

NODE.JS

Workshop Pemrograman WEB

Yunia Ikawati

Teknik Informatika-PENS



Sub CPMK

Mahasiswa mampu:

- Memahami konsep dasar Node.js
- Memahami instalasi Node.js
- Mengelola package dengan npm
- Membuat aplikasi backend sederhana

Tujuan Pembelajaran

Setelah mempelajari materi ini, mahasiswa diharapkan:

- Memahami konsep runtime Node.js
- Menggunakan Node.js untuk membuat backend
- Mengembangkan aplikasi web sederhana

Pengertian Node.js

- Node.js adalah runtime JavaScript yang berjalan di sisi server.
 - Node.js bisa membuat aplikasi backend (server, API, dll.) menggunakan JavaScript.
- Menggunakan engine V8 dari Google Chrome.
 - proses eksekusi kode menjadi cepat dan efisien.
- Bersifat asynchronous dan event-driven.
 - Node.js dapat menangani banyak proses sekaligus tanpa harus menunggu satu proses selesai dulu.

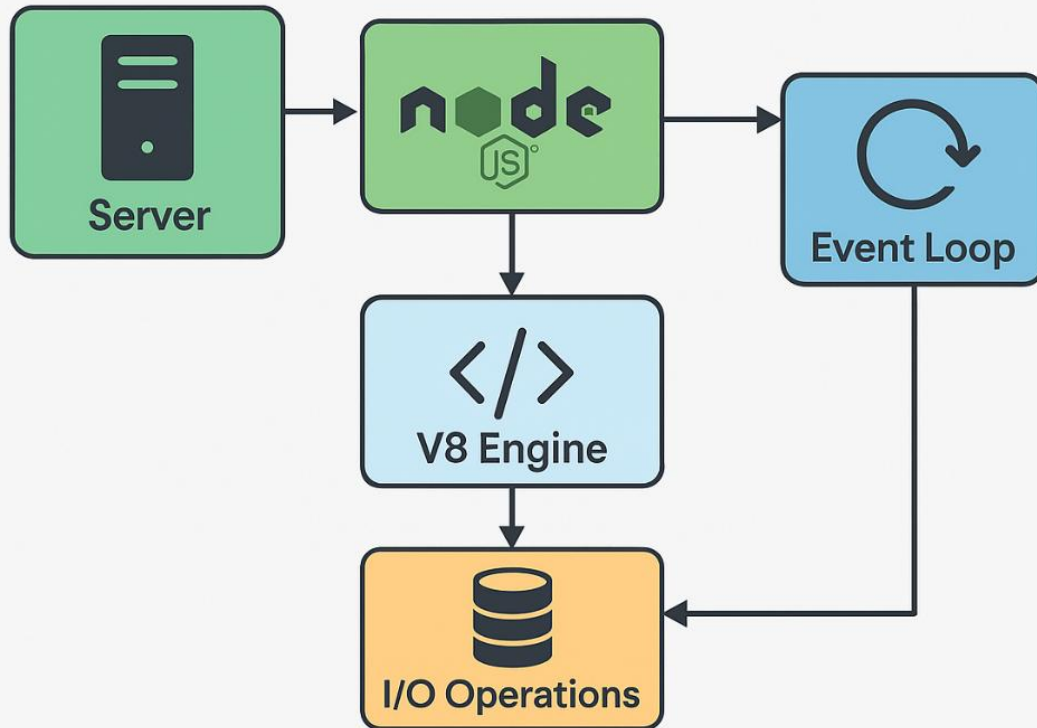




Karakteristik Node.js

1. Non-blocking I/O
 - Node.js tidak menunggu satu proses selesai sebelum menjalankan proses lain.
2. Event-driven
 - Node.js bekerja berdasarkan event (kejadian).
3. Single-threaded
 - Node.js menggunakan satu thread utama untuk menjalankan program.
4. Cepat dan ringan
 - Karena menggunakan engine V8 dan konsep non-blocking, Node.js memiliki performa tinggi dan konsumsi resource yang relatif kecil.

