DISKUSIKAN MATERI

LAPORKAN MATERI

Daftar Modul

Masukkan kata kunci

onFulfilled and onRejected Functions

Untuk menangani nilai yang dikirimkan oleh `resolve()` ketika Promise onFulfilled, kita gunakan method `.then()` pada objek promise yang kita buat. Lalu di dalam method `.then()` kita berikan parameter berupa function yang berguna sebagai *handle callback*. Contohnya seperti ini:

```
const executorFunction = (resolve, reject) => {
 const isCoffeeMakerReady = true;
  if(isCoffeeMakerReady) {
    resolve("Kopi berhasil dibuat");
  } else {
    reject("Mesin Kopi tidak bisa digunakan!")
  }
}

const handlerSuccess = resolvedValue => {
 console.log(resolvedValue);
}

const makeCoffee = new Promise(executorFunction);
makeCoffee.then(handlerSuccess)

/* output:
Kopi berhasil dibuat
*/
```

Mari kita bedah kode yang ada di atas:

- `makeCoffee` merupakan objek promise yang akan menghasilkan `resolve()` dengan membawa nilai 'Kopi berhasil dibuat'.
- Lalu kita mendeklarasikan fungsi `handlerSuccess()` yang mencetak nilai dari parameternya.
- Kemudian kita memanggil method `.then()` dari `makeCoffee` dan memberikan `handlerSuccess` sebagai parameternya.
- Ketika `makeCoffee` terpenuhi (fulfilled), maka `handlerSuccess()` akan terpanggil dengan parameter nilai yang dibawa oleh `resolve()`. Sehingga output akan menghasilkan "Kopi berhasil dibuat".

Namun bagaimana jika objek promise menghasilkan kondisi *rejected*? Bagaimana cara menangani nilai yang dikirimkan oleh `reject()`?

Kita dapat menyimpan kedua *handle callback* baik ***onFulfilled*** atau ***onRejected*** pada parameter method `.then()`, yang perlu kita perhatikan adalah pastikan handle callback untuk ***onFulfilled*** disimpan pada parameter pertama seperti ini:

```
makeCoffee.then(handlerSuccess, handlerRejected);
```

Dengan begitu kita tetap dapat menangani objek promise meskipun dalam kondisi *rejected*.

```
const executorFunction = (resolve, reject) => {
 const isCoffeeMakerReady = false;
 if(isCoffeeMakerReady) {
   resolve("Kopi berhasil dibuat");
 } else {
   reject("Mesin Kopi tidak bisa digunakan!")
 }
}

const handlerSuccess = coffee => {
 console.log(coffee);
}

const handlerRejected = rejectionReason => {
 console.log(rejectionReason)
}

const makeCoffee = new Promise(executorFunction);
makeCoffee.then(handlerSuccess, handlerRejected);

/* output:
Mesin Kopi tidak bisa digunakan!
*/
```

dicoding

image
click bila
belum muncul

DIBANTU