

JavaScript & Node.js dari Nol

Panduan Singkat Buat Anak IT

Hubungi saya:

Instagram @kevinaxellino

Tiktok @kevinaxellino

Daftar Isi

Daftar Isi	2
Apa itu JavaScript	3
Apakah JavaScript untuk Frontend atau Backend?	3
Apa itu Node.js	3
Apa Itu VS Code?	4
Cara Install Node.js di Windows dan Mac	4
1. Install Node.js di Windows:	4
2. Install Node.js di Mac:	5
Cara Install VS Code di Windows dan Mac	7
1. Install VS Code di Windows:	7
2. Install VS Code di Mac:	7
Menampilkan Teks dengan console.log	8
Operasi Dasar Matematika di JavaScript	9
Apa Itu Variabel di JavaScript?	10
Pengambilan Keputusan dengan If-Else	11
Perulangan dengan Loop di JavaScript	12
Mengenal Fungsi di JavaScript	13
Penutup	15

Apa itu JavaScript

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat website lebih interaktif. Misalnya:

- Saat kamu klik tombol, dan sesuatu berubah di halaman, itu karena JavaScript.
- Ketika website menunjukkan jam, atau animasi, itu juga JavaScript.

JavaScript sering disebut sebagai bahasa untuk **menghidupkan website**. Selain itu, JavaScript tidak hanya untuk browser, tapi juga bisa digunakan untuk membuat aplikasi server (backend) dengan bantuan Node.js.

Dengan JavaScript, kamu bisa memulai perjalanan menjadi seorang programmer yang bisa bekerja di banyak bidang, seperti membuat website, aplikasi, atau bahkan game!

Apakah JavaScript untuk Frontend atau Backend?

JavaScript awalnya dibuat untuk **frontend**, yaitu bagian depan website yang dilihat dan digunakan oleh pengguna. Contohnya, ketika kamu klik tombol dan muncul notifikasi, itu biasanya dibuat dengan JavaScript.

Namun, sejak hadirnya Node.js, JavaScript juga bisa digunakan untuk **backend**, yaitu bagian server yang mengolah data dan menjalankan logika aplikasi.

Jadi, JavaScript itu fleksibel. Bisa digunakan untuk **frontend** (browser) dan **backend** (server). Banyak developer menyebutnya sebagai bahasa **“full-stack”** karena bisa digunakan di kedua sisi.

Apa itu Node.js

Node.js adalah tool yang memungkinkan kamu menjalankan JavaScript di luar browser, seperti di komputer atau server. Dengan Node.js, kamu bisa membuat aplikasi seperti backend website, API, atau bahkan aplikasi desktop.

Node.js cepat dan efisien karena menggunakan mesin V8 dari Google Chrome untuk menjalankan kode JavaScript. Singkatnya, Node.js membuat JavaScript bisa digunakan di luar browser.

Apa Itu VS Code?

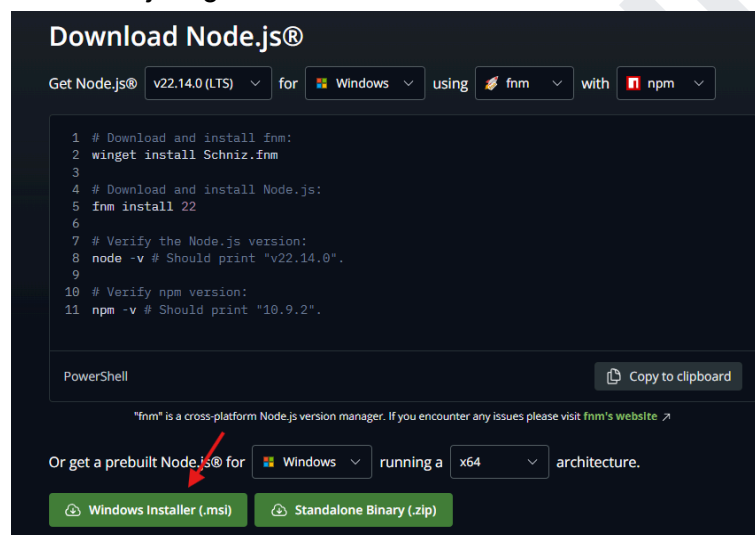
VS Code (Visual Studio Code) adalah code editor gratis yang ringan dan powerful. Dengan VS Code, kamu bisa menulis, mengedit, dan menjalankan kode dengan berbagai fitur seperti syntax highlighting, auto-completion, dan debugging.

