

LAPORAN PRAKTIKUM
JOB SHEET 1 - PENGANTAR NODEJS
PRAKTIKUM PEMROGRAMAN JARINGAN

Dosen Pengampu : Randi Proska Sandra, S.Pd, M.Sc



Disusun Oleh:

Ridho Hamdani Putra

23343052

202513430088 (INF1.62.5009)

PROGRAM STUDI INFORMATIKA
DEPARTEMEN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG

2025

A. TUJUAN

1. Mampu memahami dan menjelaskan konsep pemrograman berbasis jaringan dan prinsip dasar komunikasi antara klien dan server melalui jaringan
2. Mengetahui Lingkungan Node.js untuk Pemrograman berbasis Jaringan
3. Mampu menggambarkan dan mengidentifikasi peran Node.js dalam pengembangan aplikasi berbasis jaringan serta keunggulan Node.js dalam mengatasi tugas-tugas pemrograman jaringan
4. Mampu menginstall Node.js pada perangkat komputer masing-masing

B. HARDWARE & SOFTWARE

1. PC (Personal Computer) dengan akses Internet
2. Visual Studio Code
3. Node.js

C. URAIAN MATERI

A. Konsep Pemrograman Berbasis Jaringan

Pemrograman jaringan merupakan subdisiplin dalam ilmu komputer yang terfokus pada pengembangan aplikasi perangkat lunak yang memungkinkan interaksi, pertukaran data, dan komunikasi antara perangkat-perangkat yang terhubung dalam jaringan komputer. Pada intinya, pemrograman jaringan mencakup pendekatan teknis dan metodologi yang bertujuan untuk merancang, mengimplementasikan, dan memelihara aplikasi yang mampu berfungsi dalam lingkungan jaringan, termasuk jaringan lokal (LAN) maupun jaringan luas (WAN). Beberapa konsep utama dalam pemrograman berbasis jaringan meliputi:

1. **Arsitektur Klien-Pelayan (Client-Server Architecture):** Merupakan paradigma dasar yang mengilhami pemrograman berbasis jaringan. Model ini mendasarkan interaksi pada dua peran utama, yaitu klien (client) dan pelayan (server). Klien mengirimkan permintaan terhadap pelayan, yang kemudian merespons dengan memberikan layanan atau informasi yang dikehendaki. Arsitektur ini membentuk fondasi bagi aplikasi-aplikasi jaringan modern dan memfasilitasi distribusi tugas serta pemanfaatan sumber daya secara efisien.

2. **Protokol Jaringan:** Protokol merupakan sekumpulan aturan dan konvensi yang mengatur pertukaran data dan informasi di antara entitas-entitas dalam jaringan. Contoh nyata meliputi protokol HTTP (Hypertext Transfer Protocol) untuk komunikasi web dan protokol SMTP (Simple Mail Transfer Protocol) untuk pengiriman email. Protokol ini membentuk normatif pedoman yang memungkinkan komunikasi yang konsisten dan terstandarisasi dalam skenario jaringan.
3. **Socket:** Soket adalah antarmuka pemrograman yang menyediakan titik akses bagi aplikasi untuk terhubung dan berinteraksi melalui jaringan. Dalam ranah pemrograman berbasis jaringan, soket memiliki varian berdasarkan jenis koneksi yang dibentuk, seperti soket berbasis aliran (stream socket) untuk komunikasi berkelanjutan yang diimplementasikan dalam protokol TCP, dan soket berbasis datagram (datagram socket) yang mendukung pertukaran pesan berukuran terbatas tanpa mempertahankan koneksi seperti pada protokol UDP.
4. **Pemrograman Sinkron dan Asinkron:** Pemrograman sinkron mengacu pada pendekatan di mana eksekusi program menunggu respon dari operasi jaringan sebelum melanjutkan. Sebaliknya, pemrograman asinkron memungkinkan aplikasi untuk melanjutkan eksekusi tanpa menunggu hasil operasi jaringan secara segera. Keputusan dalam memilih antara kedua pendekatan ini akan mempertimbangkan efisiensi eksekusi, manajemen sumber daya, dan responsivitas aplikasi.
5. **Pemrograman Web:** Merupakan cabang yang signifikan dalam pemrograman berbasis jaringan, fokusnya adalah pada pengembangan aplikasi web. Aplikasi web beroperasi di server dan dapat diakses oleh pengguna melalui browser. Teknologi seperti HTML, CSS, JavaScript, serta bahasa pemrograman sisi server seperti PHP, Python, atau Node.js memainkan peran sentral dalam pengembangan aplikasi web yang dinamis dan responsif.
6. **Pemrograman Jaringan pada Perangkat Bergerak:** Dalam konteks perangkat bergerak, pemrograman jaringan memiliki relevansi yang signifikan. Aplikasi mobile sering kali bergantung pada koneksi jaringan untuk memperoleh dan menyampaikan informasi. Teknologi seperti REST API (Representational State Transfer Application Programming Interface) memfasilitasi komunikasi antara aplikasi mobile dan server, yang memungkinkan pertukaran data yang efisien.

