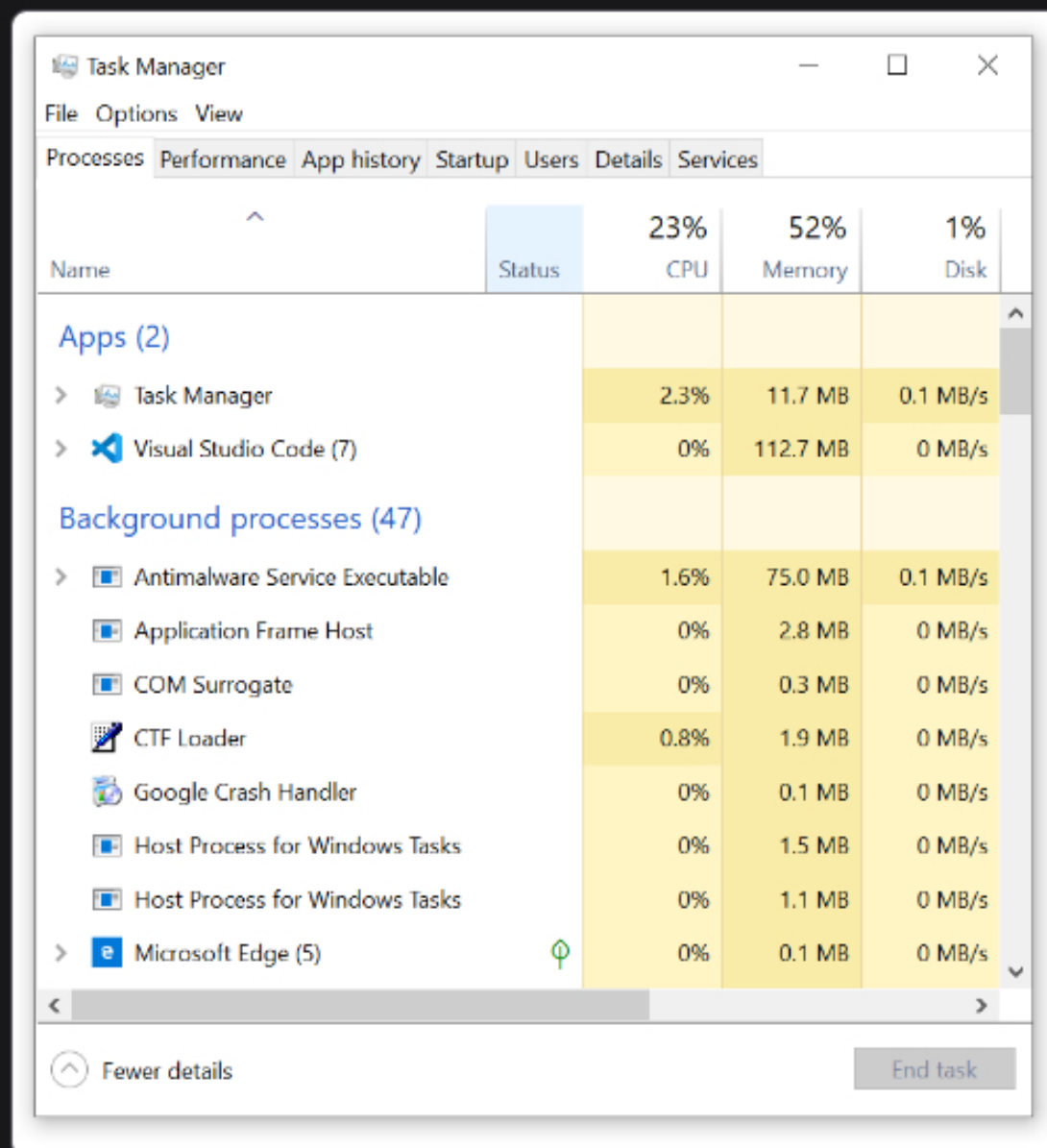


Process Object

Salah satu global objek yang penting untuk diketahui adalah process. Dalam ilmu komputer, process adalah sebuah program yang dijalankan pada satu atau lebih thread [4]. Anda bisa melihat proses yang sedang berjalan pada komputer Anda melalui Task Manager (Windows), System Monitor (Ubuntu), atau Activity Monitor (macOS).



Windows Task Manager

Pada Node.js, global objek `process` memiliki fungsi dan properti yang dapat memberikan informasi mengenai proses yang sedang berjalan.

Salah satu yang sering digunakan adalah properti `process.env`. Melalui properti ini kita dapat menyimpan nilai atau mendapatkan informasi mengenai *environment* yang digunakan selama proses sedang berlangsung. Contoh, `process.env` memiliki properti `process.env.PWD` yang menyediakan informasi mengenai lokasi di mana proses dijalankan; properti `process.env.USER` menyimpan informasi nama user pada komputer Anda; dan masih banyak properti lainnya. Anda bisa lihat daftar lengkap properti yang ada pada halaman [dokumentasi Node.js mengenai process.env](#).

