DISKUSIKAN MATERI

LAPORKAN MATERI

Daftar Modul

Masukkan kata kunci

This Keyword

Perbedaan karakteristik dari arrow function dan regular function selanjutnya ada pada penggunaan keyword *this*. Penjelasan dari *this* sendiri menyusul di materi class. Namun kita akan bahas sedikit mengenai ini untuk menggambarkan perbedaan ketika *this* digunakan oleh arrow function dan regular function.

Jika sebuah regular function dipanggil dengan menggunakan *keyword* *new*. Maka nilainya akan menjadi objek, contohnya:

```
function People(name, age, hobby) {
    this.name = name;
    this.age = age;
    this.hobby = hobby;
}

const programmer = new People("John", 18, ["Coding", "Read book", "Ping-pong"]);

console.log(programmer.name);
console.log(programmer.age);
console.log(programmer.hobby);

/* output:
John
18
[ 'Coding', 'Read book', 'Ping-pong' ]
*/
```

Objek yang dibuat menggunakan function dengan keyword *new*, sama halnya seperti kita membuat objek seperti menggunakan objek literals { }.

```
const programmer = {
    name: "John",
    age: 18,
    hobby: ["Coding", "Read book", "Ping-Pong"]
}

console.log(programmer.name);
console.log(programmer.age);
console.log(programmer.hobby);

/* output:
John
18
[ 'Coding', 'Read book', 'Ping-pong' ]
*/
```

Pada objek, *this* keyword mengembalikan nilai objeknya sendiri. *this* dapat digunakan untuk mengelola properti pada objeknya. Namun jika fungsi dipanggil tanpa menggunakan keyword new, *this* akan memiliki nilai global object (Window jika di browser).

Sedangkan fungsi yang dibuat dengan menggunakan gaya arrow tidak akan pernah memiliki nilai *this*, yang artinya kita tidak pernah bisa membuat objek menggunakan arrow function. Jika kita menggunakan *this* pada arrow function maka nilai *this* tersebut merupakan nilai objek di mana arrow function itu berada.

Perhatikan kedua contoh kode berikut:

Regular Function Arrow Function

```
function People(name, age, hobby) {
    this.name = name;
    this.age = age;
    this.hobby = hobby;
}


// menambahkan introMyself ke People
People.prototype.introMyself = function () {
    // this -> People
    setTimeout(function() {
        // this -> ??
        console.log(`Hello! Nama saya ${this.name}, umur saya ${this.age}.`)
        console.log(`Hobby saya adalah ${this.hobby}`)
    }, 300)
}


const programmer = new People("John", 18, ["Coding", "Read book", "Ping-pong"]);
programmer.introMyself();


/* output:
```

Fungsi yang dituliskan di dalam `setTimeout()` dipanggil tanpa `new`. Itu berarti nilai dari `this` jika digunakan di dalam fungsi tersebut adalah *global object*. Itulah mengapa output akan menghasilkan nilai `undefined` ketika properti `name`, `age`, dan `hobby` dipanggil.

Berbeda ketika kita menuliskan arrow function di dalam `setTimeout()`, nilai `this` memiliki nilai objek sesuai dengan konteksnya (`People`). Arrow function akan sangat berguna untuk kasus seperti ini.

← KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA →