

# Modul Praktikum

## PERTEMUAN

# 3

IF3028 PEMROGRAMAN WEB

T.A 2021/2022

---

## JAVASCRIPT I

### ASISTEN PRAKTIKUM

- Alexander Diva
- Aminudin Fadila
- Andika Saputra
- Daniel Sipangkar
- Ringgo Galih Sadewo
- Syabana Minggu Noviantosa



**TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI PRODUKSI INDUSTRI DAN INFORMASI**  
**INSTITUT TEKNOLOGI SUMATERA**  
**2021**

## Pengertian Javascript



Gambar 1.1 Logo Javascript

Javascript adalah bahasa pemrograman yang awalnya dirancang untuk berjalan di atas browser. Namun, seiring perkembangan zaman, javascript tidak hanya berjalan di atas browser saja. Javascript juga dapat digunakan pada sisi Server, Game, IoT, Desktop, dsb. Javascript awalnya bernama Mocha, lalu berubah menjadi LiveScript saat browser Netscape Navigator 2.0 rilis versi beta (September 1995). Namun, setelah itu dinamai ulang menjadi Javascript. Terinspirasi dari kesuksesan Javascript, Microsoft mengadopsi teknologi serupa. Microsoft membuat 'Javascript' versi mereka sendiri bernama JScript. Lalu di tanam pada Internet Explorer 3.0. Hal ini mengakibatkan 'perang browser', karena JScript milik Microsoft berbeda dengan Javascript racikan Netscape. Akhirnya pada tahun 1996, Netscape mengirimkan standarisasi ECMA-262 ke Ecma International. Sehingga lahirlah standarisasi kode Javascript bernama ECMAScript atau ES. Saat ini ECMAScript sudah mencapai versi 8 (ES8).

## Javascript Dasar

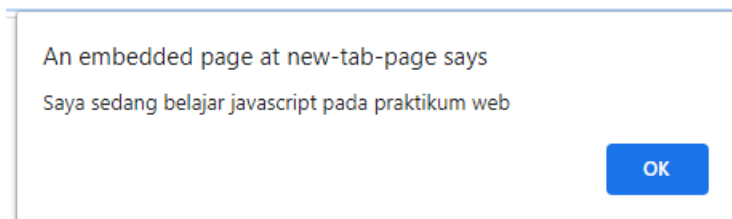
Untuk dapat menggunakan javascript kita dapat melakukan beberapa cara seperti dibawah ini.

### 1.1. Menggunakan Console

Console javascript dapat kita buka dengan klik kanan->Inspect Element->Console atau (ctrl+shift+i)



Gambar 1.2 Console Javascript



Gambar 1.3 Hasil Console Javascript

### 1.2. Embed atau Internal Script

Contoh cara penulisan attribute internal script dapat dilihat pada gambar 1.4

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hello World Javascript</title>
5 </head>
6 <body>
7   <script>
8     alert("contoh embedded/internal script");
9   </script>
10 </body>
11 </html>
```

Gambar 1.4 Contoh penulisan internal script

### 1.3. Inline Kode Javascript

Contoh cara penulisan inline kode javascript dapat dilihat pada gambar 1.5

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hello World Javascript</title>
5 </head>
6 <body>
7   <a href="#" onclick="alert('contoh inline script')">Klik aku!</a>
8 </body>
9 </html>
```

Gambar 1.5 Contoh Penulisan inline kode javascript

### 1.4. Eksternal Javascript

Cara penulisan eksternal javascript dapat anda lihat pada gambar 1.6

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hello World Javascript</title>
5 </head>
6 <body>
7   <script src="eksternal.js"></script>
8 </body>
9 </html>
```

Gambar 1.6 Contoh Penulisan *Eksternal Javascript*

## Javascript Percabangan

Pada javascript kita dapat menggunakan konsep percabangan. Ada beberapa jenis percabangan yang dapat digunakan seperti yang akan kita bahas dibawah.

### 2.1 Percabangan if

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Percabangan if</title>
5 </head>
6 <body>
7   <script>
8     var totalBelanja = prompt("Total belanja?", 0);
9
10    if(totalBelanja > 100000){
11      document.write("<h2>Selamat Anda dapat hadiah</h2>");
12    }
13
14    document.write("<p>Terimakasih sudah berbelanja di toko kami</p>");
15  </script>
16 </body>
17 </html>
```

Gambar 2.1 Percabangan if

### 2.2 Percabangan if - else

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Percabangan if/else</title>
5 </head>
6 <body>
7   <script>
8     var password = prompt("Password:");
9
10    if(password == "open"){
11      document.write("<h2>Selamat datang mahasiswa praktikum!</h2>");
12    } else {
13      document.write("<p>Password salah, coba lagi!</p>");
14    }
15
16    document.write("<p>Terima kasih sudah menggunakan aplikasi ini!</p>");
17
18  </script>
19 </body>
20 </html>
```

Gambar 2.2 Percabangan if - else

## 2.3 Percabangan else if

Percabangan else if merupakan percabangan yang memiliki kondisi percabangan lebih dari satu. Statement else if digunakan untuk memilih kondisi lain jika kondisi pada statement if bernilai salah.

