

# **Front End Bootcamp PT. Mahardika Solusi Teknologi**

Ariel Arliyanus, S.Kom

## **Chapter 8 – JavaScript - Loop & Array**

2021 © All Right Reserved

Dilarang memperbanyak dan/atau meng-copy sebagian atau seluruh material dalam dokumen ini tanpa persetujuan tertulis dari PT. Mahardika Solusi Teknologi

## Chapter 7

# JavaScript – Loop & Array

## Loop

Loop/Iteration adalah tindakan mengulang / merepetisi sebuah proses, dengan tujuan untuk mendapatkan deret hasil, atau dengan tujuan mendapatkan hasil tertentu dengan repetisi. Setiap proses repetisi ini disebut sebagai Iteration atau Looping.

Untuk melakukan looping/iteration, JavaScript menyediakan beberapa jenis iteration, yaitu:

- while-loop
- for-loop

## While-loop

While-loop adalah iterasi yang akan mengulang sebuah proses dengan sebuah kondisi tertentu.

Berikut adalah cara atau struktur penulisan while-loop:

```
while([Kondisi]) { // Kondisi yang menentukan apakah program akan melakukan iterasi.  
  // Berupa boolean atau true/false.  
  [Proses] // Merupakan proses yang akan dijalankan dalam satu iterasi  
}
```

Pada while-loop, statement **while** akan mengambil sebuah nilai **true** atau **false** dari hasil kondisi yang diberikan. Apabila statement **while** mendapatkan nilai **true**, maka proses yang berada didalam curly bracket (**{ }**) akan dijalankan. Looping akan terus dilakukan selama kondisi while-loop masih **true**.

Untuk mencegah looping berjalan terus-menerus, dilakukan sebuah proses yang akan mengubah kondisi yang sedemikian rupa yang bertujuan untuk menghentikan looping dengan menghasilkan kondisi yang **false**.

### Contoh Looping While-loop 1 Looping Angka 1-9 Sederhana

```
var flag = 1;  
while(flag < 10) { // Loop akan terus berjalan selama nilai flag masih dibawah 10  
  console.log('Iterasi ke-' + flag); // Menampilkan nilai flag pada iterasi tertentu  
  flag++; // Mengubah nilai flag dengan menambahkan 1  
}
```

## Contoh Looping While-loop 2 Looping Mengembalikan Angka Total

```
var deret = 5;
var jumlah = 0;
while(deret > 0) { // Loop akan terus berjalan selama nilai deret masih di atas 0
    jumlah += deret; // Menambahkan nilai variable jumlah dengan angka deret
    deret--; // Mengubah nilai deret dengan mengurangi 1
    console.log('Jumlah saat ini: ' + jumlah)
}

console.log(jumlah);
```

## For-loop

For-loop adalah bentuk lain dari iterasi, dimana statement **for** menjadi kontrol atas loop yang dilakukan. Hal ini yang menjadi pembeda antara for-loop dengan while-loop.

Berikut adalah cara atau struktur penulisan for-loop:

```
for([Inisialisasi], [Kondisi], [Incremental/Decremental]) {
    [Proses] // Merupakan proses yang akan dijalankan dalam satu iterasi
}
```

Pada for-loop, statement **for** akan menampung tiga parameter, yaitu sebut saja inisialisasi, kondisi, dan incremental/decremental. Ketiga parameter ini akan menjadi kontrol kapan loop ini harus berhenti. Pada parameter pertama, yaitu inisialisasi, sebuah variable diberikan nilai awal atau default. Pada parameter kedua, yaitu kondisi, for-loop akan terus berjalan selama kondisi ini masih terpenuhi, dengan kata lain, mengandung nilai **true**. Pada parameter kedua, yaitu incremental/decremental, variabel yang menjadi kontrol terhadap loop ini akan diubah nilainya.

*Best Practice:* Walaupun memang for-loop dapat mengubah kondisi di dalam proses, namun best practice dari penggunaan for-loop adalah seluruh kendali atau kontrol dari looping ditentukan oleh variable yang diinisialisasi, di increment/decrement, dan juga kondisi for-loop pun menggunakan variable tersebut.

Untuk memudahkan kamu mendapatkan gambaran jelas tentang penggunaan for-loop, mari kita gunakan kedua contoh while-loop dan kita tulis ulang dalam bentuk for-loop.