Dalam PDF ini, kita akan menggunakan VS Code untuk menulis dan menjalankan kode JavaScript dengan Node.js.

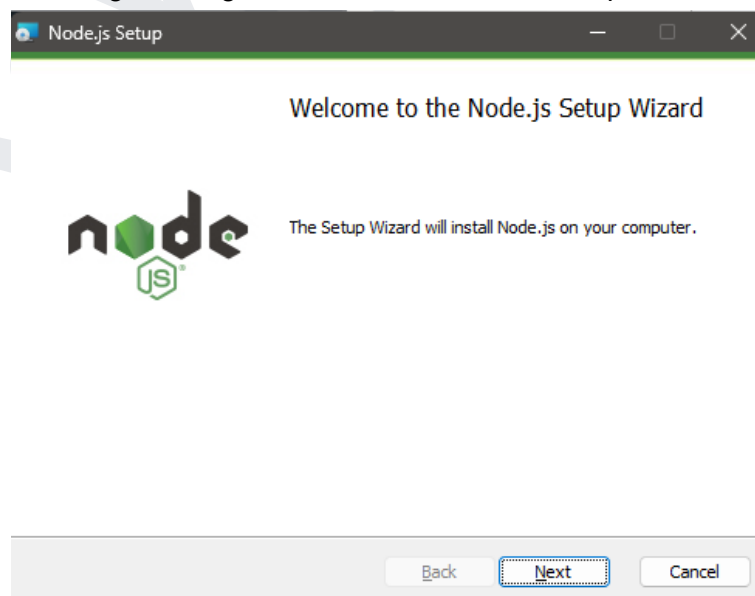
Cara Install Node.js di Windows dan Mac

1. Install Node.js di Windows:

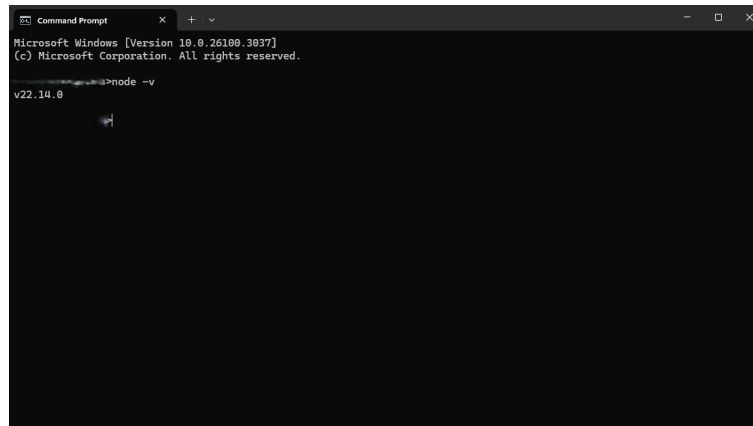
- a. Buka nodejs.org dan download versi LTS.



- b. Jalankan file instalasi (biasanya berekstensi .msi).
- c. Ikuti langkah-langkah di wizard instalasi. Cukup klik “Next” sampai selesai.



- d. Setelah selesai, buka Command Prompt (CMD) dan ketik `node -v` untuk memastikan Node.js terinstal dengan benar.



```
Microsoft Windows [Version 10.0.26100.3837]
(c) Microsoft Corporation. All rights reserved.

C:\>node -v
v22.14.0
```

2. Install Node.js di Mac:

- a. Buka nodejs.org dan download versi LTS.



