



Menjalankan JavaScript Menggunakan Node.js

Terdapat dua cara dalam menjalankan kode JavaScript menggunakan Node.js. Yang pertama dengan memanfaatkan Node REPL dan yang kedua dengan mengeksekusi berkas berekstensi JS. Mari kita kupas keduanya!

BLACKBOX AI

The Node.js REPL

Node.js memiliki fitur REPL atau **Read-Eval-Print Loop**. Sesuai namanya, fitur ini berfungsi untuk membaca kode JavaScript, mengevaluasi kode tersebut, kemudian mencetak hasil evaluasinya ke console. Nah, untuk *loop*, berarti proses tersebut selalu berulang.

REPL merupakan fitur bawaan dari Node.js. Anda bisa mengaksesnya menggunakan perintah `node` pada Terminal.

BLACKBOX AI

PROBLEMS

TERMINAL

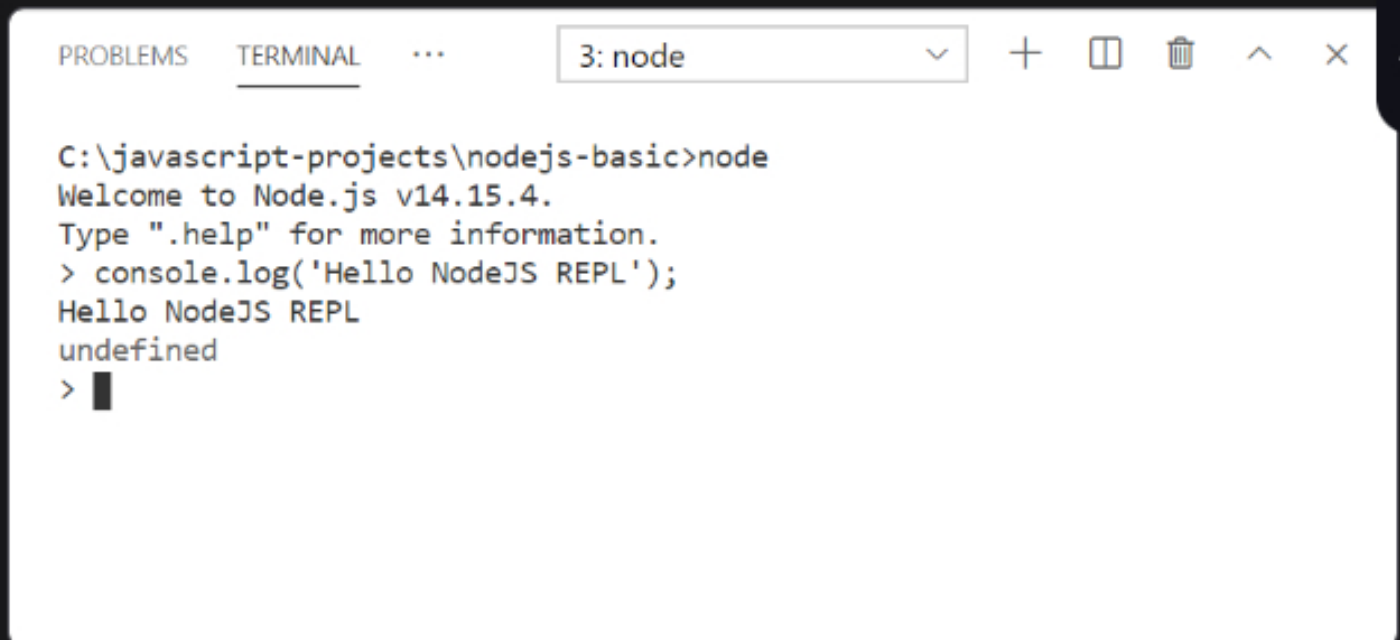


3: node



```
C:\javascript-projects\nodejs-basic>node
Welcome to Node.js v14.15.4.
Type ".help" for more information.
> █
```

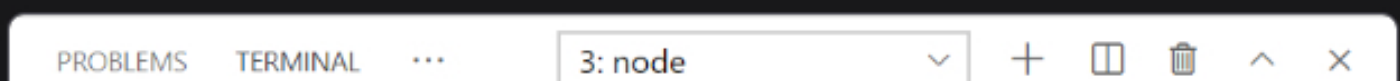
Tanda `>` pada Terminal menunjukkan Anda sudah masuk ke mode Node REPL. Sekarang, Anda bisa menuliskan kode JavaScript dan mengeksekusinya dengan menggunakan *enter*.

A screenshot of a VS Code terminal window. The title bar shows '3: node'. The terminal content is as follows:

```
C:\javascript-projects\nodejs-basic>node
Welcome to Node.js v14.15.4.
Type ".help" for more information.
> console.log('Hello NodeJS REPL');
Hello NodeJS REPL
undefined
> █
```

Lihat gambar di atas. Ketika mengeksekusi `console.log('Hello NodeJS REPL')`, selain pesan "Hello NodeJS REPL", nilai `undefined` juga tercetak. Hal tersebut karena REPL selalu menampilkan nilai evaluasi pada console. Karena method `console.log()` tidak mengembalikan nilai, jadi `undefined` lah yang tercetak pada console.

Untuk membuktikan hal itu, cobalah Anda tuliskan statement yang mengembalikan nilai. Contoh sederhananya `2+2`. Maka nilai `4` akan tercetak pada console.

