

[DISKUSIKAN MATERI](#)[LAPORKAN MATERI](#)

## Daftar Modul

[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 \(Gratis\)](#) ✓

## setTimeout

Fungsi `setTimeout()` merupakan cara yang paling mudah untuk membuat kode kita dijalankan secara asynchronous. Fungsi ini menerima dua buah parameter. Pertama adalah fungsi yang akan dijalankan secara asynchronous, dan kedua adalah nilai *number* dalam *millisecond* sebagai nilai tunggu sebelum fungsi dijalankan. Contoh penggunaannya adalah seperti ini:

```
1. console.log("Selamat datang!");
2. setTimeout(() => {
3.   console.log("Terimakasih sudah mampir, silakan datang kembali!");
4. }, 3000)
5. console.log("Ada yang bisa dibantu?")
```

Jika kita hanya mengenal program secara synchronous, maka kita dapat membayangkan hasilnya memiliki urutan sebagai berikut:

- Mencetak -> Selamat datang!
- Menunggu selama tiga detik
- Mencetak -> Terima kasih sudah mampir, silakan datang kembali!
- Mencetak -> Ada yang bisa dibantu?

Namun nyatanya `setTimeout()` tidak akan menghentikan JavaScript untuk melakukan eksekusi kode pada baris berikutnya. Sehingga urutannya menjadi seperti berikut:

- Mencetak -> Selamat datang!
- Mencetak -> Ada yang bisa dibantu?
- Menunggu selama tiga detik
- Mencetak -> Terima kasih sudah mampir, silakan datang kembali!

Jika kode tersebut dijalankan, maka akan menampilkan output sebagai berikut:

```
node v10.16.0
[]
```

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

### PERUSAHAAN

[Tentang Kami](#)[Blog](#)[Berita Terbaru](#)

### PROGRAM

[Academy](#)[Challenge](#)[Event](#)[Job](#)[Rewards](#)

### SUPPORT

[Bantuan](#)[FAQ](#)[Hubungi Kami](#)