

MODUL 3

PENGEMBANGAN APLIKASI PLATFORM KHUSUS
DAVID SETIADI

MODUL 3

Perulangan

Ada 5 bentuk perulangan di javascript yang kemudian dikelompokkan menjadi 2 bagian yaitu Counted Loop dan Uncounted Loop

Perbedaanya:

- **Counted Loop** merupakan perulangan yang jelas dan sudah tentu banyak perulangannya.
- Sedangkan **Uncounted Loop**, merupakan perulangan yang tidak jelas berapa kali ia harus mengulang.





Perulangan yang termasuk dalam *Counted Loop*:

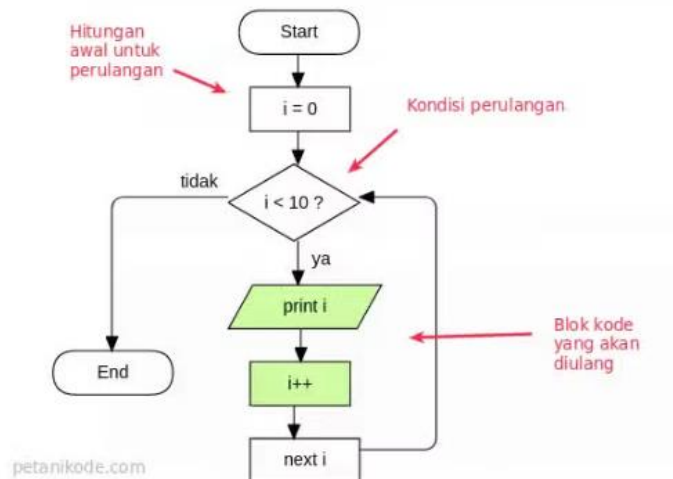
1. Perulangan For
2. Perulangan Foreach
3. Perulangan Repeat

Perulangan yang termasuk dalam *Uncounted Loop*:

1. Perulangan While
2. Perulangan Do/While

1. Perulangan For di Javascript

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



Bentuk kodenya seperti ini :

```

<> belajar.html <> belajar2.html <> belajar3.html X
<> belajar3.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Javascript Dasar</title>
5  </head>
6  <body>
7  |   <script>
8  |   for(let i = 0; i < 10; i++){
9  |       document.write("<p>Perulangan ke-" + i + "</p>")
10 |   }
11 |   </script>
12 </body>
13 </html>
  
```

Yang perlu diperhatikan adalah kondisi yang ada di dalam kurung setelah kata *“for”*.

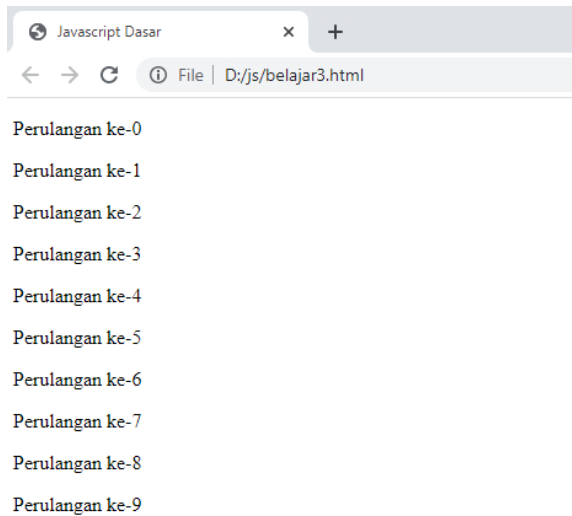
Kondisi ini akan menentukan:

- Hitungan akan dimulai dari 0 (i = 0);
- Hitungannya sampai berapa? Sampai i < 10;
- Lalu di setiap perulangan i akan bertambah +1 (i++).

Variabel i pada perulangan for berfungsi untuk menyimpan nilai hitungan.

Jadi setiap perulangan dilakukan nilai i akan selalu bertambah satu. Karena kita menentukannya di bagian i++.

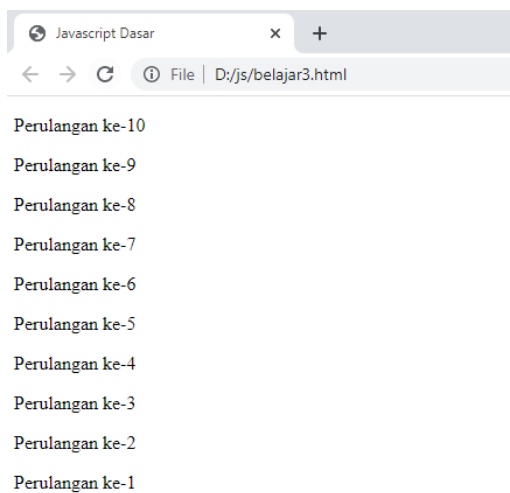
Ini hasil outputnya:



