

Looping

4 MARET 2024

1. Perulangan dengan `while`

Perulangan `while` akan terus menjalankan kode di dalam blok perulangan selama kondisi yang diberikan bernilai benar (true).

```
let a = 1;
while (a <= 10) {
  if (a % 2 !== 0) {
    console.log(a);
  }
  a++;
}
```

- Kode ini menggunakan perulangan `while` untuk mengulang dari angka 1 hingga 10.
- Setiap iterasi, program memeriksa apakah nilai `a` adalah angka ganjil dengan menggunakan operator modulus (%). Jika sisa pembagian `a` dengan 2 tidak sama dengan 0, maka itu adalah angka ganjil.
- Jika `a` adalah angka ganjil, maka angka tersebut dicetak ke konsol.
- Setelah itu, nilai `a` ditambah satu untuk mengarahkan perulangan ke angka berikutnya.

2. Perulangan dengan `do-while`

Perulangan ``do-while`` hampir sama dengan ``while``, kecuali bahwa ``do-while`` akan menjalankan blok kode setidaknya satu kali, bahkan jika kondisinya salah.

```
let b = 10;
do {
  console.log(b);
  b--;
} while (b >= 1);
```

- Kode ini menggunakan perulangan `do-while` untuk mengurangi bilangan dari 10 hingga 1.
- Pertama-tama, program mencetak nilai `b` ke konsol.
- Kemudian, nilai `b` dikurangi satu setiap iterasi.
- Perulangan akan berlanjut selama nilai `b` lebih besar dari atau sama dengan 1.

3. Perulangan dengan `for`

Perulangan ``for`` digunakan ketika kita tahu berapa kali kita ingin menjalankan blok kode.

```
for (let c = 5; c <= 10; c++) {
  for (let d = 1; d <= 10; d++) {
    console.log(c + " x " + d + " = " + (c * d));
  }
}
```

- Kode ini menggunakan perulangan `for` untuk mencetak tabel perkalian dari angka 5 hingga 10.
- Terdapat dua perulangan `for` bersarang di sini. Perulangan luar untuk mengontrol angka mana yang akan dikalikan, sedangkan perulangan dalam untuk mengontrol angka mana yang akan dikalikan dengan angka dari perulangan luar.
- Pada setiap iterasi perulangan dalam, hasil perkalian dari angka dari perulangan luar (`c`) dan angka dari perulangan dalam (`d`) dicetak ke konsol.

4. Perulangan dengan `for-in`

Perulangan `for-in` digunakan untuk mengulang melalui properti dari sebuah objek.

```
const car = {  
  brand: "Toyota",  
  model: "Camry",  
  year: 2022  
};  
  
for (let prop in car) {  
  console.log(prop + ": " + car[prop]);  
}
```

- Kode ini menggunakan perulangan `for-in` untuk mengulang melalui properti-properti objek `car`.
- Setiap properti dari objek `car` diakses menggunakan sintaks `car[prop]` dan dicetak ke konsol bersama dengan namanya.

5. Perulangan dengan `for-of`

Perulangan `for-of` digunakan untuk mengulang melalui nilai-nilai dari objek yang dapat diulang (seperti Array, Map, Set, dll).

```
const str = "JavaScript";  
for (let char of str) {  
  console.log(char);  
}
```

- Kode ini menggunakan perulangan `for-of` untuk mengulang melalui setiap karakter dalam string `"JavaScript"`.
- Setiap karakter dari string diambil satu per satu dan dicetak ke konsol.

Studi Kasus:

Mari kita buat studi kasus untuk mencetak semua angka genap dari 1 hingga 10 menggunakan masing-masing jenis perulangan:

```
// Perulangan while
let a = 1;
while (a <= 10) {
  if (a % 2 === 0) {
    console.log(a);
  }
  a++;
}

// Perulangan do-while
let b = 1;
do {
  if (b % 2 === 0) {
    console.log(b);
  }
  b++;
} while (b <= 10);

// Perulangan for
for (let c = 1; c <= 10; c++) {
  if (c % 2 === 0) {
    console.log(c);
  }
}

// Perulangan for-in (tidak cocok untuk kasus ini)

// Perulangan for-of (tidak cocok untuk kasus ini)
```

