

ILT 4: Backend

Pengenalan Back-End s.d. Deploy Web Services

Aturan **Instructor/Expert Led Session**



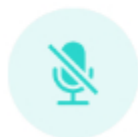
Fokus Penuh



Always-on Camera



Raise hand atau Chat jika ingin bertanya



Mute jika sedang tidak berbicara

Porsi Skor Penilaian (Tech)

Jenis Aktivitas	Porsi Skor
Rata-rata Kuis pada ILT	20%
Aktivitas selama ILT (bertanya, menjawab, membantu diskusi)	10%
Aktivitas Forum Diskusi*	10%
Rata-rata Exam/Submission Project	60%

*Jika ditemukan peserta spamming pada forum diskusi, maka nilai terbaik yang bisa didapat adalah 60 (dari skala 100)

Materi

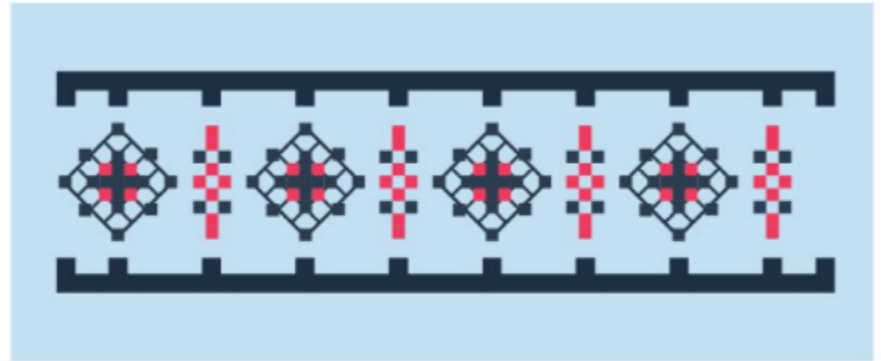
Pengenalan **Back-End**

- Apa itu Back-End?
- Web Server dan Web Service
- Komunikasi Client-Server
- REST Web Service
 - Format Request dan Response
 - HTTP Verbs
 - Response Code
 - URL Design

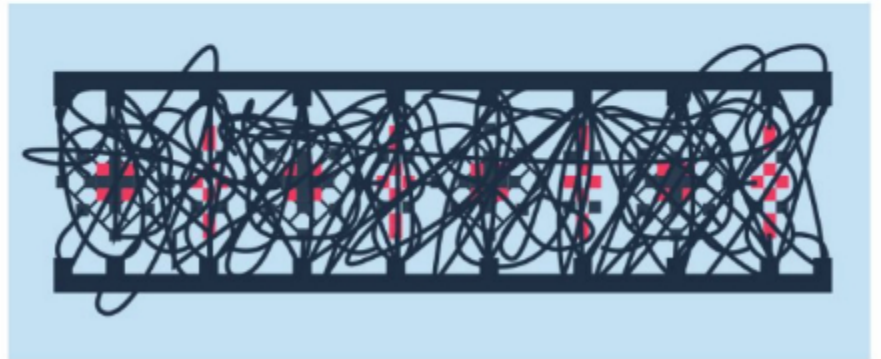
Apa itu Back-End?

- Front-End: Bagian dari aplikasi yang terlihat dan digunakan langsung oleh pengguna.
- Back-End: Bagian dari aplikasi yang menyediakan kebutuhan pengguna.

Front-End



Back-End



Web Server dan Web Service

- Web Server: Server yang dapat menjalankan program dan dapat diakses melalui internet atau intranet.
- Web Service: Program yang dijalankan di web server agar kebutuhan bisnis terpenuhi.



Image: Freepik.com

Komunikasi

Client-Server

- Salah satu protokol untuk berinteraksi dengan web server adalah HTTP/HTTPS
- HTTP/HTTPS menggunakan pola request-response
- Untuk mendapatkan sesuatu (response), kita perlu melakukan permintaan terlebih dahulu (request)



REST

Web Service

- REST : Representational State Transfer.
- REST API : Menjembatani antara sistem yang berbeda.
- RESTful API : Web service yang menerapkan arsitektur REST.