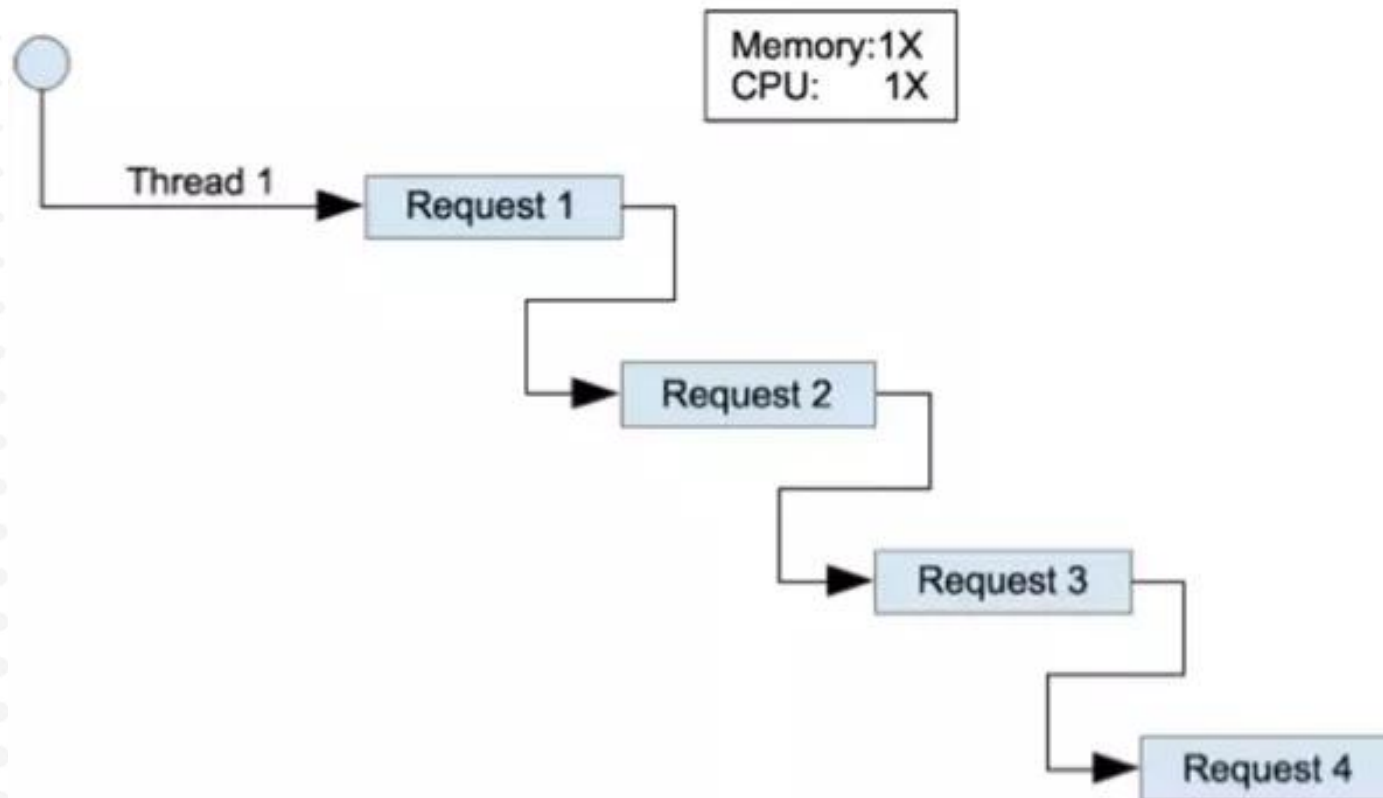
Arsitektur Node.js



1. Server menerima request
2. Node.js mengatur prosesnya
3. V8 Engine menjalankan kode JavaScript
4. Jika ada proses berat (I/O Operations seperti file, database), dikerjakan di background
5. Event Loop menunggu hasil I/O lalu menjalankan callback

