Pembaruan! Modul ini dibuat pada tanggal 6 December 2019. Pembaruan terakhir adalah: **Konversi potongan kode menjadi kode interaktif.**.
Lihat riwayat »

Variable Scope

Sejauh ini kita sudah mengenal function . Ada satu hal lagi yang harus kita tahu mengenai dasar JavaScript, yakni *scoping variable*. Ada banyak keadaan di mana kita membutuhkan variabel untuk diakses di seluruh *script* yang kita buat. Tetapi ada juga keadaan di mana kita ingin variabel tersebut hanya dapat diakses pada cakupan fungsi dan fungsi turunannya saja.

Variabel yang dapat di akses dari seluruh script disebut dengan "*globally scoped*," sementara variabel yang hanya dapat diakses hanya pada *function* tertentu disebut dengan "*locally scoped*."

Variabel JavaScript menggunakan fungsi untuk mengelola cakupannya, Jika variabel didefinisikan di luar fungsi, maka variabel akan bersifat *global*. Jika variabel didefinisikan di dalam fungsi, maka variabel bersifat lokal dan cakupannya hanya pada fungsi tersebut atau turunannya.

Untuk lebih jelasnya, berikut variabel yang dapat diakses dalam sebuah fungsi:

- Variabel argumen dari fungsinya.
- Lokal variabel yang didefinisikan pada fungsinya.
- Variabel dari induk fungsinya.
- Global variabel.

```
1.  // global variable, dapat diakses pada parent() dan child()
2.  const a = 'a';
3.
4.  function parent() {
5.      // local variable, dapat diakses pada parent() dan child(), tetapi tidak dapat diakses diluar dari fungsi tersebut.
6.      const b = 'b';
7.
8.      function child() {
9.          // local varible, dapat diakses hanya pada fungsi child().
10.         const c = 'c';
11.     }
12. }
```

Kita harus berhati-hati dalam mendefinisikan variabel di dalam fungsi. Pasalnya, kita bisa mendapatkan hasil yang tidak diperkirakan, contohnya seperti berikut:

main.js +

```
1 function multiply(num) {
2     total = num * num;
3     return total;
4 }
```

```
8
9  console.log(total)
10
11 /* output
12 400
13 */
```

INPUT RESET JALANKAN

Output:

Mungkin kita berharap nilai total akan tetap 9. Mengingat variabel total pada fungsi `multiply`, seharusnya tidak akan berpengaruh untuk kode di luar dari fungsi tersebut. Hal ini bisa terjadi karena pada fungsi `multiply()` kita tidak menetapkan variabel `total` sebagai cakupan lokal, kita tidak menggunakan keyword `const`, `let`, atau `var` ketika mendeklarasikan variabel `total` pada fungsi `multiply()` sehingga variabel `total` menjadi global.

> Perlu kita perhatikan bahwa, ketika kita lupa menuliskan keyword let, const atau var pada script ketika membuat sebuah variabel, maka variabel tersebut akan menjadi global.

Sebisa mungkin kita harus menghindari pembuatan variabel global, karena variabel global dapat diakses pada seluruh script yang kita tuliskan. Semakin banyak variabel global yang kita tuliskan, semakin tinggi kemungkinan variabel tabrakan (collision) terjadi.

← KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA →