



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)

Promise All

Materi sebelumnya kita belajar bagaimana promise dapat menangani situasi di mana terdapat *asynchronous process* yang saling membutuhkan untuk melaksanakan tugasnya. Namun bagaimana jika kita ingin menjalankan banyak promise sekaligus namun tidak memperdulikan urutan?

Kita sering pergi ke cafe untuk menikmati secangkir kopi bersama teman kuliah ataupun rekan kerja. Ketika memesan kopi, biasanya kita lakukan secara bersamaan. Meskipun kopi yang kita pesan berbeda, tak jarang pelayan menghidangkannya berbarengan dengan kopi yang teman kita pesan. Nah pada kasus inilah pelayan menggunakan teknik **Promise.all()**.

Method **Promise.all()** dapat menerima banyak promise (dalam bentuk array) pada parameternya. Kemudian method tersebut akan mengembalikan nilai seluruh hasil dari promise yang kita tetapkan dalam bentuk array.

Contohnya seperti berikut ini:

```
1. const arabicaOrder = () => {
2.   return new Promise(resolve => {
3.     setTimeout(() => {
4.       resolve("Kopi arabika selesail")
5.     }, 4000)
6.   })
7. }
8.
9. const robustaOrder = () => {
10.  return new Promise(resolve => {
11.    setTimeout(() => {
12.      resolve("Kopi robusta selesail")
13.    }, 2000)
14.  })
15. }
16.
17. const libericaOrder = () => {
18.  return new Promise(resolve => {
19.    setTimeout(() => {
20.      resolve("Kopi liberica selesail")
21.    }, 3000)
22.  })
23. }
```

Ketika kode di atas dieksekusi, kita perlu menunggu 4 detik sebelum akhirnya output pada console ditampilkan. Ini menunjukan bahwa **Promise.all()** akan mengembalikan nilai jika seluruh proses promise di dalamnya selesai dijalankan.

Urutan nilai yang dihasilkan oleh method ini sesuai dengan urutan promise yang kita tentukan pada parameternya

```
1. const promises = [arabicaOrder(), robustaOrder(), libericaOrder()];
2.
3. Promise.all(promises)
4.   .then(resolvedValue => {
5.     console.log(resolvedValue);
6.   })
```

Nilai kopi arabika akan tetap berada di posisi pertama, meskipun proses pembuatannya memakan waktu paling lama.

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

image
click bisa
bentuk manual

image
click bisa
bentuk manual

Tentang Kami

Blog

Berita Terbaru



Academy

Challenge

Event

Job

Rewards

Bantuan

FAQ

Hubungi Kami