Download Node.js®

Get Node.js® **v22.14.0 (LTS)** for **macOS** using **nvm** with **npm**

```
1 # Download and install nvm:
2 curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
3
4 # in lieu of restarting the shell
5 \. "$HOME/.nvm/nvm.sh"
6
7 # Download and install Node.js:
8 nvm install 22
9
10 # Verify the Node.js version:
11 node -v # Should print "v22.14.0".
12 nvm current # Should print "v22.14.0".
13
14 # Verify npm version:
15 npm -v # Should print "10.9.2".
```

Bash Copy to clipboard

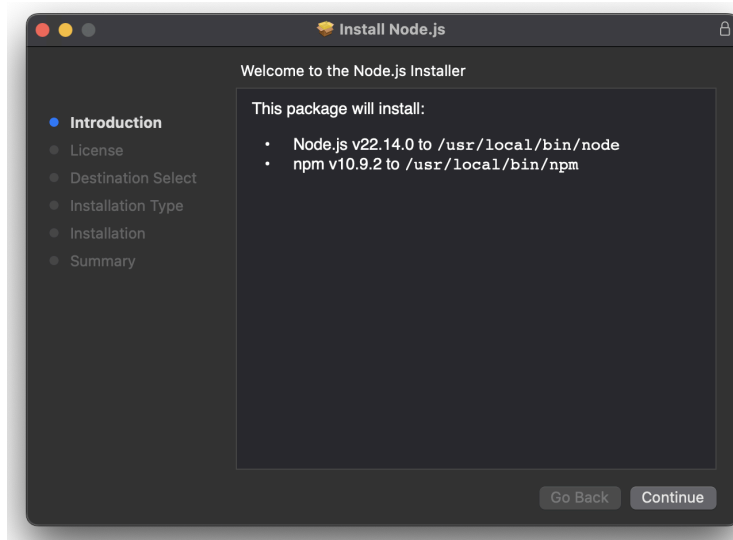
"nvm" is a cross-platform Node.js version manager. If you encounter any issues please visit [nvm's website](#)

Or get a prebuilt Node.js® for **macOS** running a **ARM64** architecture.

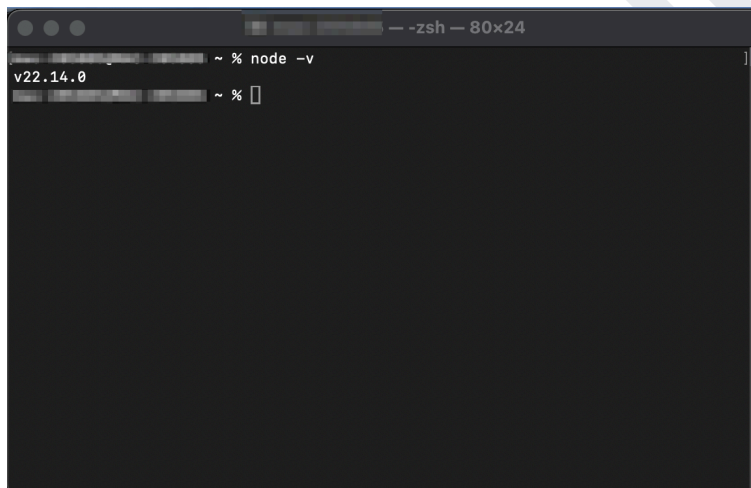
[macOS Installer \(.pkg\)](#) [Standalone Binary \(.gz\)](#)

Read the [changelog](#) for this version.
Read the [blog post](#) for this version.
Learn how to [verify](#) signed SHASUMS.
Looking for Node.js source? Download a signed [Node.js source](#) tarball.
Check out our [nightly](#) binaries or all [previous releases](#) or the [unofficial](#) binaries for other platforms.

- b. Jalankan file instalasi (biasanya berekstensi `.pkg`).
- c. Ikuti instruksi instalasi seperti biasa.