Format

Request dan Response

- REST API seringkali menggunakan JSON sebagai format request dan response.
- JSON memiliki struktur seperti JavaScript Object.
- Bedanya, Key ditulis menggunakan tanda kutip dua ("").
- Value bisa menampung nilai primitif atau non-primitif

```
{
  "message": "Berikut daftar kopi yang tersedia",
  "coffees": [
    {
      "id": 1,
      "name": "Kopi Tubruk",
      "price": 12000
    },
    {
      "id": 2,
      "name": "Kopi Tarik",
      "price": 15000
    },
    {
      "id": 3,
      "name": "Kopi Jawa",
      "price": 18000
    }
  ]
}
```

HTTP Verbs/Methods dan HTTP Response Code

- **HTTP Verbs/Methods**

- GET
- POST
- PUT
- DELETE

- **HTTP Response Code**

- 200 (OK)
- 201 (Created)
- 400 (Bad Request)
- 401 (Unauthorized)
- 403 (Forbidden)
- 404 (Not Found)
- 500 (Internal Server Error)

URL

Design

- Gunakan kata benda daripada kata kerja pada endpoint path. (~~/getArticles~~ -> /articles)
- Gunakan kata jamak pada endpoint untuk resource collection. (~~/article~~ -> /articles)
- Gunakan endpoint berantai untuk resource yang memiliki hierarki/relasi. (~~/comments~~ -> /articles/:id/comments)

Dasar-Dasar **Node.js**

- Apa itu Node.js?
- Menjalankan JavaScript Menggunakan Node.js
- Modularization
- Node Package Manager

Apa itu Node.js?

- Diciptakan tahun 2009 oleh Ryan Dahl.
- JavaScript Runtime untuk menjalankan JS di luar lingkungan browser.
- Memungkinkan kita untuk menjadi Full-Stack Developer hanya dengan belajar satu bahasa pemrograman saja. JavaScript!



Menjalankan JavaScript Menggunakan Node.js

- The Node.js REPL (Read-Eval-Print Loop)
- Berkas .js

```
// The Node.js REPL
```

```
C:\dicoding>node  
>console.log('Hello Node.js REPL');  
Hello Node.js REPL
```

```
=====
```

```
// Berkas JavaScript
```

```
const message = (name) => {  
  console.log(`Hello ${name}`);  
}  
message('JavaScript');
```

```
// Terminal
```

```
C:\dicoding>node index.js  
Hello JavaScript
```

Modularization

- Satu berkas .js hanya untuk satu tanggung jawab saja.
- Modularization : Teknik pemisahan kode menjadi modul-modul yang independen.
- Export - Import

```
// coffee.js
```

```
const coffee = {  
  name: 'Tubruk',  
  price: 15000,  
}
```

```
module.exports = coffee;
```

```
=====
```

```
// app.js
```

```
const coffee = require('./coffee');
```

```
console.log(coffee);
```


Node

Package Manager

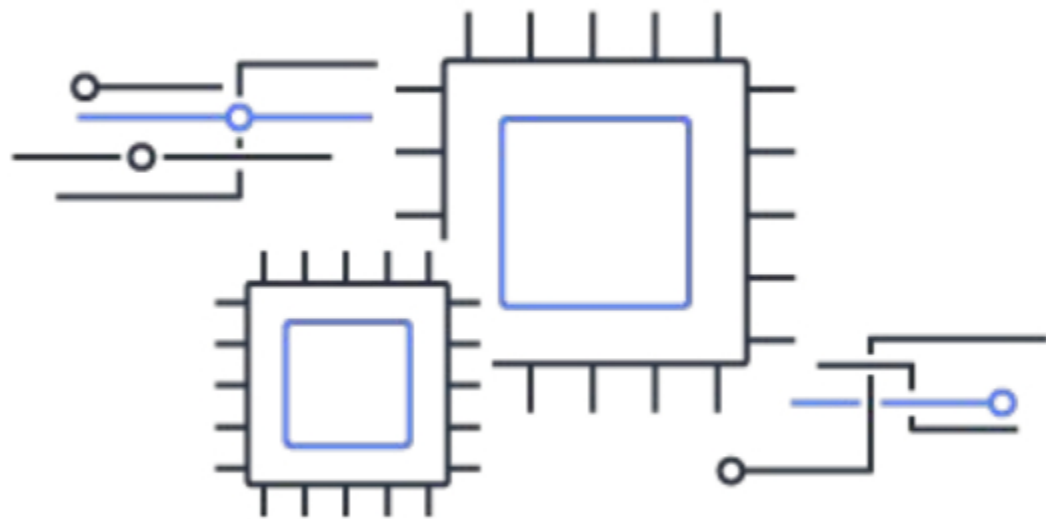
- Pengelola package untuk JavaScript
- Untuk memasang atau menghapus third party module.
- Global dan Local