## Contoh Looping For-loop 1 Looping Angka 1-9 Sederhana

```
for(var angka = 1; angka < 10; angka++) {
    console.log('Iterasi ke-' + angka);
}
```

## Contoh Looping For-loop 2 Looping Mengembalikan Angka Total

```
var jumlah = 0;
for(var deret = 5; deret > 0; deret--) {
    jumlah += deret;
    console.log('Jumlah saat ini: ' + jumlah);
}

console.log('Jumlah terakhir: ' + jumlah);
```

### Contoh Looping For-loop 3 Looping Dengan Increment dan Decrement Lebih dari 1

```
for(var deret = 0; deret < 10; deret += 2) {  
  console.log('Iterasi dengan Increment counter 2: ' + deret);  
}  
  
console.log('-----');  
  
for(var deret = 15; deret > 0; deret -= 3) {  
  console.log('Iterasi dengan Decrement counter : ' + deret);  
}
```

### Waspada! Infinite Looping!

Dengan sengaja atau tidak sengaja, kode kamu mungkin dapat menghasilkan infinite looping, atau looping yang tidak akan pernah berhenti. Bila ini terjadi, segera periksa statement kondisi kamu.

```
var flag = 1;  
while(flag < 10) { // Loop akan terus berjalan, karena nilai flag tidak pernah berubah  
  console.log('Iterasi ke-' + flag);  
}
```

# Array

Array adalah kumpulan atau tumpukan berbagai data. Cara menuliskan array yaitu dengan kurung siku (`[]`) dan elemen-elemen nya dipisah menggunakan tanda koma (`,`). Setiap elemen dari array memiliki indeks yang dimulai dari 0, 1, 2, dst. Kita dapat memanipulasi array dengan berbagai cara seperti menambahkan dan mengeluarkan elemen dalam array, menggabungkan array, atau bahkan menghapus seluruh elemen Array nya. Kita bisa memasukkan beberapa tipe data yang berbeda ke dalam Array bahkan memasukkan Array ke dalam Array.

Array juga memiliki property `.length` seperti pada string yang berarti panjang dari sebuah array.

```
var hobbies = ["coding", "cycling", "climbing", "skateboarding"]
console.log(hobbies) // [ 'coding', 'cycling', 'climbing', 'skateboarding' ]
console.log(hobbies.length) // 4

console.log(hobbies[0]) // coding
console.log(hobbies[2]) // climbing
// Mengakses elemen terakhir dari array
console.log(hobbies[hobbies.length - 1]) // skateboarding
```

## Metode Array

Berikut ini adalah beberapa metode atau built-in functions yang dimiliki oleh tipe data array:

- `push`: menambah 1 nilai ke array ke index paling belakang
- `pop`: menghapus 1 nilai dari array index paling belakang
- `unshift`: menambah 1 nilai ke array index paling depan (index 0)
- `shift`: menghapus 1 nilai dari array index paling depan (index 0)
- `join`: menggabungkan seluruh element array menjadi sebuah string dan mengambil parameter sebagai simbol penyambung antar elemen
- `sort`: mengurutkan elemen di dalam array sesuai alphabet
- `slice`: mengambil beberapa lapis data
- `splice`: mengubah nilai array dengan menghapus dan/atau menambah nilai baru ke array
- `split`: memecah string dan mengembalikan array sesuai dengan separator / pemisah yang didefinisikan

Cara menggunakan metode array adalah dengan menggunakan tanda dot (`.`) contohnya sbb:

```
var feeling = ["dag", "dig"]
feeling.push("dug") // menambahkan nilai "dug" ke index paling belakang
feeling.pop() // menghapus nilai pada elemen terakhir array
```

Beberapa metode array ada yang mengembalikan nilai array lama yang sudah diperbarui (array lama di-update) dengan metode tersebut, dan ada pula metode array yang mengembalikan sebuah array baru dan tidak mengubah data pada array yang lama (array baru yang serupa dengan array lama). Sifat seperti ini disebut mutable atau immutable.

