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)

Destructuring Object

Penulisan destructuring object pada ES6 sintaks menggunakan objek literal { } di sisi kiri dari operasi assignment.

```
const profile = {
    firstName: "John",
    lastName: "Doe",
    age: 18
}

const {firstName, lastName, age} = profile;

console.log(firstName, lastName, age);

/* output:
John Doe 18
*/
```

Pada contoh di atas tanda kurung kurawal { } merepresentasikan objek yang akan didestruksikan. Di dalamnya terdapat `firstName`, `lastName`, dan `age` yang merupakan variabel di mana kita menyimpan nilai properti dari objek `profile`. Kita juga perlu perhatikan penamaan variabel-variabelnya. Pastikan penamaannya sama seperti yang dimiliki oleh properti objeknya. Melalui nama variabel inilah nilai-nilai properti objek akan dimasukkan secara otomatis. Sehingga variabel `firstName` akan berisikan nilai `profile.firstName`, `lastName` akan berisikan nilai `profile.lastName`, begitu juga dengan variabel age akan berisikan nilai `profile.age`.

Dalam *destructuring object*, kita bisa menspesifikasikan salah satu nilai yang ingin kita desktruksikan. Sehingga kita tidak perlu membuat variabel sebanyak properti yang dimiliki objeknya. contohnya:

```
const {lastName} = profile;
```

Destructuring Assignment

Pada contoh sebelumnya kita melakukan destructuring object pada deklarasi variabel, namun pada kasus tertentu mungkin kita perlu melakukannya pada variabel yang sudah dideklarasikan. Atau kita ingin mengubah nilainya dengan nilai properti di objek.

Dalam kasus tersebut, kita bisa melakukannya dengan seperti ini:

```
const profile = {
    firstName: "John",
    lastName: "Doe",
    age: 18
}

let firstName = "Dimas";
let age = 20;

// menginisialisasi nilai baru melalui object destruction
({firstName, age} = profile);

console.log(firstName);
console.log(age);

/* output:
John
18
*/
```

Saat melakukan *destructuring assignment* kita perlu menuliskan *destructuring object* di dalam tanda kurung. Jika tidak dituliskan di dalamnya, tanda buka kurung kurawal akan membuat JavaScript mengira kita membuat *block statement*, dan *block statement* tentu tidak bisa berada pada sisi kiri *assignment*.

```
// tidak bisa karena JavaScript mengira kita membuat block statement
// block statement tidak bisa berada pada sisi kiri assignment
{firstName, age} = profile;
```

Nah inilah fungsinya tanda kurung. Ia akan memberitahu JavaScript bahwa tanda kurawal yang di dalamnya bukan sebuah block statement, melainkan sebuah expression. Sehingga assignment dapat dilakukan.

```
({firstName, age} = profile);
```

Default Values

Ketika kita mendestruksikan objek dan kita menetapkan variabel dengan nama yang bukan merupakan properti dari objek, maka nilai dari variabel tersebut menjadi `undefined`. Contohnya:

```
const profile = {
    firstName: "John",
    lastName: "Doe",
    age: 18
}


const {firstName, age, isMale} = profile;

console.log(firstName)
console.log(age)
console.log(isMale)

/* output:
John
18
undefined
*/
```

Alternatifnya, kita bisa secara opsional mendefinisikan nilai default pada properti tertentu jika tidak ditemukan. Untuk melakukanya tambahkan tanda assignment (=) setelah nama variabel dan tentukan nilai defaultnya seperti ini:

```
const profile = {
    firstName: "John",
    lastName: "Doe",
    age: 18
}


const {firstName, age, isMale = false} = profile;

console.log(firstName)
console.log(age)
console.log(isMale)

/* output:
John
18
false
*/
```

Ketika menambahkan *default value*, jika properti tidak ditemukan nilai *default* akan diterapkan pada variabel.

Assigning to Different Local Variable Names

Sampai saat ini kita tahu bahwa untuk mendekstruksikan objek pada variabel lokal kita perlu menyeragamkan penamaan lokal variabel dengan properti objeknya. Namun sebenarnya dalam mendestruksikan objek kita bisa menggunakan penamaan variabel lokal yang berbeda. ES6 menyediakan sintaks tambahan yang membuat kita dapat melakukan hal tersebut. Penulisannya mirip seperti ketika kita membuat properti beserta nilainya pada objek.

Contohnya seperti ini:

```
const profile = {
    firstName: "John",
    lastName: "Doe",
    age: 18
}

const {firstName: localFirstName, lastName: localLastName, age: localAge} = profile;

console.log(localFirstName);
console.log(localLastName);
console.log(localAge);

/* output:
John
Doe
18
*/
```

⬅ KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA ➡