



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) ✓

Solution : Module

Apakah Anda sudah berhasil menerapkan module pada proyek Club Finder? Jika belum, mari kita lakukan bersama-sama.

Langkah pertama adalah hapus tag `<script>` selain `app.js` pada berkas `index.html`.

```
1. <script src="src/script/data/clubs.js"></script>
2. <script src="src/script/data/data-source.js"></script>
3. <script src="src/script/view/main.js"></script>
4. <script src="app.js"></script>
```

Menjadi:

```
1. <script src="app.js"></script>
```

Lalu untuk mengaktifkan fitur module pada browser, tambahkan attribute type dengan nilai `"module"` pada tag `<script>`.

```
1. <script src="app.js" type="module"></script>
```

Kemudian kita lakukan export nilai-nilai yang akan digunakan tiap berkas JavaScript-nya. Mulai dari berkas `clubs.js`.

Karena pada `clubs.js` kita hanya melakukan export pada satu nilai variabel, maka gunakanlah `export default` untuk mengeksport nilai variabel `clubs`.

Tambahkan kode berikut di akhir berkas `clubs.js`:

```
1. export default clubs;
```

Selanjutnya kita export juga class `DataSource` yang ada pada berkas `data-source.js`. Tambahkan kode berikut di akhir baris kodenya:

```
1. export default DataSource;
```

Nilai terakhir yang harus kita export adalah fungsi main yang berada pada berkas `main.js`. Tambahkan kode berikut di akhir baris kodenya:

```
1. export default main;
```

Setelah kita selesai mengeksport seluruh nilai yang akan digunakan. Selanjutnya kita impor nilai-nilai yang dibutuhkan pada masing-masing berkas JavaScript.

Yang pertama, berkas `data-source.js` membutuhkan nilai `clubs` yang berasal dari module `clubs.js`. Untuk mendapatkan nilainya, kita perlu mengimpor nilai `clubs` dengan cara menambahkan kode berikut di awal baris kode `data-source.js`:

```
1. import clubs from './clubs.js';
```

Yang kedua, class `DataSource` digunakan pada berkas `main.js` sehingga kita juga perlu mengimpor `DataSource` pada module `data-source.js`. Tambahkan kode berikut di awal baris berkas `main.js`:

```
1. import DataSource from '../data/data-source.js';
```

Dan yang terakhir pada berkas **app.js** kita membutuhkan nilai main dari module **main.js**. Sehingga lakukan juga impor nilai main dengan cara menambahkan kode berikut di awal baris berkas **app.js**:

```
1. import main from './src/script/view/main.js';
```

Setelah menerapkan module, kemungkinan besar Anda akan mengalami eror:

```
1. Access to script at 'file:///...../ClubFinder/app.js' from origin 'null' has been blocked by CORS policy: Cross origin requests are only supported for protocol schemes: http, data, chrome, chrome-extension, https.
```

Eror terjadi karena ketika menggunakan **import**, kita harus tunduk terhadap peraturan CORS. Apa itu? Nanti akan kita bahas detail pada modul Asynchronous JavaScript Request. Namun singkatnya, kita tidak bisa lagi membuka proyek dengan membuka berkas **index.html** secara langsung. Namun harus menggunakan local http server (dijalankan melalui localhost).

Jika Anda terbiasa dengan Node.js dan sudah menginstalnya, Anda bisa gunakan package **http-server** yang tersedia pada tautan berikut: <https://www.npmjs.com/package/http-server>. Namun bagi Anda yang belum terbiasa menggunakannya, Anda bisa memasang extensions pada code editor yang digunakan.

- **Visual Studio Code** : Untuk Anda yang menggunakan VSCode sebagai text editor, Anda bisa memasang extensions bernama "**Live Server**" untuk menjalankan website pada localhost. Panduan cara memasang extensions pada VSCode dapat Anda pelajari pada tautan berikut: <https://code.visualstudio.com/docs/editor/extension-gallery>
- **Atom Text Editor** : Untuk Anda yang menggunakan Atom sebagai text editor, Anda bisa memasang extensions bernama "**atom-live-server**" untuk menjalankan website pada localhost. Panduan cara memasang extensions pada Atom dapat Anda pelajari pada tautan berikut: <https://flight-manual.atom.io/using-atom/sections/atom-packages/>
- **Bracket** : Untuk Anda yang menggunakan bracket. Anda bisa menjalankan project pada localhost dengan memanfaatkan fitur live preview yang tersedia langsung pada text editor tersebut.

Langkah dari solution ini bisa Anda temukan juga pada repository berikut:

<https://github.com/dicodingacademy/a163-bfwd-labs/tree/108-club-finder-module-solution>

← KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA →



PERUSAHAAN

Tentang Kami

Blog

Berita Terbaru



PROGRAM

Academy

Challenge

Event

Job

Rewards

SUPPORT

Bantuan

FAQ

Hubungi Kami