Pemrograman berbasis jaringan membuka peluang bagi pengembangan beragam jenis aplikasi, termasuk aplikasi web, aplikasi seluler, permainan daring, layanan berbasis awan, dan lainnya. Semua ini bergantung pada kemampuan perangkat lunak untuk mengimplementasikan interaksi yang efektif dan handal melalui medium jaringan yang semakin kompleks dan diversifikasi.

Dalam sebagian besar aplikasi berbasis jaringan, umumnya terjadi proses komunikasi antara klien dan server.

Komunikasi antara klien dan server dalam pemrograman berbasis jaringan melibatkan serangkaian langkah terstruktur. Klien mengirimkan permintaan kepada server menggunakan protokol komunikasi seperti HTTP. Server menerima permintaan tersebut, memprosesnya sesuai dengan jenis dan isi permintaan, lalu mengirimkan respons kembali ke klien. Respons ini berisi informasi status, data yang diminta, atau pesan yang menggambarkan hasil tindakan server. Proses ini memungkinkan pertukaran data dan interaksi antara kedua entitas dalam konteks aplikasi jaringan.

B. Node.js

Node.js adalah lingkungan runtime yang memungkinkan eksekusi kode JavaScript di sisi server, melengkapi paradigma pemrograman berbasis jaringan dengan pendekatan asinkron dan event-driven. Keberadaan Node.js memberi dampak yang signifikan dalam memfasilitasi implementasi aplikasi jaringan yang responsif dan skalabel. Korelasi Node.js dengan pemrograman berbasis jaringan muncul melalui konseptualisasi yang kuat terkait asinkronisitas dan manajemen event dalam konteks operasi jaringan. Node.js merangkul pemrograman asinkron dengan pendekatan non-blokir (non-blocking), di mana operasi yang membutuhkan waktu dapat dilakukan tanpa memblokir eksekusi kode lain. Ini sangat relevan dalam mengatasi operasi jaringan yang mungkin memiliki latensi, seperti permintaan jaringan atau akses ke penyimpanan.

Node.js menggunakan model event loop yang efisien untuk mengatasi pemrograman asinkron. Pendekatan ini memungkinkan pengelolaan beberapa operasi secara bersamaan tanpa menghambat responsivitas aplikasi. Keunggulan ini terutama bermanfaat dalam pengembangan aplikasi jaringan yang menghadapi

keterlambatan jaringan, serta situasi di mana skalabilitas dan responsivitas menjadi imperatif.

Dalam ranah pemrograman jaringan, Node.js menyediakan modul bawaan yang mendukung pembangunan server dan komunikasi jaringan. Modul `'http'` dan `'https'` memungkinkan implementasi server HTTP dan HTTPS, sementara modul `'net'` mengizinkan pembangunan server TCP atau UDP kustom. Ekosistem NPM (node package manager) sebagai repositori pustaka dan modul pihak ketiga juga memperkaya Node.js dengan beragam solusi untuk tugas-tugas pemrograman jaringan. Keunggulan Node.js meliputi:

