Modul 01

Pengantar Pemrograman Berbasis Jaringan

TUJUAN PEMBELAJARAN

1. Mampu memahami dan menjelaskan konsep pemrograman berbasis jaringan dan prinsip dasar komunikasi antara klien dan server melalui jaringan
2. Mengenal 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

Hardware & Software

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

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).

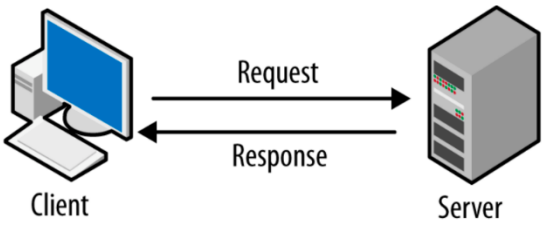
Beberapa konsep utama dalam pemrograman berbasis jaringan meliputi:

1. **Arsitektur Klien-Pelayan (Client-Server Architecture):** Merupakan paradigmatis 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.



Gambar 1. Ilustrasi Komunikasi antara *client* 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 meliput:

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.

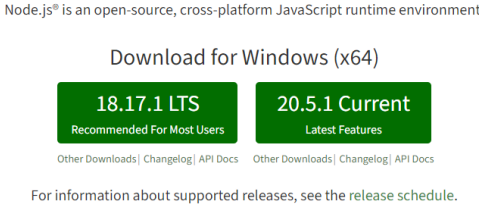
LATIHAN

a. Instalasi Visual Studio Code

1. Silakan download visual studio code melalui link 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

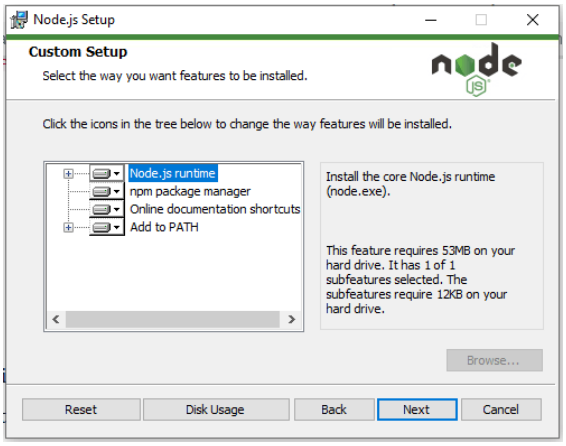
b. Instalasi Node.js

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).



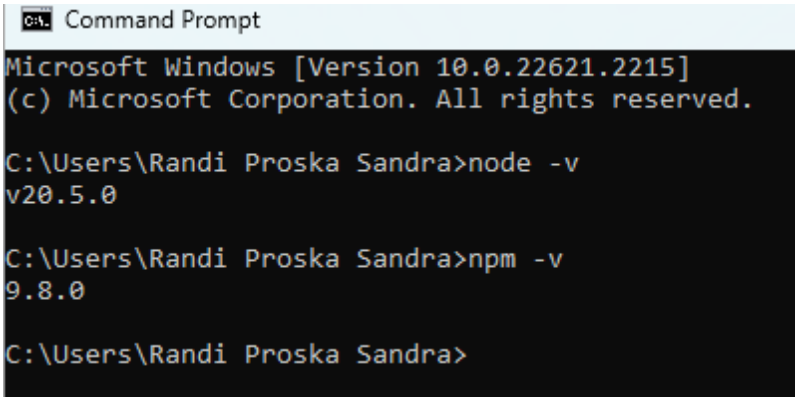
Gambar 2. Halaman download Node.js

2. Setelah unduhan selesai, jalankan berkas instalasi yang telah diunduh. Pada Windows, biasanya akan berupa berkas dengan ekstensi **.msi**, dan pada macOS berkas dengan ekstensi **.pkg**. Di Linux, mungkin perlu menggunakan manajer paket atau unduh dari sumber yang sesuai.
3. Ikuti langkah-langkah yang ada dalam proses instalasi. Hal ini dapat mencakup pemilihan direktori instalasi, persetujuan terhadap syarat dan ketentuan, dan konfigurasi lainnya.