```
if (time < 10) {
    greeting = "Good morning";
} else if (time < 20) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

Gambar 2.3 *else if statement*

## 2.4 Switch Case

Switch case adalah percabangan kode program dimana kita membandingkan isi sebuah variabel dengan beberapa nilai. Jika proses perbandingan tersebut menghasilkan nilai true, maka blok kode program akan dijalankan.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Percabangan switch/case</title>
5 </head>
6 <body>
7   <script>
8     var jawab = prompt("Kamu beruntung! Silahkan pilih hadiahmu dengan memasukan angka 1 sampai 5");
9     var hadiah = "";
10    switch(jawab){
11      case "1":
12        hadiah = "Tisu";
13        break;
14      case "2":
15        hadiah = "1 Kotak Kopi";
16        break;
17      case "3":
18        hadiah = "Sticker";
19        break;
20      case "4":
21        hadiah = "Minyak Goreng";
22        break;
23      default:
24        document.write("<p>Opps! anda salah pilih</p>");
25    }
26
27    if(hadiah == ""){
28      document.write("<p>Kamu gagal mendapat hadiah</p>");
29    } else {
30      document.write("<h2>Selamat kamu mendapatkan " + hadiah + "</h2>");
31    }
32  </script>
33 </body>
34 </html>
```

Gambar 2.4 *Switch Case*

## Javascript Perulangan

Pada javascript kita dapat menggunakan konsep perulangan. Ada beberapa jenis perulangan yang dapat digunakan seperti yang akan kita bahas dibawah.

### 3.1 Perulangan for

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

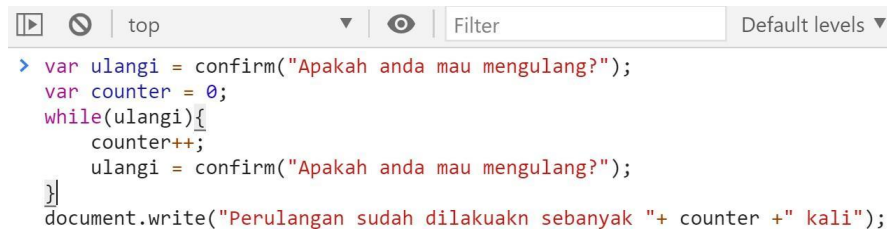


```
Select an element in the page to inspect it Ctrl+Shift+C
> for(counter = 0; counter < 10; counter+=2){
  document.write("<p>Perulangan ke-\"+counter+\"</p>");
}
< undefined
```

Gambar 3.1 Perulangan for

### 3.2 While

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

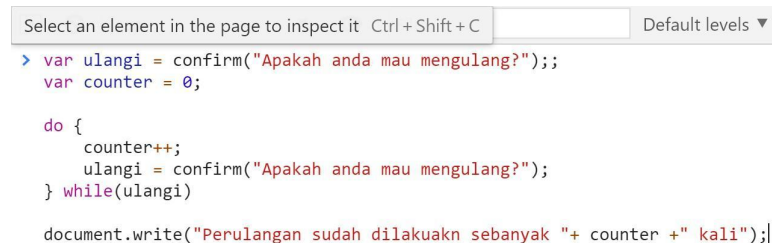


```
top Filter Default levels
> var ulangi = confirm(\"Apakah anda mau mengulang?\");
var counter = 0;
while(ulangi){
  counter++;
  ulangi = confirm(\"Apakah anda mau mengulang?\");
}
document.write(\"Perulangan sudah dilakuakn sebanyak \" + counter + \" kali\");
```

Gambar 3.2 Perulangan While

### 3.3 Do While

Perulangan do/while sama seperti perulangan while. Perbedaan perulangan do/while akan melakukan perulangan sebanyak 1 kali terlebih dahulu, lalu mengecek kondisi yang ada di dalam kurung while.



```
Select an element in the page to inspect it Ctrl+Shift+C Default levels
> var ulangi = confirm(\"Apakah anda mau mengulang?\");
var counter = 0;

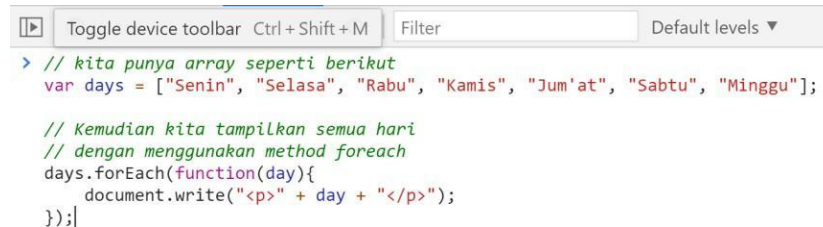
do {
  counter++;
  ulangi = confirm(\"Apakah anda mau mengulang?\");
} while(ulangi)

document.write(\"Perulangan sudah dilakuakn sebanyak \" + counter + \" kali\");
```

Gambar 3.3 Perulangan Do While

### 3.4 Foreach

Perulangan foreach biasanya digunakan untuk mencetak item di dalam array. Perulangan ini termasuk dalam perulangan counted loop, karena jumlah perulangannya akan ditentukan oleh panjang dari array.




