



# Writable Stream

Pada materi sebelumnya kita sudah belajar bagaimana membaca berkas menggunakan teknik stream (readable stream). Tapi apakah Anda tahu bahwa teknik stream juga dapat digunakan untuk menulis berkas? Teknik ini disebut *writable stream*.

Untuk membuat writable stream dalam menulis berkas gunakanlah method `createWriteStream()` dari core module `fs`.

```
1. const fs = require('fs');  
2.  
3. const writableStream = fs.createWriteStream('output.txt');
```

Fungsi ini menerima satu argumen yakni alamat berkas untuk menyimpan hasil data yang dituliskan. Berkas output akan dibuat secara otomatis jika tidak ada, namun bila berkas tersebut sudah ada, maka data sebelumnya akan tertimpa! Anda harus hati-hati yah!

Lalu untuk menuliskan data pada writable stream, gunakan method `write()`.

```
1. const fs = require('fs');  
2.
```

```
3. const writableStream = fs.createWriteStream('output.txt');
4.
5. writableStream.write('Ini merupakan teks baris pertama!\n');
6. writableStream.write('Ini merupakan teks baris kedua!\n');
7. writableStream.end();
```

Method `end()` digunakan untuk menandakan akhir dari writable stream sekaligus bisa digunakan sebagai penulisan writeable terakhir.

```
1. const fs = require('fs');
2.
3. const writableStream = fs.createWriteStream('output.txt');
4.
5. writableStream.write('Ini merupakan teks baris pertama!\n');
6. writableStream.write('Ini merupakan teks baris kedua!\n');
7. writableStream.end('Akhir dari writable stream!');
```

Kode di atas akan menghasilkan **output.txt** dengan data seperti ini:

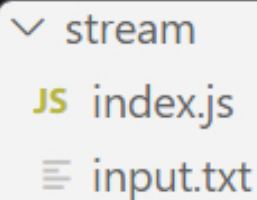
```
1. Ini merupakan teks baris pertama!
2. Ini merupakan teks baris kedua!
3. Akhir dari writable stream!
```

## Latihan: Stream

Sebelum mengakhiri modul ini, mari kita latihan lagi ya. Karena Anda sudah belajar teknik stream baik readable stream ataupun writable stream, kini saatnya kita praktikkan apa yang sudah Anda pelajari.

Kita akan kombinasikan kedua teknik tersebut untuk membuat program membaca teks dan memodifikasinya melalui teknik stream.

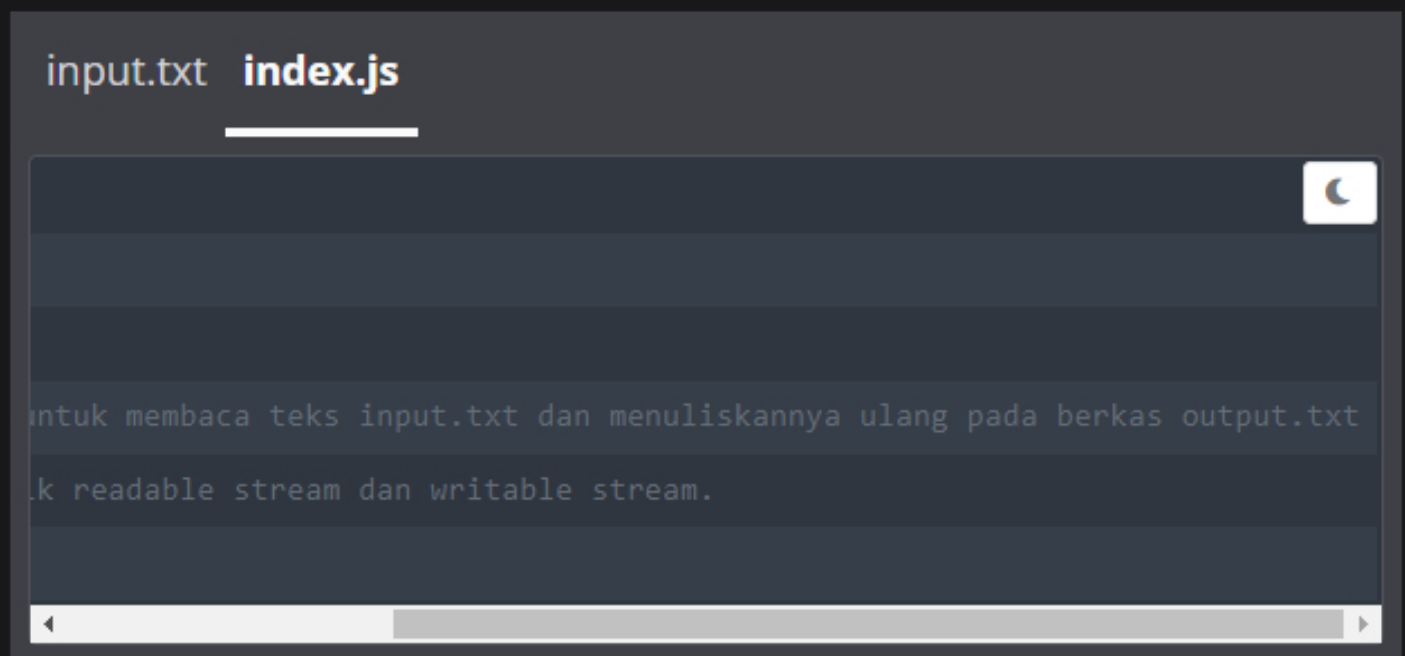
Silakan buat folder baru dengan nama **stream**. Di dalamnya buat dua berkas baru dengan nama **index.js** dan **input.txt**.



▼ stream

- JS index.js
- ≡ input.txt

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



input.txt **index.js**

```
// Untuk membaca teks input.txt dan menuliskannya ulang pada berkas output.txt  
// menggunakan teknik readable stream dan writable stream.
```

Seperti yang tertera pada berkas **index.js**, tugas Anda yaitu untuk membuat program membaca dan menulis ulang teks melalui stream dengan ketentuan berikut:

- Teks yang dibaca oleh readable stream memiliki ukuran 15 karakter tiap bagiannya. Tentukan nilai `highWaterMark`-nya.
- Tulis ulang teks dengan menggunakan teknik writable stream pada berkas **output.txt**. Untuk tiap bagian teks yang dibaca melalui readable stream, pisahkan dengan baris baru (`\n`).

Bila TODO selesai Anda kerjakan, maka eksekusi berkas `index.js` dengan menggunakan perintah:

```
1. node ./stream/index.js
```

Setelah mengeksekusi berkas tersebut, maka akan tercipta berkas baru dengan nama **output.txt**

```
▼ stream
  JS index.js
  ≡ input.txt
  ≡ output.txt
```

Yang berisikan teks berikut:

```
17. e irure dolor i
18. n reprehenderit
19. in voluptate v
20. elit esse cillu
21. m dolore eu fug
```

```
22. iat nulla paria
23. tur. Excepteur
24. sint occaecat c
25. upidatat non pr
26. oident, sunt in
27. culpa qui offi
28. cia deserunt mo
29. llit anim id es
30. t laborum.
31. null
```

**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 [stream](#) sebelum Anda mencobanya sendiri.



## Writeable Stream

