**Tugas 3**

**Perulangan (Loop) di Javascript**

Nama : Ryan Febriansyah

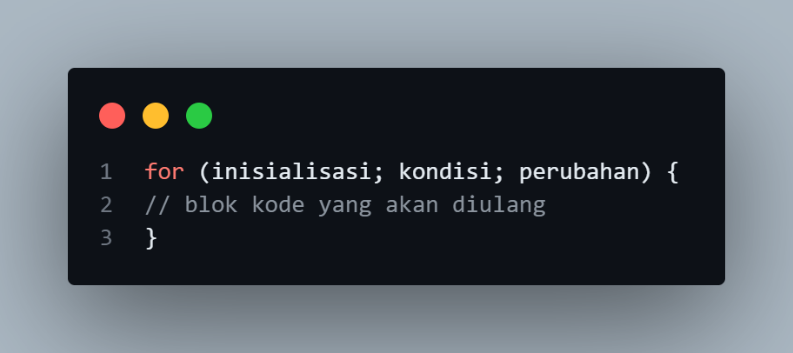
Program : Project Based Learning Tahap 2

Kejuruan : Pengembangan Web dengan Node js dan React

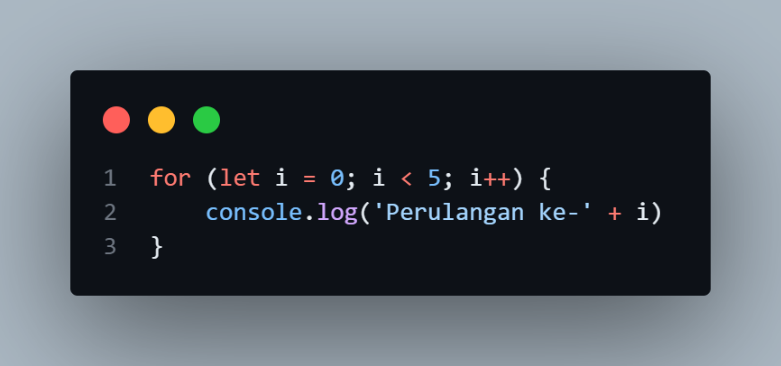
1. **Perulangan for**

Digunakan untuk mengeksekusi sekumpulan perintah dalam jumlah yang sudah ditentukan.

**Struktur Dasar**

****

**Contoh**

****

**Variasi**

* Perulangan mundur: for (let i = 10; i > 0; i--)
* Langkah khusus: for (let i = 0; i <= 10; i += 2)
* break: menghentikan perulangan lebih awal.
* continue: melewati iterasi tertentu dan melanjutkan ke iterasi berikutnya

1. **Perulangan while**

Digunakan jika jumlah iterasi belum pasti, tetapi bergantung pada kondisi.

**Struktur Dasar**



**Contoh**

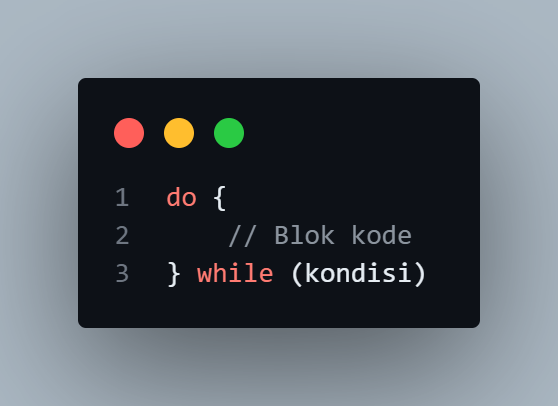
****

*Karakterisik: kondisi diperiksa sebelum perulangan dijalankan.*

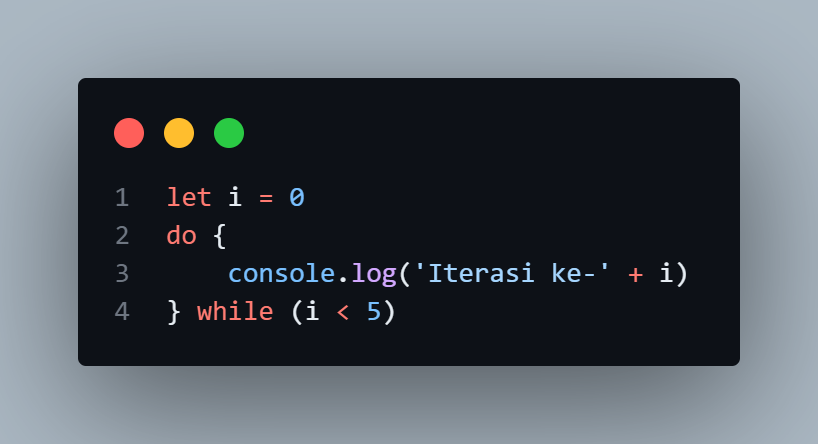
1. **Perulangan do … while**

Mirip dengan while, tetapi dijalankan minimal satu kali meskipun kondisi bernilai false.