A partial screenshot of a VS Code terminal window, showing the title bar '3: node' and the first line of the terminal content:

```
C:\javascript-projects\nodejs-basic>
```

```
C:\javascript-projects\nodejs-basic>node
Welcome to Node.js v14.15.4.
Type ".help" for more information.
> console.log('Hello NodeJS REPL');
Hello NodeJS REPL
undefined
> 2+2
4
> █
```

Cukup asik kan fitur REPL? Tapi kok terkesan hanya dapat mengeksekusi kode satu baris saja ya? Bila Anda beranggapan seperti itu, sebenarnya tidak tepat karena di dalam REPL terdapat mode editor yang berfungsi untuk menuliskan kode JavaScript lebih dari satu baris. Untuk menggunakan mode editor, Anda bisa tuliskan perintah `.editor`.

```
PROBLEMS  TERMINAL  ...  3: node  +  [ ]  [ ]  ^  x

C:\javascript-projects\nodejs-basic>node
Welcome to Node.js v14.15.4.
Type ".help" for more information.
> console.log('Hello NodeJS REPL');
Hello NodeJS REPL
undefined
> 2+2
4
> .editor
// Entering editor mode (Ctrl+D to finish, Ctrl+C to cancel)
█
```

Ketika masuk ke mode editor, Anda bisa secara leluasa menuliskan kode JavaScript lebih dari satu baris menggunakan *enter*. Fungsi untuk mengeksekusi kode digantikan dengan kombinasi tombol *CTRL+D*. Untuk keluar dari mode editor, gunakan kombinasi *CTRL+C*.

```
PS C:\javascript-projects\nodejs-basic> node
Welcome to Node.js v14.15.4.
Type ".help" for more information.
> console.log('Hallo NodeJS REPL');
Hallo NodeJS REPL
undefined
> 2+2
4
> .editor
// Entering editor mode (Ctrl+D to finish, Ctrl+C to cancel)
const welcome = (name) => {
  return `Welcome ${name}`;
}

welcome('Node REPL');
'Welcome Node REPL'
>
```

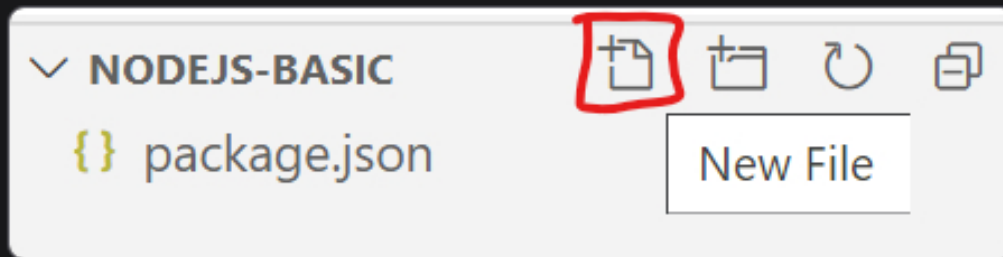
Nilai variabel yang Anda buat di REPL dapat diakses selama Anda masih berada di dalam REPL. Jika Anda menutup Terminal atau keluar dari REPL menggunakan perintah `.exit`, variabel yang sudah Anda buat sebelumnya tidak bisa diakses kembali. Itu artinya, REPL hanya menyimpan memory ketika session masih berlangsung.

Fitur REPL sangat berguna ketika Anda hendak melakukan kalkulasi sederhana, bereksperimen, atau belajar potongan kode JavaScript. Karena melalui REPL Anda bisa mengeksekusi kode JavaScript dan mendapatkan hasil dengan cepat tanpa harus membuat berkas JavaScript terlebih dahulu.

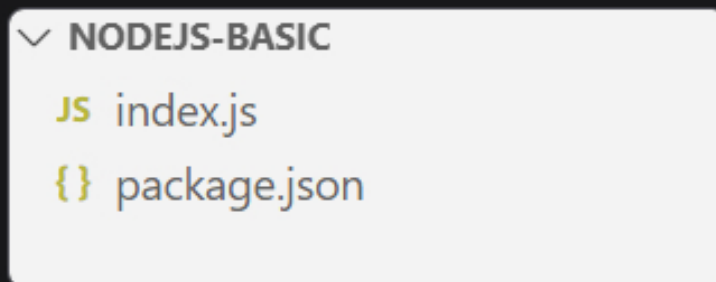
Running JavaScript File using Node.js

Cara lain untuk mengeksekusi kode JavaScript menggunakan Node.js

Cara lain untuk mengeksekusi kode JavaScript menggunakan Node.js adalah melalui berkas JS. Silakan buat berkas JavaScript pada proyek nodejs-basic. Gunakan VSCode agar lebih mudah yah.



Buatlah berkas JavaScript dengan nama **"index.js"**.



Di dalam berkas **index.js**, Anda bisa menuliskan kode JavaScript sesuka Anda. Pastikan kode yang Anda tulis menampilkan nilai di console yah, jadi Anda bisa melihat nilai yang tampak pada console. Jika bingung, silakan tuliskan saja kode berikut.

index.js

```
1. const message = (name) => {  
2.   console.log(`Hello ${name}`);  
3. }  
4.  
5. message('JavaScript');
```


Untuk mengeksekusi kode tersebut, silakan simpan perubahan berkas **index.js** dan buka kembali Terminal. Kemudian, tuliskan perintah:

```
1. node index.js
```

Node.js pun akan mengeksekusi berkas **'index.js'**. Bila Anda menuliskan kode seperti yang dicontohkan di atas, maka akan muncul teks **'Hello JavaScript'** pada console.

TERMINAL

3: cmd

```
C:\javascript-projects\nodejs-basic>node index.js  
Hello JavaScript
```

```
C:\javascript-projects\nodejs-basic>
```