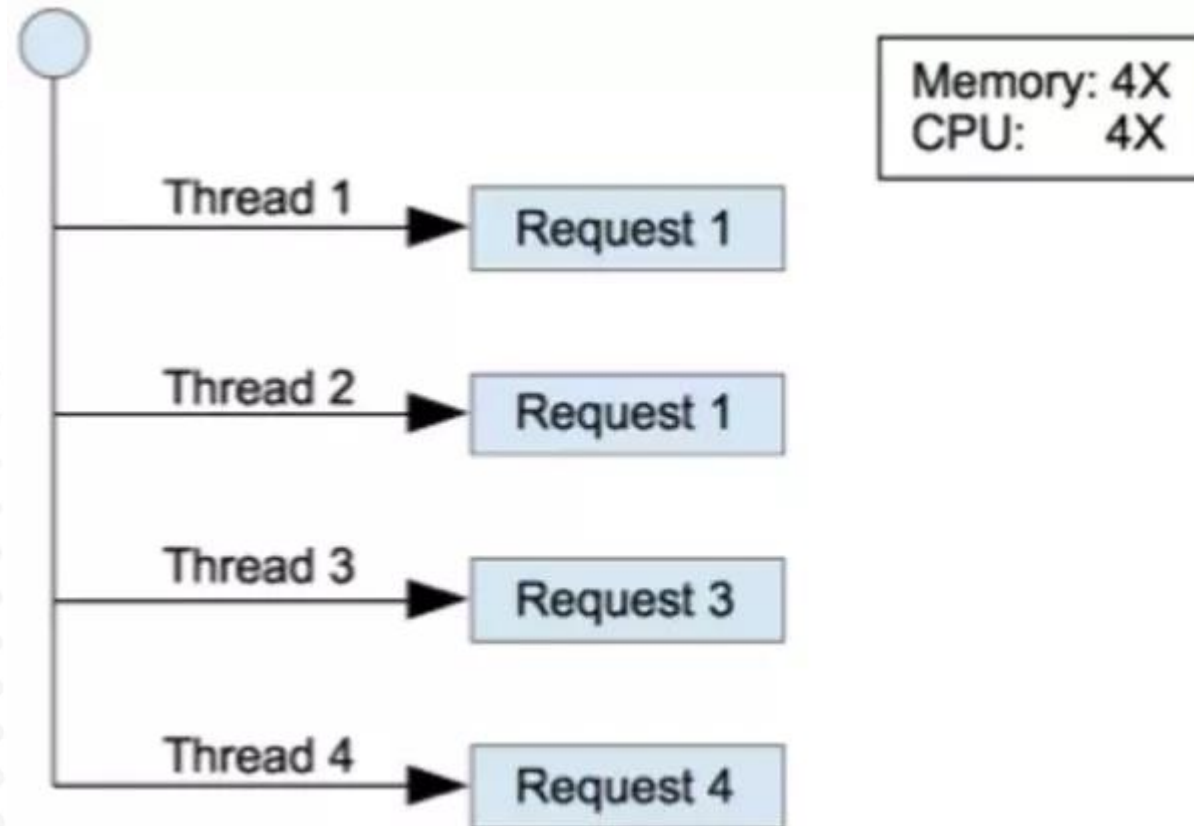
Single Threaded pada Node.js

Hanya 1 thread untuk menjalankan JavaScript



Multi Threaded pada PHP

Banyak thread berjalan bersamaan





Node.js VS PHP

1. PHP dan Node.js sebenarnya berbeda dalam hal konsep dasar.
 - PHP adalah sebuah bahasa pemrograman back-end murni.
 - Node.js adalah sebuah *platform*. (bahasa pemrograman **JavaScript**)
2. Node.js menggunakan konsep **non-blocking I/O model**. Konsep ini membuat pemrosesan di Node.js lebih efisien karena tidak *terkunci* ketika sebuah proses sedang berjalan.
3. Node.js juga menggunakan konsep **single threaded** (hanya ada 1 proses yang berjalan sepanjang waktu). Ini berbeda dengan PHP yang menggunakan konsep **multi threaded**.

Penggunaan PHP vs Node.js

A. PHP

- untuk web yang butuh pemrosesan besar, konsep *multi thread* dengan PHP akan lebih baik. PHP 7

B. Node.js

- penggunaan terutama website yang terus-menerus realtime dan butuh banyak JavaScript.
- Menggunakan konsep single thread.
- Contoh: aplikasi Google seperti **Gmail, Google Docs, Google Analytics** itu full JavaScript, chat, streaming, atau API dengan banyak pengguna.

