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)

Node.js Export and Import Modules

Sebelumnya sudah dijelaskan bahwa *module* bekerja dengan cara *exporting* atau *importing* nilai baik itu variabel, fungsi, array, objek ataupun class agar dapat digunakan pada berkas JavaScript lain. Dalam satu berkas JavaScript terdiri dari satu module, dan di dalamnya kita dapat melakukan *export* lebih dari satu nilai.

Dalam *environment* Node.js, untuk melakukan *import* dan *export* module kita gunakan ***module.exports***. Setiap berkas JavaScript yang berjalan pada Node, memiliki objek ***module*** lokal yang memiliki properti exports. Properti tersebut digunakan untuk mendefinisikan nilai apa yang akan diekspor dari berkas tersebut.

Kode di bawah ini merupakan contoh bagaimana cara melakukan export nilai dengan menggunakan ***module.exports***.

```
1.  const coffeeStock = {
2.    arabica: 100,
3.    robusta: 150,
4.    liberica: 200
5.  }
6.
7.  module.exports = coffeeStock;
```

Berkas ***state.js***

Kode ***module.exports = coffeeStock*** membuat object ***coffeeStock*** diterapkan sebagai nilai dari ***module.exports***. Di mana nanti nilai properti exports ini akan digunakan (import) pada berkas JavaScript lain.

Jika mencoba melihat nilai module yang ada pada berkas state.js dengan menambahkan kode ***console.log(module)*** di akhir berkasnya. Maka kita akan melihat objek ***coffeeStock*** menjadi nilai dari properti ***exports***.

```
1.  Module {
2.    id: '.',
3.    exports: { arabica: 100, robusta: 150, liberica: 200 },
4.    parent: null,
5.    filename: '/home/runner/163-02-export-import/state.js',
6.    loaded: false,
7.    children: [],
8.    paths:
9.     [ '/home/runner/163-02-export-import/node_modules',
10.      '/home/runner/node_modules',
11.      '/home/node_modules',
12.      '/node_modules' ] }
```

Lalu bagaimana caranya untuk melakukan import atau menggunakan objek yang sudah di-export pada berkas lain? Caranya adalah dengan menggunakan method ***require()***.

index.js state.js

```
1.  const coffeeStock = require('./state.js');
2.
3.
4.  console.log(coffeeStock);
5.
6.
7.  /* output
8.  { arabica: 100, robusta: 150, liberica: 200 }
9.  */
```

Dalam inisialisasi variabel *coffeeStock* (nama variabel bebas kita tentukan), kita gunakan method *require()* dengan memberikan parameter lokasi dari berkas *state.js*. Dengan begitu variabel *coffeeStock* akan memiliki nilai *module.exports* yang sama pada berkas *state.js*. Setelah mendapatkan nilainya, kita bebas menggunakannya layaknya variabel lokal pada biasanya.

index.js state.js

```
1.  const coffeeStock = require('./state.js');
2.
3.
4.  const makeCoffee = (type, miligrams) => {
5.   if(coffeeStock[type] >= miligrams) {
6.     console.log("Kopi berhasil dibuat!")
7.   } else {
8.     console.log("Biji kopi habis!")
9.   }
10.  }
11.
12.
13.  makeCoffee("robusta", 80);
14.
15.
16.  /* output:
17.  Kopi berhasil dibuat!
18.  */
```

> "Tips dalam memberikan lokasi pada method ***require()***: Jika kita gunakan lokasi yang relatif (dapat berubah/dipindahkan), pastikan awali dengan menuliskan *./*. Contohnya, berkas *index.js* dan *state.js* berada pada folder yang sama, maka kita cukup menuliskannya dengan ***./state.js***."

⬅ KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA ➡