Deploy **Web Services**

- Amazon EC2
- AWS account
- Deploy ke Amazon EC2
- Process Manager

Amazon EC2

- Server virtual yang berada di infrastruktur AWS
- Lebih hemat ketimbang server tradisional.
- Pay-as-you-go dan Free Tier



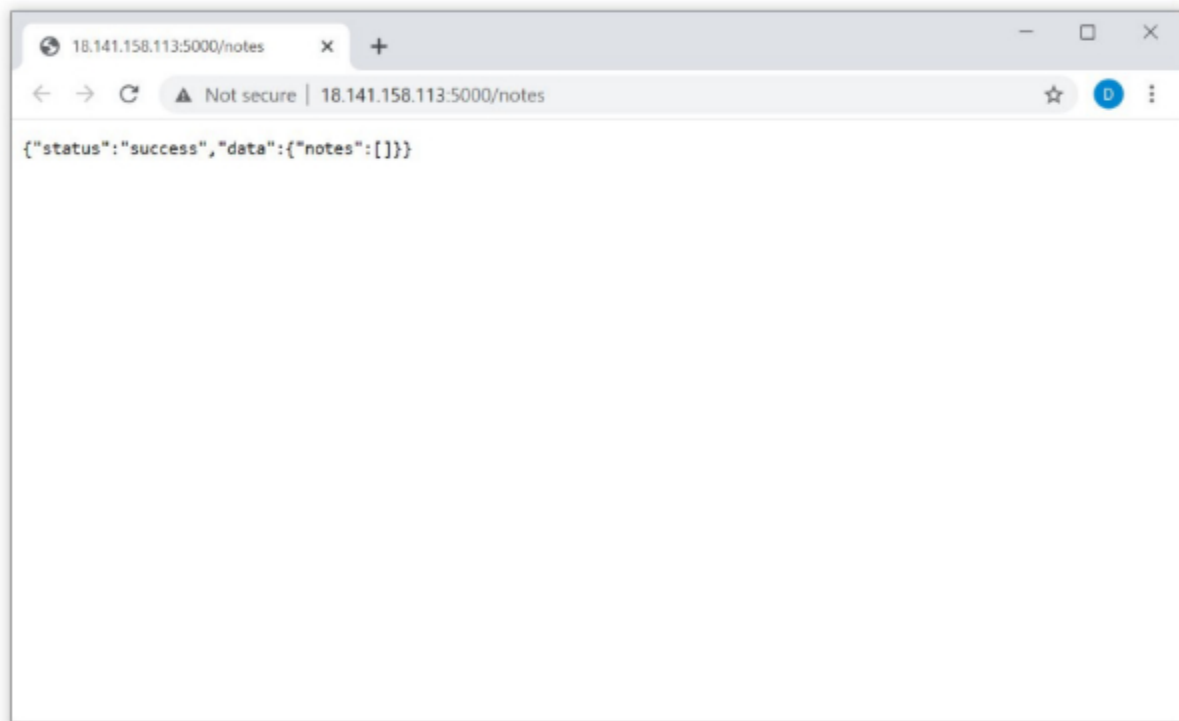
AWS Account

- **Root user** : Akun utama yang punya hak akses tertinggi.
- **IAM user** : Akun yang memiliki hak akses terbatas.



Deploy ke Amazon EC2

- Buat dan jalankan EC2 instance
- Terhubung ke instance melalui SSH
- Unggah proyek web service ke GitHub
- Unduh repository GitHub ke server
- Buka menggunakan IP public dan ubah host menjadi 0.0.0.0



Process Manager

- Web service harus terus beroperasi.
- pm2 : Tools untuk process manager
- Memastikan process akan terus bekerja. Jika process terhenti, pm2 akan otomatis menjalankan ulang process tersebut.

```
ubuntu@ip-172-31-26-251: ~/notes-app-back-end
```

```
[PM2] Spawning PM2 daemon with pm2_home=/home/ubuntu/.pm2
```

```
[PM2] PM2 Successfully daemonized
```

```
[PM2] Starting /home/ubuntu/.nvm/versions/node/v14.15.4/bin/npm in fork_mode (1 instance)
```

```
[PM2] Done.
```

id	name	mode		status	cpu	memory
0	notes-api	fork	0	online	0%	26.6mb

```
ubuntu@ip-172-31-26-251:~/notes-app-back-end$
```

Ujian

Sesi Sharing

Sesi Diskusi

dicoding

Kampus
Merdeka
INDONESIA JAYA