Jika ingin kondisinya dibalik menjadi hitungan mundur maka kodenya :

```
belajar3.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Javascript Dasar</title>
5  </head>
6  <body>
7  |   <script>
8  |   for(let i = 10; i > 0; i--){
9  |       document.write("<p>Perulangan ke-" + i + "</p>")
10 |   }
11 |   </script>
12 </body>
13 </html>
```

Maka hasilnya :



2. Perulangan While di Javascript

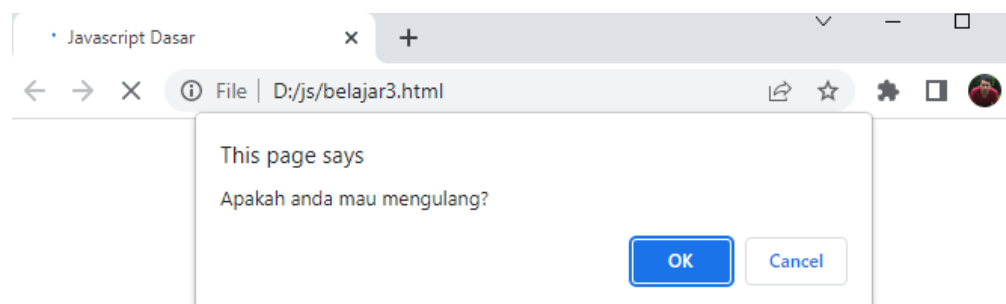
Perulangan while merupakan perulangan yang termasuk dalam perulangan uncounted loop.

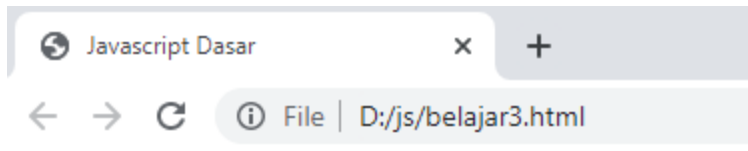
Perulangan while juga dapat menjadi perulangan yang counted loop dengan memberikan counter di dalamnya.

Contoh kodingnya :

```
<> belajar3.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Javascript Dasar</title>
5  </head>
6  <body>
7  |   <script>
8  |   var ulang = confirm("Apakah anda mau mengulang?");
9  |   var counter = 0;
10 |
11 |   while(ulang){
12 |       var jawab = confirm("Apakah anda mau mengulang?")
13 |       counter++;
14 |       if(jawab == false){
15 |           ulang = false;
16 |       }
17 |   }
18 |   document.write("Perulangan sudah dilakuakn sebanyak "+ counter +" kali");
19 |   </script>
20 </body>
21 </html>
```

Hasilnya selama anda menekan ok maka akan terus diulang dan counter akan terus menghitung, dan jika ditekan cancel maka perulangan akan selesai.





Perulangan sudah dilakuakn sebanyak 9 kali

Perulangan akan terjadi selama variabel ulangi bernilai true.

Lalu kita menggunakan fungsi `confirm()` untuk menampilkan dialog konfirmasi.

Selama kita memilih Ok pada dialog konfirmasi, maka variabel ulangi akan terus bernilai true.

Tapi kalau kita pilih Cancel, maka variabel ulangi akan bernilai false.

Saat variabel ulangi bernilai false, maka perulangan akan dihentikan.

3. Perulangan Do/While di Javascript

Perulangan do/while sama seperti perulangan while.

Perbedaanya:

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

Bentuknya seperti ini:

```
do {  
    // blok kode yang akan diulang  
} while (<kondisi>;
```

Jadi perbedaanya:

Perulangan do/while akan mengecek kondisi di belakang (sesudah mengulang), sedangkan while akan mengecek kondisi di depan atau awal (sebelum mengulang).

4. Perulangan Foreach di Javascript

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.