**Struktur Data**

****

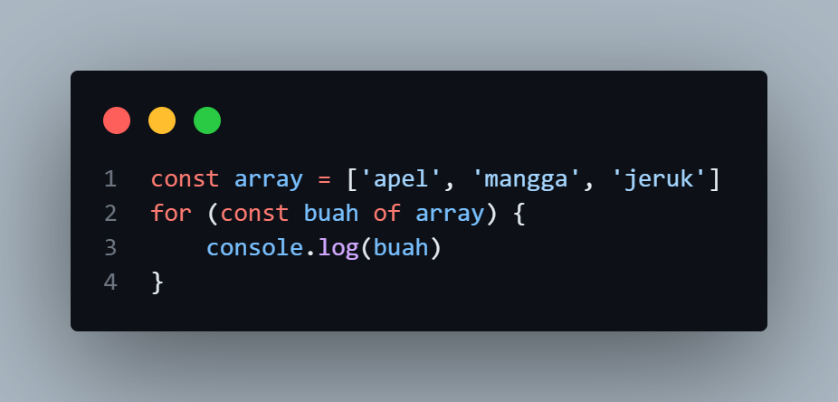
**Contoh**

****

1. **for … of**

Digunakan untuk mengiterasi nilai (array, string, Map, Set, dsb).

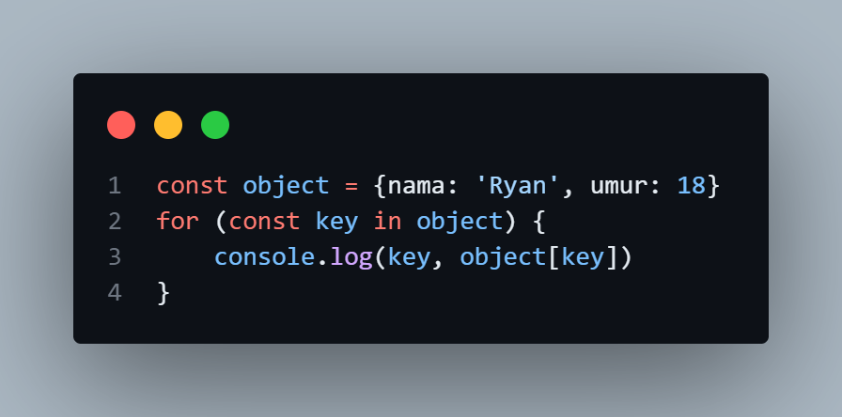
**Contoh**

****

1. **for … in**

Digunakan untuk mengiterasi property/ key dari sebuah objek.

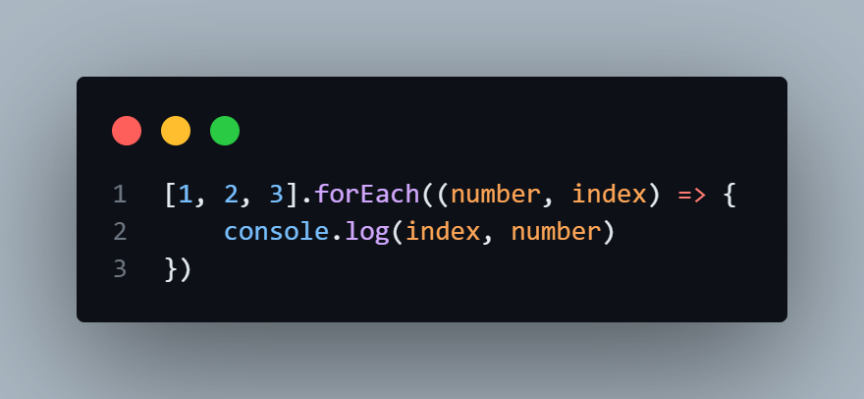
**Contoh**

****

1. **Array.prototype.forEach**

Metode bawaan array untuk melakukan iterasi pada setiap elemen.

**Contoh**



*Catatan: forEach tidak mendukung penggunaan break atau continue secara langsung.*

1. **Perulangan Bersarang (Nested Loop)**

Digunakan ketika terdapat perulangan di dalam perulangan.

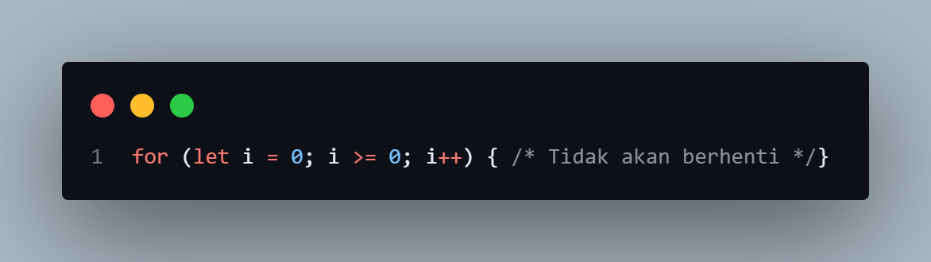
**Contoh**



1. **Infinite Loop**

Perulangan yang tidak pernah berhenti karena kondisi selalu bernilai true. Perlu dihindari kecuali memang diperlukan secara khusus.

**Contoh**

****

1. **Hal yang Perlu Diperhatikan**

* Pemilihan jenis loop: gunakan for jika jumlah iterasi diketahui, gunakan while atau do … while jika tidak diketahui.
* var vs let: gunakan let untuk menghindari masalah scope.
* Kinerja: nested loop dapat menurunkan performa pada data besar.