1. **Perulangan `while`**: Kita menggunakan variabel `a` yang diinisialisasi dengan nilai 1. Selama nilai `a` kurang dari atau sama dengan 10, kita memeriksa apakah `a` adalah angka genap. Jika ya, maka kita mencetaknya. Setiap iterasi, nilai `a` ditambah satu.
2. **Perulangan `do-while`**: Kita menggunakan variabel `b` yang diinisialisasi dengan nilai 10. Dalam blok `do`, kita mencetak nilai `b` dan kemudian mengurangnya. Setelah itu, kita memeriksa apakah nilai `b` masih lebih besar dari atau sama dengan 1. Jika ya, kita melanjutkan perulangan.
3. **Perulangan `for`**: Kita menggunakan perulangan `for` dengan inisialisasi `c` di 5, dan akan dijalankan selama `c` kurang dari atau sama dengan 10. Pada setiap iterasi, kita memeriksa apakah nilai `c` adalah angka genap. Jika ya, kita mencetaknya.
4. **Perulangan `for-in`**: Tidak digunakan dalam kasus ini karena kita tidak bekerja dengan objek.
5. **Perulangan `for-of`**: Tidak digunakan dalam kasus ini karena kita tidak bekerja dengan array atau objek yang dapat diulang.

KESIMPULAN DAN PERBEDAAN ANTARA JENIS PERULANGAN DALAM JAVASCRIPT

1. Perulangan ``while``:

- Perulangan ``while`` akan terus menjalankan blok kode selama kondisi yang diberikan bernilai benar (true).
- Kondisi dievaluasi sebelum eksekusi blok kode, sehingga ada kemungkinan blok kode tidak pernah dijalankan jika kondisi awalnya salah.

2. Perulangan ``do-while``:

- Perulangan ``do-while`` hampir sama dengan ``while``, tetapi blok kode di dalamnya akan dijalankan setidaknya sekali, bahkan jika kondisinya salah.
- Kondisi dievaluasi setelah eksekusi blok kode, sehingga setidaknya satu iterasi akan dilakukan.

3. Perulangan `for`:

- Perulangan `for` digunakan ketika kita tahu berapa kali kita ingin menjalankan blok kode.
- Terdiri dari tiga bagian: inisialisasi, kondisi, dan ekspresi peningkatan.
- Cocok digunakan untuk perulangan dengan jumlah iterasi yang sudah diketahui sebelumnya.

4. Perulangan `for-in`:

- Perulangan `for-in` digunakan untuk mengulang melalui properti dari sebuah objek.
- Tidak cocok untuk mengulang melalui array, karena urutan iterasi tidak dijamin.

5. Perulangan `for-of`:

- Perulangan `for-of` digunakan untuk mengulang melalui nilai-nilai dari objek yang dapat diulang (seperti Array, Map, Set, dll).
- Cocok untuk mengulang melalui elemen-elemen dalam array atau nilai-nilai dalam objek yang dapat diiterasi.

Perbedaan Utama:

- Perulangan `while` dan `do-while` mengevaluasi kondisi sebelum menjalankan blok kode, sementara `for` mengevaluasi kondisi di awal sebelum setiap iterasi.
- Perulangan `do-while` pasti menjalankan blok kode setidaknya sekali, sementara `while` tidak menjamin hal tersebut.
- Perulangan `for` memiliki struktur khusus yang terdiri dari tiga bagian: inisialisasi, kondisi, dan ekspresi peningkatan.
- Perulangan `for-in` digunakan untuk mengulang melalui properti-properti objek, sedangkan `for-of` digunakan untuk mengulang melalui nilai-nilai objek yang dapat diiterasi.
- Perulangan `for-in` tidak menjamin urutan iterasi, sementara `for-of` akan mengulangi nilai-nilai dalam urutan yang dijamin oleh objek yang diiterasi.

Pemilihan jenis perulangan tergantung pada situasi dan struktur data yang digunakan dalam program, serta kebutuhan spesifik dari algoritma yang dibangun.

Latihan Praktek:

1. Buatlah sebuah perulangan `while` yang mencetak angka ganjil dari 1 hingga 10!
2. Gunakan perulangan apa pun untuk mencetak hasil dari pengurangan bilangan 10 hingga 1!
3. Buatlah sebuah perulangan yang mencetak angka 1 – 100!

Hint :

1.

```
if (1 % 2 != 0)
```

2.

Langkah 1: Tetapkan nilai awal bilangan (deklarasi variable)

Langkah 2: Buat perulangan apa pun

Langkah 3: Tampilkan nilai bilangan

Langkah 4: Kurangi nilai bilangan

Langkah 5: Periksa kondisi untuk keluar dari loop

3.

Langkah 1. Mulailah dengan menetapkan variabel awal dengan nilai 1.

Langkah 2. Gunakan loop untuk mengiterasi dari nilai awal hingga 100

Langkah 3. Di setiap iterasi, cetak nilai iterasi saat ini. 4. Perbarui nilai awal setiap iterasi.