Ada dua cara menggunakan perulangan **foreach** di Javascript:

1. Menggunakan **for** dengan operator **in**;

2. Menggunakan method **forEach()**.

Contoh:

Berikut ini bentuk perulangan "*foreach*" tanpa menggunakan operator **in**:

```
var languages = ["Javascript", "HTML", "CSS", "Typescript"];

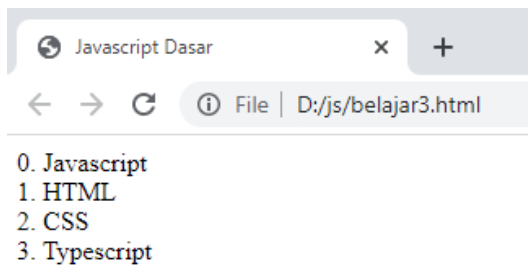
for(i = 0; i < languages.length; i++){
    document.write(i+". " + languages[i] + "<br/>");
}
```

Perulangan ini dapat dibuat lebih sederhana lagi dengan menggunakan operator **in** seperti ini:

```
var languages = ["Javascript", "HTML", "CSS", "Typescript"];

for(i in languages){
    document.write(i+". " + languages[i] + "<br/>");
}
```

Hasilnya :



Cara kedua membuat perulangan *foreach* ialah dengan menggunakan method **forEach()** dari array.

Contoh:


```

<> belajar3.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Javascript Dasar</title>
5  </head>
6  <body>
7  |   <script>
8  |   // kita punya array seperti berikut
9  |   var days = ["Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu", "Minggu"];
10 |
11 |   // Kemudian kita tampilkan semua hari
12 |   // dengan menggunakan method foreach
13 |   days.forEach(function(day){
14 |       document.write("<p>" + day + "</p>");
15 |   });
16 |   </script>
17 </body>
18 </html>

```

Javascript Dasar x +
 File | D:/js/belajar3.html
 Senin
 Selasa
 Rabu
 Kamis
 Jum'at
 Sabtu
 Minggu

5. Perulangan dengan Method repeat()

Perulangan dengan method atau fungsi repeat() termasuk dalam perulangan counted loop.

Fungsi ini khusus digunakan untuk mengulang sebuah teks (string).

Bisa dibilang:

Ini merupakan singkat dari perulangan for.

Contoh:

Apabila kita menggunakan perulangan for:

```

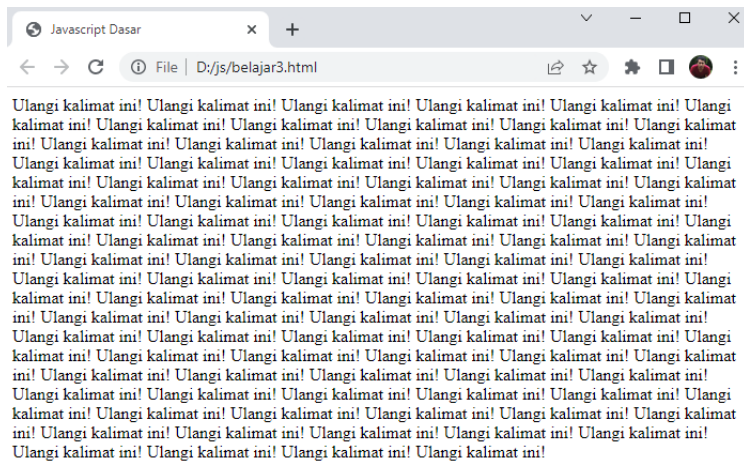
for( let i = 0; i < 100; i++){
    document.write("Ulangi kalimat ini!");
}

```

Apabila kita menggunakan fungsi repeat():

```
document.write("Ulangi kalimat ini! ".repeat(100));
```

Outputnya :



6. Perulangan Bersarang (Nested)

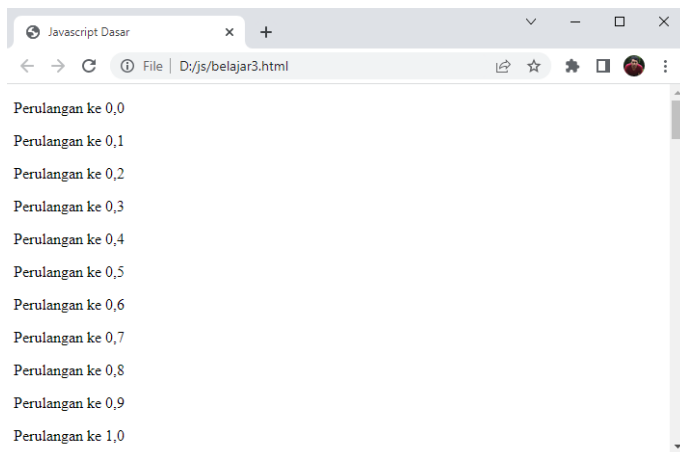
Di dalam blok perulangan, kita juga dapat membuat perulangan.

Ini disebut dengan nested loop atau perulangan bersarang atau perulangan di dalam perulangan.