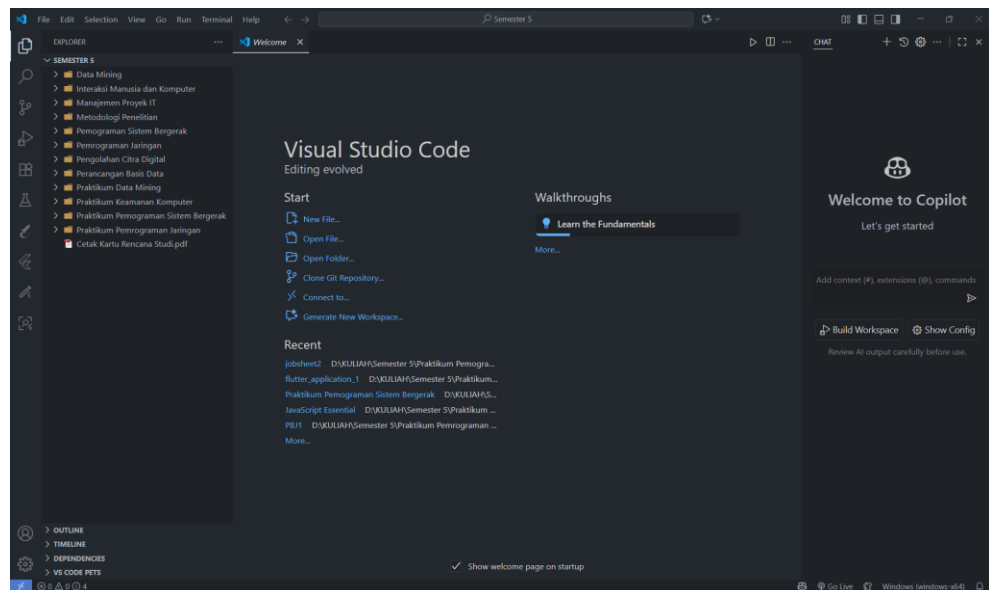
- 1. Efisiensi Skalabilitas (Efficient Scalability):** Dukungan terhadap operasi asinkron dan manajemen sumber daya yang cermat menjadikan Node.js cocok untuk membangun aplikasi jaringan yang dapat tumbuh dan skalabel dengan efisiensi.
- 2. Responsivitas Tinggi:** Model event loop yang efisien dalam Node.js memastikan bahwa aplikasi tetap responsif bahkan saat berurusan dengan operasi I/O yang memakan waktu.
- 3. Dukungan Multiplatform (Cross-Platform Support):** Node.js berjalan pada berbagai platform, memungkinkan pengembang untuk membuat aplikasi jaringan yang dapat beroperasi lintas platform dengan lebih mudah.
- 4. Integrasi Kode client-side dan server-side:** Konsistensi bahasa JavaScript di antara lingkungan klien dan server memungkinkan pengembang untuk memanfaatkan pengetahuan dan kode yang sudah ada.
- 5. Ekosistem NPM yang Kaya:** Ketersediaan beragam modul dan pustaka melalui ekosistem NPM memberikan akses ke solusi jaringan yang matang dan teruji.

Dengan demikian, Node.js mengemuka sebagai pilar fundamental dalam pemrograman berbasis jaringan, yang mempromosikan model pengembangan aplikasi yang responsif, efisien dalam sumber daya, dan dapat diandalkan dalam menghadapi tantangan kompleks dalam lingkungan jaringan modern.

D. LATIHAN

a. Instalasi Visual Studio Code

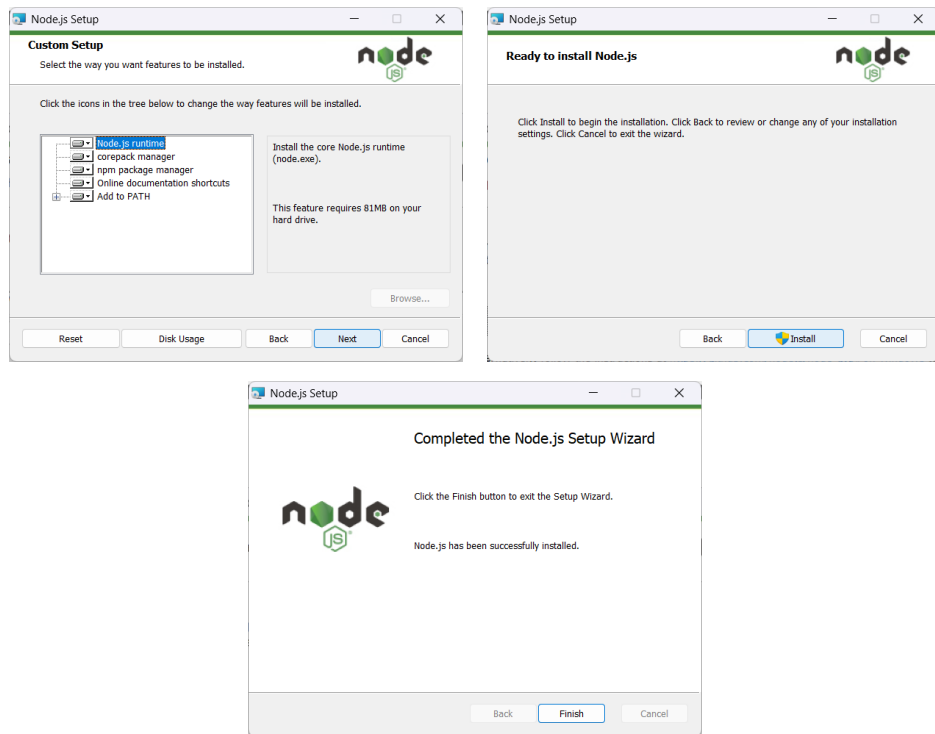
1. Silakandownload visual studio code melaluilink berikut ini
<https://code.visualstudio.com/>
2. Setelah diunduh, jalankan installer (VSCodeUserSetup-{versi}.exe). Ini hanya akan memakan waktu sebentar.
3. Secara default, VS Code diinstal pada direktori
C:\Users\{Username}\AppData\Local\Programs\Microsoft VS Code