Instalasi Node.js

1. Unduh Node.js dari: <https://nodejs.org>
2. Pilih LTS (Long Term Support) untuk stabilitas.
3. Referensi Install Node.js.
<https://www.youtube.com/watch?v=nzMK3Z1Y5NA>
4. Jalankan installer dan ikuti petunjuk.
5. Verifikasi di terminal:

```
node -v  
npm -v
```



NPM

- **NPM (Node Package Manager)** adalah manajer paket untuk Node.js
- Digunakan untuk:
 - Mengelola dependensi proyek.
 - Menjalankan script.
 - Menginstall pustaka dari komunitas.
- Contoh:
 - `npm init`
 - `npm install express`

Membuat Proyek Node.js

1. Buat folder proyek di terminal:

```
mkdir proyek-node && cd proyek-node  
npm init -y
```

2. Buat file **index.js**

3. Jalankan dengan:

```
node index.js
```

Contoh Sederhana Node.js

```
const http = require('http');

const server = http.createServer((req, res) => {
  // Set header
  res.writeHead(200, { 'Content-Type': 'text/plain' });

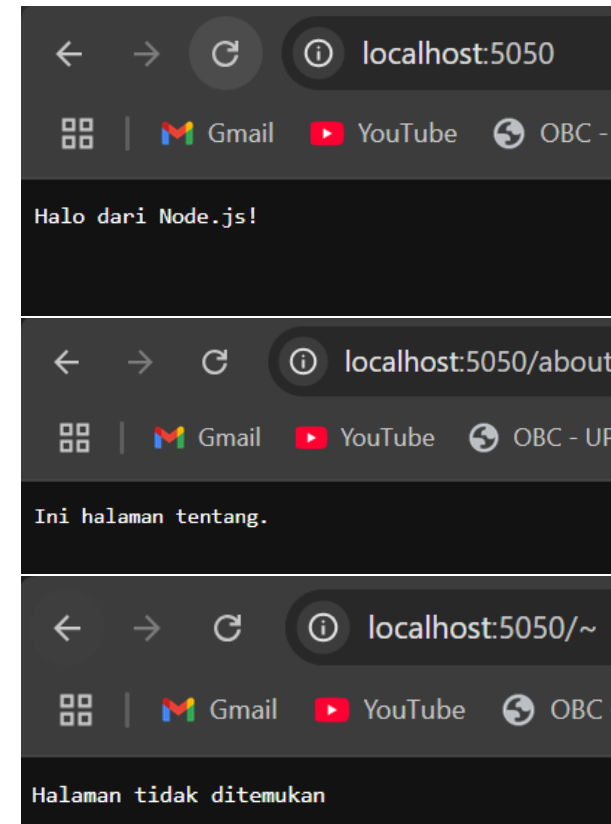
  // Cek URL
  if (req.url === '/') {
    res.end('Halo dari Node.js!');
  } else if (req.url === '/about') {
    res.end('Ini halaman tentang.');
```

```
  } else {
    res.end('Halaman tidak ditemukan');
  }
});

server.listen(5050, () => {
  console.log('Server berjalan di http://localhost:5050');
```

```
});
```

Buka browser: <http://localhost:5050>





Penerapan Node.js di Dunia Nyata

- Backend API (Express.js)
- Aplikasi Chat (Socket.io)
- Realtime Dashboard
- Microservices & Serverless
- Command-line tools

Kesimpulan

- Node.js = runtime JavaScript modern untuk backend.
- Instalasi mudah, banyak pustaka lewat npm.
- Cocok untuk aplikasi cepat, realtime, dan skalabel.



Latihan

1. Praktekkan pembuatan program untuk cara Handle Form Input di Node.js sesuai web berikut: <https://ngide.net/belajar-nodejs-form>
2. Selanjutnya praktekkan untuk membuat web sederhana dengan Node.js (modifikasi web sederhana tersebut). <https://ngide.net/belajar-nodejs-website>
3. Buatlah Sistem Login & Register Menggunakan Node.js terkoneksi ke database mongodb.(bisa dimodifikasi dari sisi tampilan).
<https://dev.to/tienbku/node-js-express-login-example-with-mongodb-2ofc>
4. Buatlah tampilan web sederhana dengan Menggunakan Node.js terkoneksi ke database mongodb sesuai kreasi masing-masing mahasiswa.
5. Buat laporan dan tuliskan codingan dan capture masing-masing prosesnya.
6. Yang dikumpulakn di ethol, seluruh file (node.js dan laporan) dalam bentuk .rar atau .zip.
7. Referensi lain bisa melihat di **we3school**