### `.push()`

*Push* adalah metode array untuk menambahkan nilai di belakang elemen terakhir di array. metode `push` menerima sebuah parameter yaitu nilai yang ingin kita tambahkan ke dalam array.

```
var numbers = [0, 1, 2]
```

```
numbers.push(3)
console.log(numbers) // [0, 1, 2, 3]
// Bisa juga memasukkan lebih dari satu nilai menggunakan metode push
numbers.push(4, 5)
console.log(numbers) // [0, 1, 2, 3, 4, 5]
```

### **.pop()**

Pop adalah kebalikan dari push yaitu menghapus nilai elemen paling terakhir dari sebuah array. metode Pop tidak menerima parameter apapun sehingga metode pop hanya bisa mengeluarkan satu elemen saja yaitu yang paling terakhir dari sebuah array.

```
var numbers = [0, 1, 2, 3, 4, 5]
numbers.pop()
console.log(numbers) // [0, 1, 2, 3, 4]
```

### **.unshift()**

Unshift yaitu menambahkan nilai pada index ke-0 sehingga elemen-elemen sebelumnya bergeser.

```
var numbers = [0, 1, 2, 3]
numbers.unshift(-1)
console.log(numbers) // [-1, 0, 1, 2, 3]
```

### **.shift()**

Shift kebalikan dari unshift yaitu menghapus nilai pada elemen terdepan dari sebuah array. metode Shift tidak menerima parameter apapun.

```
var numbers = [0, 1, 2, 3]
numbers.shift()
console.log(numbers) // [1, 2, 3]
```

### **.sort()**

Sort adalah metode untuk mengurutkan nilai pada array. Secara otomatis, sort akan mengurutkan secara ascending (dari rendah ke tinggi) dan diurutkan berdasarkan unicode dari karakter. Urutan unicode artinya ada karakter yang secara nilai lebih besar dibandingkan dengan karakter yang lainnya. contohnya adalah karakter "b" akan lebih besar daripada "a", karakter "c" lebih besar daripada karakter "b", dst.

```
var animals = ["kera", "gajah", "musang"]
animals.sort()
console.log(animals) // ["gajah", "kera", "musang"]
```

### **.slice()**

slice adalah metode untuk mengambil irisan dari sebuah array. Metode slice bisa menerima satu atau dua parameter. Parameter pertama adalah nomer index pertama yang akan kita ambil sebagai irisan, sedangkan parameter kedua adalah nomer index terakhir yang ingin kita ambil sebagai irisan.

```
var angka = [0, 1, 2, 3]
var irisan1 = angka.slice(1,3)
console.log(irisan1) // [1, 2, 3]
var irisan2 = angka.slice(0,2)
console.log(irisan2) // [0, 1, 2]
```

Jika parameter kedua tidak diisi maka secara otomatis slice akan mengiris array dari indeks di paramer pertama sampai ke indeks terakhir array tersebut.

```
var angka = [0, 1, 2, 3]
var irisan3 = angka.slice(2)
console.log(irisan3) // [2, 3]
```

Jika diperhatikan, metode slice tidak mengubah Array yang awal dan hasil irisan array dapat kita tampung kembali ke variable yang baru. Oleh karena itu metode Slice juga berfungsi untuk

membuat salinan atau “shallow copy” dari sebuah array. Cara membuat shallow copy dengan metode slice yaitu dengan tanpa memberikan parameter apapun pada metode slice.

```
var angka = [0, 1, 2, 3]
var salinAngka = angka.slice()
console.log(salinAngka) // [0, 1, 2, 3]
```

## **.splice()**

Splice yaitu metode untuk menghapus dan/atau menambahkan nilai elemen pada array. Metode splice bisa menerima parameter sebanyak dua atau lebih parameter. Jika ingin menggunakan splice untuk menghapus elemen pada index tertentu maka digunakan 2 parameter. Jika ingin menggunakan splice untuk menambahkan elemen pada index tertentu maka digunakan tiga parameter.

```
array.splice([IndexMulai], [JumlahNilaiYangDihapus], [NilaiYangDitambahkan1], [NilaiYangDitambahkan2], ...);
```

Berikut contoh penggunaan metode **splice()**

```
var fruits = [ "banana", "orange", "grape"]
fruits.splice(1, 0, "watermelon")
console.log(fruits) // [ "banana", "watermelon", "orange", "grape"]
```

Contoh di atas berarti kita mulai mengubah array fruits dari indeks no 1, dan melakukan penghapusan nilai sebanyak 0 (yang berarti tidak menghapus apapun), dan menambahkan nilai “watermelon” pada indeks tersebut.

```
var fruits = [ "banana", "orange", "grape"]
fruits.splice(0, 2)
console.log(fruits) // ["grape"]
```

Contoh di atas berarti kita mulai mengubah array fruits dari indeks ke-0 dan melakukan penghapusan data sebanyak 2 elemen.