- d. Setelah selesai, buka Terminal dan ketik `node -v` untuk memastikan Node.js sudah terinstal.

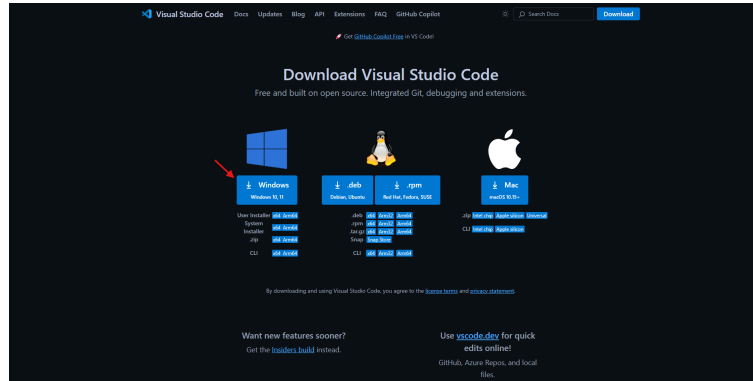


- e. Catatan: Kalau kamu sudah punya Homebrew di Mac, kamu juga bisa ketik `brew install node` di Terminal untuk install Node.js dengan cepat.

Cara Install VS Code di Windows dan Mac

1. Install VS Code di Windows:

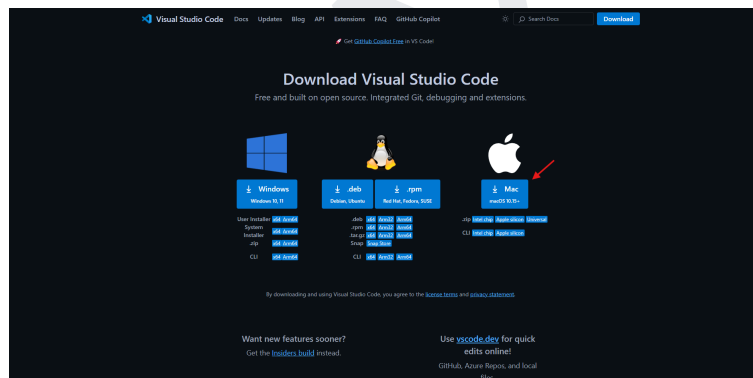
- a. Buka code.visualstudio.com dan download versi untuk Windows.



- b.
- c. Jalankan file instalasi (biasanya berekstensi .exe).
- d. Ikuti wizard instalasi, pastikan mencentang opsi “Add to PATH” saat instalasi.
- e. Setelah selesai, buka VS Code dari Start Menu atau Desktop.

2. Install VS Code di Mac:

- a. Buka code.visualstudio.com dan download versi untuk Mac.



- b. File yang terunduh biasanya berbentuk .zip. Ekstrak file tersebut.
- c. Pindahkan aplikasi **Visual Studio Code** ke folder **Applications**.
- d. Setelah selesai, buka VS Code dari Start Menu

Menampilkan Teks dengan console.log

console.log adalah cara termudah di JavaScript untuk menampilkan sesuatu di terminal. Kamu bisa pakai ini untuk melihat apa yang terjadi di kode yang kamu tulis.

Contoh Penggunaan:

Buat file bernama app.js dan isi dengan kode ini:

```
console.log("Hello, World!"); // Tulis teks apa pun yang mau ditampilkan
```

Cara Menjalankan di Terminal:

1. Buka terminal di folder tempat file app.js berada.
2. Ketik perintah ini:

```
node app.js
```

3. Hasil yang muncul:

```
Hello, World!
```


Operasi Dasar Matematika di JavaScript

Di JavaScript, kamu bisa melakukan berbagai operasi matematika seperti penjumlahan, pengurangan, perkalian, dan pembagian. Operasi ini disebut **arithmetic expressions**.

