

Laporan For Loop Pada Javascript

1. Pendahuluan

Materi ini membahas berbagai metode perulangan dan control statement dalam JavaScript. Tujuannya adalah untuk memahami bagaimana cara kerja berbagai jenis perulangan serta bagaimana cara mengontrol eksekusi perulangan tersebut menggunakan statement tertentu.

2. Jenis-Jenis Perulangan di JavaScript

a. For

For adalah cara looping yang paling umum dilakukan di JavaScript. Semenjak kehadiran ES6, for terbagi lagi menjadi beberapa jenis. Saat ini, for terdiri dari for loop, for in, for of. Simak penjelasan berikut:

b. For Loop

Pertama, kita akan bahas for loop. Struktur dari for loop tampak seperti berikut ini.

```
for (variabel awal; test kondisi; increment) {  
    // do something  
}
```

Variabel awal adalah nilai variabel sebelum looping dilakukan. Test kondisi adalah evaluasi dari looping, jika bernilai false, looping akan berhenti. Increment adalah nilai variabel yang bertambah setiap looping dilakukan. Perhatikan contoh berikut ini.

```
for (let i = 0; i < 5; i++) {  
    console.log(`Angka ke-${i} adalah ${i}`);  
}
```

```
Angka ke-0 adalah 0
Angka ke-1 adalah 1
Angka ke-2 adalah 2
Angka ke-3 adalah 3
Angka ke-4 adalah 4
```

Pada contoh di atas, nol merupakan nilai awal sebelum looping dijalankan. Pengecekan kondisi `i < 5` sebelum looping, ketika bernilai false, looping akan terhenti. Terakhir, `i++` artinya adalah nilai `i` akan bertambah setiap looping dilakukan.

c. For In

For in banyak digunakan untuk pengulangan pada object karena ia dapat melakukan iterasi ke seluruh data di dalam objek. Bahkan, ia juga dapat melakukan iterasi ke properti inheritance dari object seperti `length`. Berikut contoh penggunaan for in.

```
const person = { name: 'Fulan', origin: 'Bandung',
  birthYear: 2024 };
```

```
for (const property in person) {
  console.log(`${property} bernilai
  ${person[property]}`);
}
```

```
name bernilai Fulan
origin bernilai Bandung
birthYear bernilai 2024
```

Sintaks dari for in tidak berbeda jauh dengan for loop. Dengan menggunakan for in kita tidak perlu menuliskan nilai variabel awal, test kondisi dan

Formatted: Centered

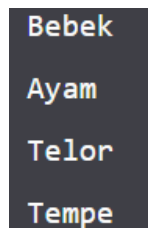
increment. For in akan melakukan perulangan item yang ada di dalam object (pada contoh object person).

d. For Of

Kehadiran for of dimulai pada ECMAScript 2015 (ES6). For of berbeda dengan for in. For of lebih sederhana karena kita tidak perlu memikirkan property dan key. Perhatikan contoh berikut.

```
const names = ['Bebek', 'Ayam', 'Telor', 'Tempe'];
```

```
for (const item of names) {  
  console.log(item);  
}
```



Bebek
Ayam
Telor
Tempe

Dengan for of datanya bisa kita dapatkan langsung tanpa menambahkan indeks atau nama propertinya seperti contoh di atas. Sintaksnya lebih ringkas karena kita hanya perlu memanggil variabel item.

e. While

Perulangan di JavaScript tak hanya menggunakan for, tetapi ada cara lainnya yaitu while statement. While statement akan mengeksekusi statement ketika kondisinya bernilai truthy. Strukturnya:

```
while (condition)
```

```
  statement
```

Contohnya:

```
let i = 0;
```

```
while (i < 5) {
```

```
  console.log(`Angka ke-${i} adalah ${i}.`);
```

```
i++;  
}
```

```
Angka ke-0 adalah 0.  
Angka ke-1 adalah 1.  
Angka ke-2 adalah 2.  
Angka ke-3 adalah 3.  
Angka ke-4 adalah 4.
```

While hanya peduli dengan kondisi yang kita berikan. Jika kondisi tidak diubah, bisa terjadi **infinite loop**, contohnya:

```
let i = 0;  
  
while (i < 5) {  
  console.log(`Angka ke-${i} adalah ${i}.`);  
}
```

f. Do-While

While dan do-while tidak berbeda jauh. Perbedaannya hanya pada urutan pengecekan kondisi. Contohnya:

```
do {  
  // Do something  
} while (condition)
```

Contohnya:

```
let i = 0;  
  
do {  
  console.log(`Angka ke-${i} adalah ${i}.`);  
  i++;  
} while (i < 5);
```

```
Angka ke-0 adalah 0.  
Angka ke-1 adalah 1.  
Angka ke-2 adalah 2.  
Angka ke-3 adalah 3.  
Angka ke-4 adalah 4.
```

Karena kondisi dicek di akhir, maka isi do akan selalu dijalankan minimal satu kali.

3. Control Statement

Control statement berfungsi untuk menghentikan atau mengontrol eksekusi kode saat perulangan dilakukan.

a. Break

Break statement menghentikan eksekusi kode dalam perulangan atau percabangan. Contoh penggunaan pada loop:

```
for (let i = 0; i < 10; i++) {  
  if (i === 5) {  
    break;  
  }  
  
  console.log(i);  
}
```

```
0  
1  
2  
3  
4
```

Contoh penggunaan pada switch:

```
const number = 1;
```

```
switch (number) {  
  case 1:  
    console.log('Ini 1');  
    break;  
  case 2:  
    console.log('Ini 2');  
    break;  
  case 3:  
    console.log('Ini 3');  
    break;  
  default:  
    console.log('Ini default');  
}
```

Ini 1

b. Continue

Continue statement melewati satu iterasi dalam loop dan langsung melanjutkan ke iterasi berikutnya. Contohnya:

```
for (let i = 0; i < 10; i++) {  
  if (i === 5) {  
    continue;  
  }  
  console.log(i);  
}
```

0
1
2
3
4
6
7
8
9

Dalam contoh tersebut, angka 5 tidak akan dicetak karena continue akan langsung melanjutkan ke iterasi berikutnya.