Gambar 2. Custom setup Node.js

4. Setelah mengkonfirmasi pilihan-pilihan instalasi, tunggulah 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.

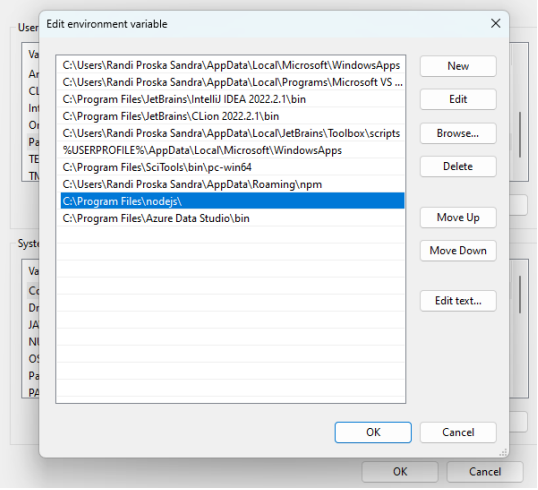


Gambar 3. Verifikasi instalasi Node.js

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`



Gambar 4. Setting Environment Variables Node.js

c. Tes Perintah Node.js pada Command Prompt

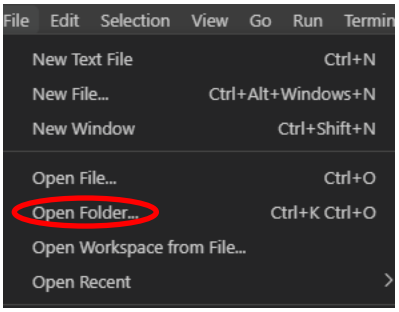
1. Silakan masukan perintah berikut pada command prompt untuk uji coba penggunaan node.js

Gambar 5. Uji coba perintah-perintah terkait Node.js pada command prompt

2. Silakan gunakan perintah `process.exit()` untuk keluar

d. Program Hello-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



Gambar 6. Open folder Visual Studio Code

3. Setelah terbuka buatlah folder baru pada folder tersebut melalui visual studio code dengan nama **testground** dan dalam folder tersebut

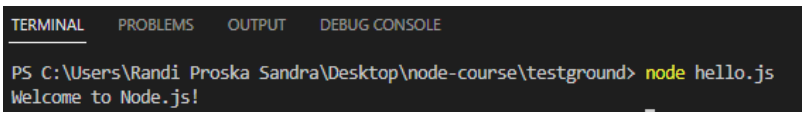
Gambar 7. Menambahkan folder baru pada visual studio code

4. Buatlah file baru dengan nama **hello.js** dalam folder yang anda buat pada Langkah 3 dan cobalah ketikan kode berikut ini

```
console.log('Welcome to Node.js!')
```

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
```



Gambar 8. Running program melalui terminal

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

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

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Gambar 9. Program hello-world

7. Jalankan program tersebut dan akan muncul informasi `Server running at http://127.0.0.1:3000/`
8. Bukalah url tersebut melalui browser anda dan akan muncul tampilan `Hello-World`

TUGAS

1. Jelaskan tentang JavaScript Engine, V8 dan bagaimana perbedaan antara Node.js dan JavaScript Engine pada browser Google Chrome?

REFERENCES

1. Kurose, J., & Ross, K. (2016, October 5). *Computer Networking: a Top-Down Approach, Global Edition*.
2. Stevens, R. W., Fenner, B., Rudoff, A. M., & Stevens, W. R. (2003, November 14). *UNIX Network Programming: The Sockets Networking API*. https://doi.org/10.1604/9780131411555
3. Pasquali, S., & Faaborg, K. (2017, December 29). Mastering Node.js: Build Robust and Scalable Real-Time Server-Side Web Applications Efficiently.
4. About | Node.js. (n.d.). Node.js. https://nodejs.org/en/about
5. *Beej's Guide to Network Programming*. (n.d.). Beej's Guide to Network Programming. https://beej.us/guide/bgnet/