Mari kita coba lihat contohnya:

```
for(let i = 0; i < 10; i++){  
  for(let j = 0; j < 10; j++){  
    document.write("<p>Perulangan ke " + i + ", " + j + "</p>");  
  }  
}
```

Hasilnya :



Contoh Lain :

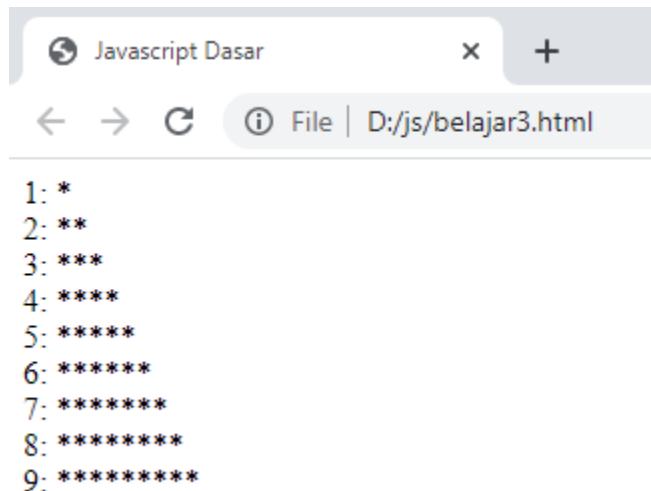
```

var ulang = confirm("apakah anda ingin mengulang?");
var counter = 0;

while (ulang) {
    counter++;
    var bintang = "*".repeat(counter) + "<br>";
    document.write(counter + ": " + bintang);
    ulang = confirm("apakah anda ingin mengulang?");
}

```

Outputnya :



```

1: *
2: **
3: ***
4: ****
5: *****
6: *****
7: *****
8: *****
9: *****

```

Materi tambahan dari modul 2

Kita akan mencoba mengambil nilai dari isian form HTML dengan menggunakan Javascript

Pada bagian ini yang harus kita perhatikan adalah atribut ID pada elemen HTML yang akan menjadi patokan atau trigger oleh javascript

Contoh pada program dibawah nilai ini saya menambahkan sebuah tombol dengan tag <button id="tombol_form">. Tombol ini akan digunakan sebagai 'trigger' untuk pengambilan nilai.

Jika di perhatikan, tag <form> memiliki atribut onsubmit="return false". Atribut ini adalah kode JavaScript yang berfungsi untuk menghentikan fitur standar form yang ketika tombol submit ditekan, form akan pindah ke halaman tertentu untuk di proses (biasanya diproses menggunakan PHP).

Dengan kata lain, menambahkan atribut onsubmit="return false" pada tag form, akan mematikan fungsi submit. Hal ini di lakukan karena kita akan menampilkan hasil form dengan JavaScript, bukan dengan PHP seperti biasanya, sehingga saya tidak butuh fungsi bawaan tersebut.

addEventListener adalah method (penyebutan untuk function di dalam pemrograman objek) yang digunakan untuk 'menempelkan' event kepada tag HTML.

Jika biasanya kita menggunakan event JavaScript: onClick langsung pada HTML, seperti: <button onclick="tampilkan()">, maka kita bisa menggantinya dengan addEventListener("click", tampilkan). Dengan menggunakan addEventListener, kode HTML bisa bebas dari JavaScript.

Akan tetapi metod addEventListener harus ditulis setelah kode HTML selesai dikirim ke web browser. Oleh karena itu, kita harus menempatkannya setelah kode HTML.

Contoh kode :

```
belajar2.html > html > body > h2 > span#hadiah
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="UTF-8">
5  <title>Tutorial Belajar JavaScript</title>
6  </head>
7  <body>
8  <form onsubmit="return false">
9    <label>Masukan Nilai: </label></form><br>
10   <input type="text" id="input_jumlah" value=""/><br>
11   <input type="text" id="input_harga" value=""/><br>
12   <button id="tombol_form">
13     Tampilkan Nilai pada jendela baru
14   </button>
15 </form>
16
17 <script>
18 document.getElementById("tombol_form").
19 addEventListener("click", tampilkan_nilai_form);
20
21 function tampilkan_nilai_form(){
22   var jumlah=document.getElementById("input_jumlah").value;
23   var harga=document.getElementById("input_harga").value;
24   var hasil=jumlah*harga;
25   document.getElementById("hasil").innerHTML=hasil;
26   if(hasil >= 10){
27     document.getElementById("hadiah").innerHTML="Selamat anda mendapatkan Hadiah";
28     return;
29   }
30   else{
31     document.getElementById("hadiah").innerHTML="";
32   }
33 }
34
35 </script>
36 <h2>Hasil: <span id="hasil"></span></h2>
37 <h2><span id="hadiah"></span></h2>
38 </body>
39 </html>
```

Masukan Nilai:

10

2

Tampilkan Nilai pada jendela baru

Hasil: 20

Selamat anda mendapatkan Hadiah