## **.split() dan .join()**

Metode split yaitu memecah sebuah string sehingga menjadi sebuah array. Split menerima sebuah parameter berupa karakter yang menjadi separator untuk memecah string.

```
var biodata = "name:john,doe"
var name = biodata.split(":")
console.log(name) // [ "name", "john,doe"]
```

Metode join yaitu kebalikan dari split yaitu mengubah sebuah array menjadi string dengan cara menggabungkan (join) seluruh elemen array menjadi satu dengan sebuah karakter pemisah yang dikirim melalui parameter. contohnya sebagai berikut:

```
var title = ["my", "first", "experience", "as", "programmer"]
var slug = title.join("-")
console.log(slug) // "my-first-experience-as-programmer"
```

# Tugas

## Soal 1

Pada tugas ini kamu diminta untuk melakukan looping dalam JavaScript dengan menggunakan syntax `while`. Untuk membuat tantangan ini lebih menarik, kamu juga diminta untuk membuat suatu looping yang menghitung maju dan menghitung mundur. Jangan lupa tampilkan di console juga judul 'LOOPING PERTAMA' dan 'LOOPING KEDUA'."

Output:

LOOPING PERTAMA

2 - I love coding  
4 - I love coding  
6 - I love coding  
8 - I love coding  
10 - I love coding  
12 - I love coding  
14 - I love coding  
16 - I love coding  
18 - I love coding  
20 - I love coding

LOOPING KEDUA

20 - I will become a frontend developer  
18 - I will become a frontend developer  
16 - I will become a frontend developer  
14 - I will become a frontend developer  
12 - I will become a frontend developer  
10 - I will become a frontend developer  
8 - I will become a frontend developer  
6 - I will become a frontend developer  
4 - I will become a frontend developer  
2 - I will become a frontend developer



## Soal 2

Pada tugas ini kamu diminta untuk melakukan looping dalam JavaScript dengan menggunakan syntax **for**. Untuk membuat tantangan ini lebih menarik, kamu juga diminta untuk memenuhi syarat tertentu yaitu:

SYARAT:

- A. Jika angka ganjil maka tampilkan Santai
- B. Jika angka genap maka tampilkan Berkualitas
- C. Jika angka yang sedang ditampilkan adalah kelipatan 3 DAN angka ganjil maka tampilkan I Love Coding.

Output:

```
1 - Santai
2 - Berkualitas
3 - I Love Coding
4 - Berkualitas
5 - Santai
6 - Berkualitas
7 - Santai
8 - Berkualitas
9 - I Love Coding
10 - Berkualitas
11 - Santai
12 - Berkualitas
13 - Santai
14 - Berkualitas
15 - I Love Coding
16 - Berkualitas
17 - Santai
18 - Berkualitas
19 - Santai
20 - Berkualitas
```

## soal 3

Kali ini kamu diminta untuk menampilkan sebuah segitiga dengan tanda pagar (#) dengan dimensi tinggi 7 dan alas 7. Looping boleh menggunakan syntax apa pun (**while**, **for**, **do while**).

Output:

```
#
##
###
####
#####
#####
#####
```

## Soal 4

**Buatlah variabel seperti di bawah ini**

```
var kalimat=["aku", "saya", "sangat", "sangat", "senang", "belajar",  
"javascript"]
```

ubah kalimat diatas menjadi seperti di bawah ini:

```
["saya", "sangat", "senang", "belajar", "javascript"]
```

lalu tampilkan dengan output seperti di bawah ini:

```
"saya sangat senang belajar javascript"
```

## Soal 5

buatlah variabel seperti di bawah ini

```
var sayuran=[]
```

tambahkan data di bawah ini ke variabel sayuran:

- Kangkung
- Bayam
- Buncis
- Kubis
- Timun
- Seledri
- Tauge

lalu urutkan berdasarkan alfabet dan lalu tampilkan dengan loop dan beri angka di depannya sehingga menghasilkan output seperti ini:

1. Bayam
2. Buncis
3. Kangkung
4. Kubis
5. Seledri
6. Tauge
7. Timun

Agar soal mudah di periksa trainer maka sebelum jawaban soal sisipkan komentar seperti di bawah ini

```
// soal 1  
.... jawaban soal 1
```

```
// soal 2  
.... jawaban soal 2
```

dan seterusnya

**Update tugas masing-masing di repository yang sudah dibuat.**