```
Toggle device toolbar Ctrl+Shift+M Filter Default levels ▾  
> // kita punya array seperti berikut  
var days = ["Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu", "Minggu"];  
  
// Kemudian kita tampilkan semua hari  
// dengan menggunakan method foreach  
days.forEach(function(day){  
    document.write("<p>" + day + "</p>");  
});|
```

Gambar 3.4 Perulangan Foreach

### 3.5 Dengan Method repeat()

Perulangan dengan method atau fungsi repeat() termasuk dalam perulangan counted loop. Fungsi ini khusus digunakan untuk mengulang sebuah teks (string).



```
top Filter  
> document.write("Harta, Tahta, Nilai A. ".repeat(10));  
.....
```

Gambar 3.5 Perulangan dengan Method repeat()

## Javascript Fungsi

Pada javascript kita dapat menggunakan konsep fungsi. Ada beberapa jenis fungsi yang dapat digunakan seperti yang akan kita bahas dibawah.

### 4.1 Fungsi Tanpa Parameter

Salah satu cara penulisan fungsi pada javascript adalah fungsi tanpa parameter seperti pada gambar dibawah.



```
▶ | top | Filter
> // membuat fungsi
function sayHello(){
  console.log("Hello World!");
}

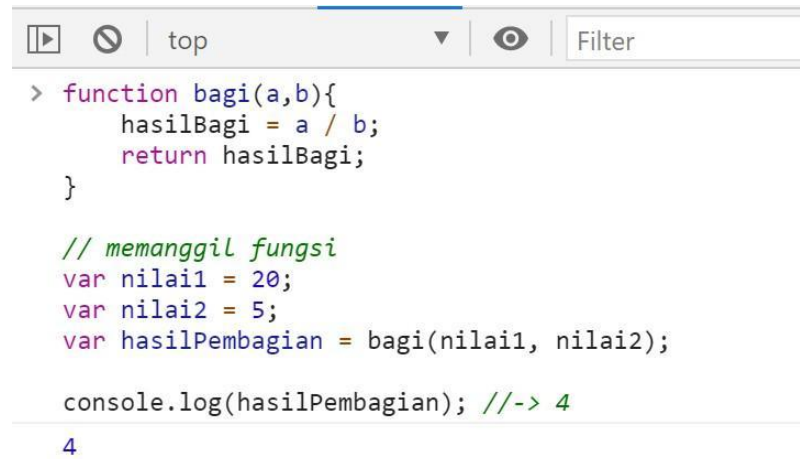
// memanggil fungsi
sayHello() // maka akan menghasilkan -> Hello World!

Hello World!
< undefined
```

Gambar 4.1 Fungsi Tanpa Parameter

### 4.2 Fungsi Dengan Parameter

Selain penulisan fungsi tanpa parameter, fungsi pada javascript juga dapat ditulis dengan menggunakan parameter seperti pada gambar dibawah ini.



```
▶ | top | Filter
> function bagi(a,b){
  hasilBagi = a / b;
  return hasilBagi;
}

// memanggil fungsi
var nilai1 = 20;
var nilai2 = 5;
var hasilPembagian = bagi(nilai1, nilai2);

console.log(hasilPembagian); //-> 4

4
```

Gambar 4.2 Fungsi dengan Parameter

## Document Object Model (DOM)

DOM merupakan singkatan dari Document Object Model. Artinya, dokumen (HTML) yang dimodelkan dalam sebuah objek. Objek dari dokumen ini menyediakan sekumpulan fungsi dan atribut/data yang bisa kita manfaatkan dalam membuat program Javascript. Inilah yang disebut API (Application Programming Interface).

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Memilih Elemen Berdasarkan ID</title>
5 </head>
6 <body>
7
8   <!-- Elemen div yang akan kita pilih dari JS -->
9   <div id="tutorial"></div>
10
11
12   <script type="text/javascript">
13     // mengakses elemen tutorial
14     var tutorial = document.getElementById("tutorial");
15
16     // mengisi teks ke dalam elemen
17     tutorial.innerHTML = "Tutorial Javascript";
18
19     // memberikan CSS ke elemen
20     tutorial.style.backgroundColor = "yellow";
21     tutorial.style.padding = "10px";
22
23   </script>
24
25 </body>
26 </html>
```

Gambar 5.1 Penggunaan DOM pada Javascript

- getElementById() fungsi untuk memilih elemen berdasarkan atribut id.
- getElementByName() fungsi untuk memilih elemen berdasarkan atribut name.
- getElementByClassName() fungsi untuk memilih elemen berdasarkan atribut class.
- getElementByTagName() fungsi untuk memilih elemen berdasarkan nama tag.
- getElementByTagNameNS() fungsi untuk memilih elemen berdasarkan nama tag.
- querySelector() fungsi untuk memilih elemen berdasarkan query.
- querySelectorAll() fungsi untuk memilih elemen berdasarkan query.