

Filesystem

Seluruh data di komputer dikelola dan diakses melalui *filesystem*. Ketika menjalankan kode JavaScript pada browser, sangat penting untuk membatasi JavaScript dalam mengakses filesystem. Teknik ini dinamakan dengan *sandboxing*. Sandboxing melindungi kita dari program jahat serta tindakan pencurian yang dapat merampas privasi penggunanya.

Bagaimana dengan JavaScript yang dijalankan di back-end? Limitasi tentu tetap ada, namun tidak seketat ketika JavaScript dieksekusi pada browser. Di back-end malah filesystem menjadi fitur esensial karena dalam pengembangan back-end akan sering sekali mengakses atau menulis sebuah berkas di dalam komputer.

Node.js menyediakan core modules `fs` yang dapat mempermudah kita dalam mengakses filesystem. Setiap method yang ada di module `fs` tersedia dalam dua versi, yakni versi asynchronous (default) dan versi synchronous.

Tentu Anda sudah tahu apa itu asynchronous dan synchronous kan? Jika belum, pelajari kedua hal tersebut pada kelas [Belajar Dasar Pemrograman JavaScript](#).

Untuk mengakses berkas pada komputer kita dapat menggunakan method `fs.readFile()`. Method ini menerima tiga argumen yakni: lokasi berkas, encoding, dan callback function yang akan terpanggil bila berkas berhasil/gagal diakses.



```
1. const fs = require('fs');
2.
3. const fileReadCallback = (error, data) => {
4.     if(error) {
5.         console.log('Gagal membaca berkas');
6.         return;
7.     }
8.     console.log(data);
9. };
10.
11. fs.readFile('todo.txt', 'UTF-8', fileReadCallback);
```

Sebagai alternatif, Anda juga bisa gunakan method versi synchronous `fs.readFileSync()` .



```
1. const fs = require('fs');
2.
3. const data = fs.readFileSync('todo.txt', 'UTF-8');
4. console.log(data);
```

Latihan: Filesystem

Pada latihan kali ini, Anda akan ditugaskan untuk membuat program JavaScript yang dapat membaca teks dari berkas `.txt`. Karena Anda sudah

belajar filesystem, tentu ini tidak akan sulit.

Silakan buat folder baru dengan nama filesystem. Di dalamnya buat dua berkas, yakni **index.js** dan **notes.txt**.

✓ NODEJS-BASIC

> events

✓ filesystem

JS index.js

≡ notes.txt

Di dalam masing-masing berkas, tuliskan kode/teks berikut:

notes.txt index.js

1. Di hari minggu saya akan:
2. 1. Berolahraga pagi.
3. 2. Membersihkan halaman rumah.
4. 3. Menonton film.
5. 4. Membaca buku Laskar Pelangi.

Tugas Anda tertulis jelas seperti yang ada pada berkas index.js. Cobalah tampilkan teks pada berkas notes.txt pada console menggunakan filesystem. Jika TODO sudah selesai dikerjakan, eksekusi berkas index.js dengan perintah:

1. node ./filesystem/index.js

Bila TODO berhasil Anda kerjakan, maka outputnya tampak seperti ini:

```
C:\javascript-projects\nodejs-basic>node ./filesystem/index.js
Di hari minggu saya akan:
1. Berolahraga pagi.
2. Membersihkan halaman rumah.
3. Menonton film.
4. Membaca buku Laskar Pelangi.
```

Tips: Anda bisa gunakan method

```
1. path.resolve(__dirname, 'notes.txt');
```

Dari core modules **path** dalam menetapkan alamat berkas secara lengkap dan dinamis.

Mengalami kesulitan dalam menyelesaikan latihan?

Cobalah untuk ulas kembali materi yang diberikan atau tanyakan kesulitan yang Anda alami pada [forum diskusi](#). Hindari melihat atau membandingkan [kode solusi](#) pada latihan filesystem. sebelum Anda mencobanya sendiri.