VS Code sudah terinstall

b. Instalasi Node.js

1. Kunjungi situs resmi Node.js di <https://nodejs.org/> dan unduh versi Node.js yang sesuai dengan sistem operasi Anda (Windows, macOS, atau Linux).



4. Setelah mengkonfirmasi pilihan-pilihan instalasi, tungguhlah sampai proses instalasi selesai.
5. Setelah instalasi selesai, buka terminal atau command prompt (tergantung pada sistem operasi) dan ketikkan perintah `node -v`. Hal ini akan mengeluarkan versi Node.js yang telah diinstal. Selain itu, juga dapat menguji dengan menjalankan perintah `npm -v` untuk memastikan bahwa npm (Node Package Manager) juga diinstal.

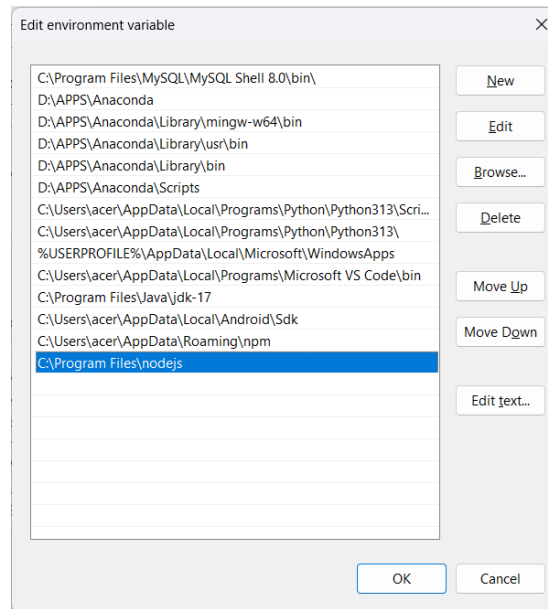
```
Command Prompt
Microsoft Windows [Version 10.0.26100.5074]
(c) Microsoft Corporation. All rights reserved.

C:\Users\acer>node -v
v22.19.0

C:\Users\acer>npm -v
10.9.3
```

6. Jika node.js belum terdeteksi silakan set PATH node.js pada environment variables. Anda bisa melakukannya dengan perintah berikut pada command prompt. (pastkan lokasi direktori node.js sesuai dengan folder instalasi)
`SET PATH=C:\Program Files\Nodejs;%PATH%`

7. Atau membuka setting environment variables dengan cara membuka Control Panel -> System and Security -> System -> Advanced System Settings -> Environment Variables
8. Edit variabel path pada user variables dan system variables dan tambahkan direktori instalasi node.js C:\Program Files\nodejs



c. Tes Perintah Node.js pada Command Prompt

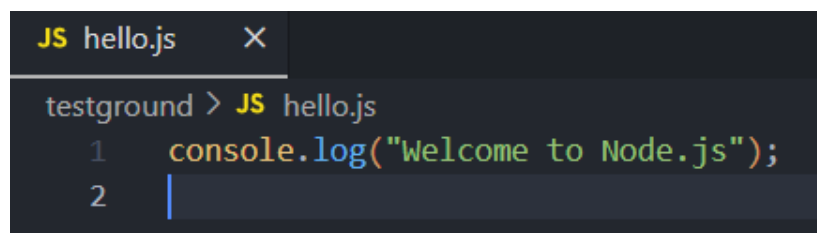
1. Silakan masukan perintah berikut pada command prompt untuk uji coba penggunaan node.js
2. Silakan gunakan perintah `process.exit()` untuk keluar

```
Microsoft Windows [Version 10.0.26100.5074]
(c) Microsoft Corporation. All rights reserved.

C:\Users\acer>node
Welcome to Node.js v22.19.0.
Type ".help" for more information.
> 12 * 13
156
> 'Ridho'.toLowerCase()
'ridho'
> 'Ridho'.toUpperCase()
'RIDHO'
> global
<ref *1> Object [global] {
  global: [Circular *1],
  clearImmediate: [Function: clearImmediate],
  setImmediate: [Function: setImmediate] {
    [Symbol(nodejs.util.promisify.custom)]: [Getter]
  },
  clearInterval: [Function: clearInterval],
  clearTimeout: [Function: clearTimeout],
  setInterval: [Function: setInterval],
  setTimeout: [Function: setTimeout] {
    [Symbol(nodejs.util.promisify.custom)]: [Getter]
  },
  queueMicrotask: [Function: queueMicrotask],
  structuredClone: [Function: structuredClone],
  atob: [Function: atob],
  btoa: [Function: btoa],
  performance: [Getter/Setter],
  fetch: [Function: fetch],
  navigator: [Getter],
  crypto: [Getter]
}
> process.exit()
```

