

Node.js

Apa Itu Node.js?

- Node.js adalah runtime environment JavaScript yang bersifat open-source dan berbasis JavaScript V8 engine dari Google.
- Dirancang untuk menjalankan kode JavaScript di sisi server, bukan hanya di browser.
- Menggabungkan kemampuan bahasa JavaScript dengan operasi I/O non-blocking yang efisien.

Runtime environment: is a program that provides a software platform for applications to run.

Runtime vs Runtime Environment

Feature	Runtime	Runtime environment
What it is	The phase when a program is executing	The platform or system that allows a program to run
When it happens	After compilation	During runtime
Analogy	Cooking a dish	The kitchen and its tools
Example	JavaScript code running in the Chrome browser	Node.js script connecting to a database

Karakteristik Utama Node.js

- **Non-Bloking dan Event-Driven:** Operasi I/O non-blocking memungkinkan Node.js untuk menangani banyak koneksi secara efisien.
- **Server-Side:** Digunakan untuk pengembangan aplikasi server, seperti aplikasi web dan API.
- **Server Environment:** Didesain untuk menggantikan server tradisional, seperti Apache HTTP Server.

Non-blocking: Node.js tidak akan menunggu operasi I/O selesai, melainkan akan melanjutkan ke operasi berikutnya.

Event-driven: Node.js menggunakan event loop untuk menangani proses yang akan dieksekusi.

Karakteristik Utama Node.js

- **JavaScript Everywhere:** Memungkinkan penggunaan bahasa JavaScript di kedua sisi, server dan klien.
- **Modular dan Ekosistem Besar:** Memiliki ekosistem modul yang besar melalui npm (Node Package Manager).
- **Pemrograman Asinkron:** Cocok untuk aplikasi real-time dan berkinerja tinggi.

Contoh Penggunaan Node.js

- Aplikasi web server dengan framework seperti Express.js.
- Aplikasi real-time seperti chat dan game online.
- Aplikasi berbasis mikrokontroler dan IoT.
- Alat pengembangan (tools) seperti Vite dan Webpack.
- Aplikasi berbasis command line (CLI).

Mengapa Node.js?

- Performa Tinggi: Kemampuan operasi I/O non-blocking dan event-driven.
- Fleksibel: Cocok untuk berbagai jenis aplikasi, dari web hingga aplikasi real-time.
- Komunitas yang Kuat: Node.js memiliki komunitas pengembang yang besar yang terus berkembang.
- Ekosistem Modul: Dapat menggunakan ribuan modul pihak ketiga melalui `npm`.

Node Package Manager `npm`

- Package manager adalah tools yang digunakan untuk mengelola modul atau package yang digunakan dalam suatu proyek.
- `npm` menyediakan akses ke ribuan modul pihak ketiga yang dapat digunakan dalam proyek Node.js.
- `npm` juga dapat digunakan untuk mengelola proyek Node.js, seperti menginisialisasi proyek, menginstal modul, dan menjalankan skrip.
- `npm` telah disertakan dalam instalasi default Node.js.

Package Manager Alternatives

`npm` bukan satu-satunya package manager yang dapat digunakan dalam proyek Node.js. Ada package manager alternatif yang dapat digunakan, seperti:

- `yarn` (<https://yarnpkg.com/>)
- `pnpm` (<https://pnpm.io/>)

Node.js Project Initialization

- Inisialisasi proyek Node.js adalah langkah pertama dalam memulai pengembangan aplikasi Node.js.
- Gunakan perintah `npm init` untuk membuat berkas `package.json`.
- Berkas `package.json` berisi informasi proyek dan dependensi.