.log("Hari tidak dikenali.");
27  }
28
```

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 // Contoh penggunaan perulangan for
2
3 // Menampilkan angka dari 1 sampai 5
4 for (let i = 1; i <= 5; i++) {
5     console.log(i); // Output: 1, 2, 3, 4, 5
6 }
7
8 // Menjumlahkan angka dari 1 sampai 10
9 let total = 0;
10 for (let i = 1; i <= 10; i++) {
11     total += i; // Menambahkan nilai i ke total
12 }
13 console.log("Jumlah total dari 1 sampai 10 adalah:", total); // Output: 55
14
15 // Perulangan dengan array
16 let buah = ["Apel", "Jeruk", "Pisang", "Mangga"];
17 for (let i = 0; i < buah.length; i++) {
18     console.log(buah[i]); // Output: Apel, Jeruk, Pisang, Mangga
19 }
20
```

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 // Contoh penggunaan perulangan while
2
3 // Menampilkan angka dari 1 sampai 5
4 let i = 1;
5 while (i <= 5) {
6     console.log(i); // Output: 1, 2, 3, 4, 5
7     i++; // Increment nilai i setiap kali perulangan
8 }
9
10 // Menjumlahkan angka dari 1 sampai 10
11 let total = 0;
12 let j = 1;
13 while (j <= 10) {
14     total += j; // Menambahkan nilai j ke total
15     j++; // Increment nilai j setiap kali perulangan
16 }
17 console.log("Jumlah total dari 1 sampai 10 adalah:", total); // Output: 55
18
19 // Perulangan dengan kondisi yang lebih kompleks
20 let angka = 1;
21 while (angka <= 10) {
22     if (angka % 2 === 0) { // Memeriksa apakah angka genap
23         console.log(angka); // Output: 2, 4, 6, 8, 10
24     }
25     angka++;
26 }
27
```

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 // Contoh penggunaan perulangan do...while
2
3 // Menampilkan angka dari 1 sampai 5
4 let i = 1;
5 do {
6     console.log(i); // Output: 1, 2, 3, 4, 5
7     i++; // Increment nilai i setiap kali perulangan
8 } while (i <= 5);
9
10 // Menjumlahkan angka dari 1 sampai 10
11 let total = 0;
12 let j = 1;
13 do {
14     total += j; // Menambahkan nilai j ke total
15     j++; // Increment nilai j setiap kali perulangan
16 } while (j <= 10);
17 console.log("Jumlah total dari 1 sampai 10 adalah:", total); // Output: 55
18
19 // Perulangan dengan kondisi lebih kompleks
20 let angka = 1;
21 do {
22     if (angka % 2 === 0) { // Memeriksa apakah angka genap
23         console.log(angka); // Output: 2, 4, 6, 8, 10
24     }
25     angka++;
26 } while (angka <= 10);
27
```

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

CSS

```
1  body {
2    font-family: Arial, sans-serif;
3    background-color: #f4f4f9;
4    margin: 0;
5    padding: 0;
6  }
7
8  .container {
9    width: 80%;
10   max-width: 600px;
11   margin: 50px auto;
12   padding: 20px;
13   background-color: #fff;
14   border-radius: 8px;
15   box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
16 }
17
18 h1 {
19   text-align: center;
20   color: #333;
21 }
22
23 .form-group {
24   margin-bottom: 20px;
25 }
26
27 label {
28   display: block;
29   margin-bottom: 5px;
30   font-weight: bold;
31 }
32
33 input {
34   width: 100%;
35   padding: 8px;
36   margin-bottom: 10px;
37   border-radius: 4px;
38   border: 1px solid #ccc;
39 }
40
41 button {
42   padding: 10px 20px;
43   background-color: #4CAF50;
44   color: white;
45   border: none;
46   border-radius: 5px;
47   cursor: pointer;
48 }
49
50 button:hover {
51   background-color: #45a049;
52 }
53
54 p {
55   font-size: 18px;
56   font-weight: bold;
57 }
58
```

code

MODUL 4

Javascript

```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Perhitungan Volume Limas Segiempat Rhosid</title>
7   <link rel="stylesheet" href="limasc.css">
8 </head>
9 <body>
10  <div class="container">
11    <h1>Perhitungan Volume Limas Segiempat Rhosid</h1>
12
13    <!-- Input untuk Limas Segiempat -->
14    <div class="form-group">
15      <label for="sisiAlas">Panjang Sisi Alas (cm):</label>
16      <input type="number" id="sisiAlas" placeholder="Masukkan panjang sisi alas">
17      <label for="tinggiLimas">Tinggi Limas (cm):</label>
18      <input type="number" id="tinggiLimas" placeholder="Masukkan tinggi limas">
19      <button onclick="hitungLimas()">Hitung Volume Limas</button>
20      <p id="hasilLimas"></p>
21    </div>
22  </div>
23
24  <script src="limasc.js"></script>
25 </body>
26 </html>
27
```

JS

```
1 // Fungsi untuk menghitung volume limas segiempat
2 function hitungLimas() {
3   let sisiAlas = document.getElementById("sisiAlas").value;
4   let tinggiLimas = document.getElementById("tinggiLimas").value;
5
6   // Menghitung volume limas segiempat dengan rumus: (1/3) * sisiAlas^2 * tinggiLimas
7   let volumeLimas = (1 / 3) * Math.pow(sisiAlas, 2) * tinggiLimas;
8
9   // Menampilkan hasil volume limas segiempat
10  document.getElementById("hasilLimas").innerText = "Volume Limas Segiempat: " + volumeLimas.toFixed(2) + " cm³";
11 }
12
```

Output

Perhitungan Volume Limas Segiempat Rhosid

Panjang Sisi Alas (cm):

Tinggi Limas (cm):

Hitung Volume Limas