Berikut simbol-simbol dasar yang digunakan:

- **Penjumlahan (+):** Menambah dua angka.
- **Pengurangan (-):** Mengurangi angka.
- **Perkalian (*):** Mengalikan angka.
- **Pembagian (/):** Membagi angka.

Contoh Penggunaan di Kode:

Buat file math.js dan isi dengan kode berikut:

```
console.log(5 + 3); // Menampilkan 8
console.log(10 - 4); // Menampilkan 6
console.log(2 * 3); // Menampilkan 6
console.log(12 / 4); // Menampilkan 3
```

Cara Menjalankan di Terminal:

1. Buka terminal di folder tempat file math.js berada.
2. Ketik perintah ini:

```
node math.js
```

3. Hasil yang muncul di terminal:

```
8
6
6
3
```

Catatan Penting:

Operasi ini hanya bekerja untuk angka. Kalau kamu mencoba menjumlahkan teks (string) dengan angka, hasilnya mungkin tidak seperti yang kamu harapkan.

Apa Itu Variabel di JavaScript?

Variabel adalah tempat untuk menyimpan data di JavaScript. Bayangkan variabel seperti kotak yang bisa kamu beri nama dan isi dengan apa pun, seperti angka, teks, atau data lainnya.

Cara Membuat Variabel:

Gunakan kata kunci `let` atau `const` untuk membuat variabel.

- **let:** Variabel yang isinya bisa diubah.
- **const:** Variabel yang isinya tidak bisa diubah (konstan)

Contoh Penggunaan:

```
let nama = "Kevin"; // Membuat variabel nama dengan isi "Kevin"
console.log(nama); // Menampilkan: Kevin

const umur = 20; // Membuat variabel umur dengan isi 20
console.log(umur); // Menampilkan: 20
```

Cara Kerja Variabel:

1. Beri nama variabel.
2. Isi variabel dengan data (angka, teks, dll.).
3. Gunakan `console.log` untuk melihat hasilnya di terminal.

Catatan:

- Gunakan nama variabel yang mudah dimengerti, seperti nama, umur, atau alamat.
- Jangan gunakan nama variabel yang sama dua kali dengan `let` dan `const`.

Pengambilan Keputusan dengan If-Else

Di JavaScript, **if-else** digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Kamu bisa memakai if-else untuk menentukan apa yang harus dilakukan oleh program, tergantung situasinya.

Struktur Dasar If-Else:

```
if (kondisi) {  
    // Kode yang dijalankan jika kondisi benar (true)  
} else {  
    // Kode yang dijalankan jika kondisi salah (false)  
}
```

Contoh Penggunaan:

```
let nilai = 75;  
  
if (nilai >= 70) {  
    console.log("Selamat, kamu lulus!"); // Dijalankan jika nilai >= 70  
} else {  
    console.log("Maaf, kamu harus mengulang."); // Dijalankan jika nilai <  
    70  
}
```

Penjelasan Contoh:

1. Program mengecek apakah nilai lebih besar atau sama dengan 70.
2. Jika iya (**true**), maka akan menampilkan "Selamat, kamu lulus!".
3. Jika tidak (**false**), maka akan menampilkan "Maaf, kamu harus mengulang.".

Catatan:

- Kondisi adalah ekspresi yang menghasilkan **true** atau **false**, seperti nilai `>= 70`.
- Kamu bisa menambahkan lebih banyak kondisi dengan **else if**, misalnya:

```
if (nilai >= 85) {  
    console.log("Nilai kamu A");  
} else if (nilai >= 70) {  
    console.log("Nilai kamu B");  
} else {  
    console.log("Nilai kamu C");  
}
```

Dengan if-else, kamu bisa membuat program yang lebih pintar dan dinamis.

Perulangan dengan Loop di JavaScript

Loop digunakan untuk menjalankan kode yang sama berulang kali. Ini sangat berguna ketika kamu ingin mengulangi sesuatu tanpa harus menulis kode berulang-ulang.