Anda juga bisa secara manual menyimpan nilai di dalam `process.env`. Hal ini berguna untuk menentukan alur code seperti if-else dalam program berdasarkan *environment* yang Anda berikan. Contohnya, ketika Anda ingin nilai variabel `host` berbeda di kala pengembangan (*development*) dan produksi (*production*), Anda bisa membuat properti `NODE_ENV` pada `process.env`. Jadi, Anda bisa menentukan nilai `host` berdasarkan kondisi `NODE_ENV`.

app.js

```
1. const server = new Server({
2.   host: process.env.NODE_ENV !== 'production' ? 'localhost' : 'disprod
3. });
```

Untuk memberikan nilai pada properti `process.env`, kita dapat memberikannya ketika mengeksekusi berkas JavaScript. Caranya seperti ini:

Linux dan macOS Windows CMD

```
1. SET NODE_ENV=production && node app.js
```

Nilai yang ada pada `process.env` hanya dapat diakses di dalam cakupan proses Node.js. Itu berarti Anda tidak dapat menggunakan nilainya pada program lain seperti menampilkan nilainya melalui program `echo`.

```
1. // perintah ini tidak akan berjalan
2. node -e 'process.env.foo = "bar" && echo $foo'
```

Selain untuk menetapkan dan mendapatkan informasi mengenai environment, objek `process` memiliki kegunaan lain. Salah satunya adalah mendapatkan informasi tentang penggunaan memory ketika proses berjalan. Anda dapat mengakses informasi tersebut melalui fungsi `process.memoryUsage()`.

```
1. const memoryInformation = process.memoryUsage();
2.
3. console.log(memoryInformation);
4.
5. /* output
```

```
6. {  
7.   rss: 14569472,  
8.   heapTotal: 2654208,  
9.   heapUsed: 1788896,  
10.  external: 855681,  
11.  arrayBuffers: 9898  
12. }  
13. */
```

Yang terakhir dan tak kalah pentingnya adalah properti `process.argv`. Properti ini dapat menampung nilai baris perintah dalam bentuk array ketika menjalankan proses. Contoh jika kita menjalankan baris perintah berikut:

```
1. node app.js "harry" "potter"
```

Maka array `process.argv` akan bernilai:

- Elemen pertama : Alamat (path) lengkap dari lokasi node yang menjalankan prosesnya.
- Element kedua : Alamat (path) berkas JavaScript yang dieksekusi (app.js)
- Element ketiga : "harry"
- Element keempat : "potter"

Bila **app.js** memiliki kode seperti ini:

```
1. const firstName = process.argv[2];  
2. const lastName = process.argv[3];  
3.   
4. console.log(`Hello ${firstName} ${lastName}`);
```

Maka output yang dihasilkan tampak seperti ini:

```
1. Hello harry potter
```

Kita hanya membahas sedikit tentang properti dan fungsi yang ada pada **process** objek. Anda bisa mendalaminya dengan membaca [dokumentasi](#) tentang objek process.

Tantangan: Process Object

Anda sudah mengenal process object yang ada di Node.js. Agar lebih paham lagi, sekarang giliran Anda untuk coba sendiri penggunaan dari beberapa global object yang ada di Node.js.

Untuk tantangan kali ini, buatlah berkas **index.js** baru di dalam folder baru **process-object** pada proyek nodejs-basic.

▼ NODEJS-BASIC

▼ process-object

JS index.js

Kemudian, tuliskan starter code berikut pada **index.js**.

index.js

```
1. const initialMemoryUsage = // TODO 1
2. const yourName = // TODO 2
3. const environment = // TODO 3
4.
5. for(let i = 0; i <= 10000; i++) {
6. // Proses looping ini akan membuat penggunaan memori naik
7. }
8.
9. const currentMemoryUsage = // TODO 4
10.
11. console.log(`Hai, ${yourName}`);
12. console.log(`Mode environment: ${environment}`)
13. console.log(`Penggunaan memori dari ${initialMemoryUsage} naik ke ${cur
```

Selesaikan kode yang ditandai TODO dengan ketentuan berikut:

- **TODO 1** : Isi dengan nilai `heapUsed` dari instance `process.memoryUsage`.
- **TODO 2** : Isi dengan nilai index ke-2 dari `process.argv`.
- **TODO 3** : Isi dengan nilai `NODE_ENV` dari `process.env`.
- **TODO 4** : Isi dengan nilai `heapUsed` dari instance `process.memoryUsage`.

Setelah mengerjakan seluruh TODO, eksekusi berkas JavaScript dengan perintah:

Windows CMD Linux dan macOS Terminal

```
1. SET NODE_ENV=development && node ./process-object/index.js <Nama Anda>
```

Ganti <Nama Anda> dengan nama depan Anda. Bila TODO berhasil dikerjakan dengan baik, maka console akan menghasilkan output:

```
C:\javascript-projects\nodejs-basic>SET NODE_ENV=development && node ./process-object/index.js Dimas
Hai, Dimas
Mode environment: development
Penggunaan memori dari 1793812 naik ke 1795768
```

Mengalami kesulitan dalam menyelesaikan tantangan?

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