d. Program Hellow-World

1. Silakan buat sebuah folder dengan nama PBJ1 atau sesuai keinginan anda dan simpan pada direktori yang anda inginkan. Jangan memberikan spasi pada nama folder anda.
2. Bukalah visual studio code dan kemudian bukalah folder tersebut pada visual studio code
3. Setelah terbuka buatlah folder baru pada folder tersebut melalui visual studio code dengan nama testground dan dalam folder tersebut
4. Buatlah file baru dengan nama hello.js dalam folder yang anda buat pada Langkah 3 dan cobalah ketikkan kode berikut ini



```
JS hello.js X
testground > JS hello.js
1 console.log("Welcome to Node.js");
2 |
```

5. Jalankan file tersebut dan lihat bagaimana hasilnya. Anda juga dapat menjalankan program tersebut melalui terminal pada visual studio code dengan memasukan perintah berikut

PS C:\Users\{NamaPengguna}\Desktop\{namaFolder}\testground> node hello.js



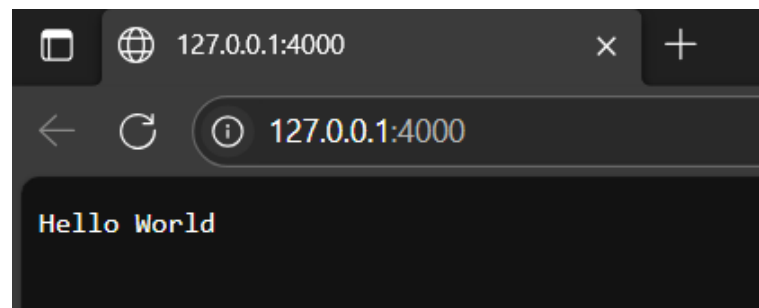
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 5\Praktikum Pemrograman Jaringan\PBJ1> cd testground
PS D:\KULIAH\Semester 5\Praktikum Pemrograman Jaringan\PBJ1\testground> node hello.js
Welcome to Node.js
PS D:\KULIAH\Semester 5\Praktikum Pemrograman Jaringan\PBJ1\testground> |
```

6. Buatlah sebuah file baru lagi dengan nama hello-world.js pada folder yang sama dan cobalah ketikkan kode berikut ini

```
JS hello-world.js X
testground > JS hello-world.js > ...
1  const http = require("http");
2
3  const hostname = "127.0.0.1";
4  const port = 4000;
5
6  const server = http.createServer((req, res) => {
7    res.statusCode = 200;
8    res.setHeader("Content-Type", "text/plain");
9    res.end("Hello World");
10 });
11
12 server.listen(port, hostname, () => {
13   console.log(`Server running at http://${hostname}:${port}`);
14 });
15

PS D:\KULIAH\Semester 5\Praktikum Pemrograman Jaringan\PBJ1\testground> node hello-world.js
Server running at http://127.0.0.1:4000
```

7. Jalankan program tersebut dan akan muncul informasi Server running at <http://127.0.0.1:4000/>
8. Bukalah url tersebut melalui browser anda dan akan muncul tampilan Hello-World



E. KESIMPULAN

Melalui praktikum Jobsheet 1 ini dapat disimpulkan bahwa pemrograman berbasis jaringan merupakan dasar penting dalam pengembangan aplikasi modern yang memungkinkan komunikasi antara klien dan server melalui jaringan. Node.js hadir sebagai lingkungan runtime JavaScript di sisi server yang mendukung pemrograman asinkron, event-driven, serta memiliki efisiensi tinggi dalam menangani proses I/O. Dengan melakukan instalasi Node.js dan Visual Studio Code, praktikum ini mampu memperlihatkan cara kerja dasar Node.js, menguji perintah dasar pada terminal, hingga

membuat program sederhana seperti Hello World. Hal ini membuktikan bahwa Node.js sangat relevan dan bermanfaat untuk membangun aplikasi berbasis jaringan yang responsif, efisien, serta mudah dikembangkan lintas platform.