Jenis-Jenis Loop di JavaScript:

- **for:** Digunakan jika kamu tahu berapa kali loop akan dijalankan.
- **while:** Digunakan jika kamu tidak tahu pasti berapa kali loop akan berjalan, tapi ada kondisi tertentu.

Struktur Dasar For Loop:

```
for (let i = 0; i < 5; i++) {  
  // Kode di sini akan dijalankan 5 kali  
}
```

Contoh For Loop:

```
for (let i = 1; i <= 5; i++) {  
  console.log("Angka ke-" + i); // Menampilkan Angka ke-1, Angka ke-2,  
  dst.  
}
```

Struktur Dasar While Loop:

```
let i = 0;  
while (i < 5) {  
  // Kode di sini akan dijalankan selama kondisi true  
  i++;  
}
```

Contoh While Loop:

```
let i = 1;  
while (i <= 5) {  
  console.log("Nomor: " + i); // Menampilkan Nomor: 1, Nomor: 2, dst.  
  i++;  
}
```

Mengenal Fungsi di JavaScript

Fungsi (function) adalah blok kode yang dirancang untuk melakukan tugas tertentu. Dengan fungsi, kamu bisa mengelompokkan kode agar lebih rapi dan mudah digunakan kembali.

```
function namaFungsi() {  
    // Kode yang akan dijalankan  
}
```

Cara Memanggil Fungsi:

Untuk menjalankan kode di dalam fungsi, kamu perlu memanggилnya dengan menuliskan nama fungsi diikuti tanda kurung:

```
namaFungsi();
```

Contoh Fungsi Sederhana:

```
function sapa() {  
    console.log("Halo, selamat datang!");  
}  
  
sapa(); // Menampilkan: Halo, selamat datang!
```

Fungsi dengan Parameter:

Parameter adalah data yang bisa kamu kirim ke fungsi untuk digunakan di dalamnya.

```
function sapa(nama) {  
    console.log("Halo, " + nama + "!");  
}  
  
sapa("Kevin"); // Menampilkan: Halo, Kevin!  
sapa("Axel"); // Menampilkan: Halo, Axel!
```

Fungsi dengan Nilai Balik (Return):

Fungsi juga bisa mengembalikan nilai menggunakan kata kunci return.

```
function tambah(angka1, angka2) {  
    return angka1 + angka2;  
}
```

```
let hasil = tambah(5, 3);  
console.log(hasil); // Menampilkan: 8
```

Catatan:

- Gunakan fungsi untuk menyederhanakan kode, terutama jika ada tugas yang sering diulang.
- Nama fungsi sebaiknya mencerminkan tugasnya, seperti sapa, tambah, atau hitungUmur.

Kevin Axellino

Penutup

Selamat! Kamu telah menyelesaikan perjalanan untuk mempelajari dasar-dasar JavaScript dan Node.js. Materi dalam PDF ini dirancang untuk membangun fondasi pemahaman yang kuat bagi pemula, khususnya mahasiswa IT. Dengan mempelajari topik seperti variabel, perulangan, fungsi, hingga pengaturan lingkungan kerja, kamu kini memiliki kemampuan dasar untuk memulai eksplorasi lebih dalam ke dunia pemrograman.

Pemrograman adalah perjalanan belajar yang tidak ada habisnya. Semakin sering kamu berlatih dan mencoba hal baru, semakin mahir kamu dalam memecahkan masalah dengan kode. Jangan ragu untuk membuat proyek kecil-kecilan, seperti kalkulator sederhana atau aplikasi yang memanfaatkan perulangan dan fungsi.

Jika ada kesulitan, jangan khawatir! Komunitas pemrograman di luar sana sangat besar dan selalu siap membantu. Kamu bisa bergabung di forum, grup media sosial, atau bahkan bertanya pada teman-teman sekelasmu.

Teruslah belajar, bereksperimen, dan menikmati prosesnya. Semoga materi ini bermanfaat dalam perjalananmu sebagai programmer.

Selamat ngoding! 😊