



DISKUSIKAN MATERI

LAPORKAN MATERI

## Daftar Modul

Masukkan kata kunci

Persetujuan Hak Cipta ✓

Modul 1: Introduction Course (Gratis) ✓

Prasyarat Kelas (Gratis) ✓

Apa yang Akan Kita Pelajari? (Gratis) ✓

Tools Requirement ✓

Modul 2: ECMAScript 6 (Gratis) ✓

Running Snippet Code (Gratis) ✓

Persiapan Project Latihan (Gratis) ✓

## Function Parameter

Pada penggunaan arrow function kita melihat bahwa variabel yang diletakan sebelum tanda panah ( $\Rightarrow$ ) adalah merupakan parameter dari fungsi.

```
1. // name merupakan parameter dari fungsi
2. const sayName = name => console.log(`Nama saya ${name}`);
```

Namun penulisan tersebut hanya berlaku jika fungsi memiliki satu parameter saja. Lantas bagaimana jika kita ingin membuat fungsi yang memiliki lebih dari satu parameter atau tanpa sebuah parameter? Hal itu sangat biasa dilakukan bukan?

Pada arrow function jika terdapat dua atau lebih parameter fungsi kita perlu membungkusnya dengan tanda kurung seperti ini:

```
1. const sayHello = (name, greet) => console.log(`${greet}, ${name}!`);
2.
3. sayHello("Dimas", "Selamat Pagi")
4.
5. /* output:
6. Selamat Pagi, Dimas!
7. */
```

Namun jika kita sama sekali tidak membutuhkan parameter, biarkan saja tanda kurung tersebut kosong.

```
1. const sayHello = () => console.log("Selamat pagi semuanya!");
2.
3. sayHello()
4.
5. /* output:
6. Selamat pagi semuanya!
7. */
```

Atau beberapa developer biasanya mengubah tanda kurung menjadi garis bawah (`_`) agar penulisannya lebih simpel.

## Block Body Function

Fungsi merupakan *mini program* sehingga sangat mungkin terdapat lebih dari satu logika di dalamnya. Seperti yang kita ketahui bahwa logika-logika pada pemrograman terdiri dari banyak *expression* ataupun *statement*.

Pada contoh kode arrow function sebelumnya kita hanya menuliskan satu buah expression sehingga penulisannya bisa sangat ringkas. Namun bagaimana jika dalam sebuah fungsi terdapat banyak logika di dalamnya? Apakah bisa dituliskan menggunakan arrow function? Jawabannya tentu bisa!

Sama seperti regular function, arrow function sebenarnya tidak benar-benar menghilangkan tanda kurung kurawal (`{ }`) dalam penulisannya. Tanda kurung kurawal pun berfungsi sama seperti regular function yakni menampung *body function* di mana tempat logika fungsi dituliskan. Penulisan tanda kurung kurawal menjadi opsional ketika body fungsi hanya terdiri dari satu expression atau statement saja.

Jika kita butuh lebih dari satu baris dalam body function, kita bisa menuliskannya seperti ini:

```
1. const sayHello = language => {
2.   if(language.toUpperCase() === "INDONESIA") {
3.     return "Selamat Pagi!";
4.   } else {
5.     return "Good Morning!";
6.   }
7. }
8.
9. console.log(sayHello("Indonesia"));
10.
11. /* output:
12. Selamat Pagi!
13. */
```

Selain itu juga, kita perlu menuliskan kembali keyword `return` agar fungsi tersebut dapat mengembalikan nilai.

[← KEMBALI KE MATERI SEBELUMNYA](#)[LANJUTKAN KE MATERI BERIKUTNYA →](#)



image  
click bisa  
dapatkan thumbnail

image  
click bisa  
dapatkan thumbnail

#### PERUSAHAAN

Tentang Kami

Blog

Berita Terbaru



#### PROGRAM

Academy

Challenge

Event

Job

Rewards

#### SUPPORT

Bantuan

FAQ

Hubungi Kami

