## Nama : Husni Nazar

## Nim : 201111140

## **Setup Environment**

Langkah langkah menyiapkan environment yang akan digunakan untuk simulasi lifecycle API.

## Install Node.JS anda bisa mengintal node js di link ini [Node.js — Run JavaScript Everywhere](https://nodejs.org/en) [(nodejs.org)](https://nodejs.org/en)

1. Setup project

##  Buatlah suatu folder di terminal visual setusio codde lalu tulis **“npm init –y”.**

 Install Express dengan **“npm install express”.**

1. Dan buat lah file di VSC (visual studi code) dengan nama bebas contoh (**index.js) namun dengan format js.**

# CODE SAYA :

# // Import dependencies

# const express = require('express');

# const app = express();

# const port = 3000;

# // Middleware untuk parsing JSON dan logging

# app.use(express.json());

# app.use((req, res, next) => {

# console.log(`${req.method} request for '${req.url}'`);

# next();

# });

# // Data sementara untuk simulasi database

# let resources = [];

# // GET method - Mengambil semua resource

# app.get('/api/resources', (req, res) => {

# res.status(200).json(resources);

# });

# // POST method - Menambahkan resource baru

# app.post('/api/resources', (req, res) => {

# const { name, value } = req.body;

# if (!name || !value) {

# return res.status(400).send('Name and value are required');

# }

# const newResource = { id: resources.length + 1, name, value };

# resources.push(newResource);

# res.status(201).json(newResource);

# });

# // PUT method - Memperbarui resource yang ada

# app.put('/api/resources/:id', (req, res) => {

# const { id } = req.params;

# const { name, value } = req.body;

# const resource = resources.find(r => r.id == id);

# 

# if (!resource) {

# return res.status(404).send('Resource not found');

# }

# if (!name || !value) {

# return res.status(400).send('Name and value are required');

# }

# resource.name = name;

# resource.value = value;

# res.status(200).json(resource);

# });

# // DELETE method - Menghapus resource

# app.delete('/api/resources/:id', (req, res) => {

# const { id } = req.params;

# const resourceIndex = resources.findIndex(r => r.id == id);

# 

# if (resourceIndex === -1) {

# return res.status(404).send('Resource not found');

# }

# resources.splice(resourceIndex, 1);

# res.status(204).send();

# });

# // PATCH method - Memperbarui sebagian dari resource

# app.patch('/api/resources/:id', (req, res) => {

# const { id } = req.params;

# const { name, value } = req.body;

# const resource = resources.find(r => r.id == id);

# 

# if (!resource) {

# return res.status(404).send('Resource not found');

# }

# if (name) resource.name = name;

# if (value) resource.value = value;

# res.status(200).json(resource);

# });

# // HEAD method - Memeriksa keberadaan endpoint

# app.head('/api/resources', (req, res) => {

# res.status(200).end();

# });

# // Menjalankan server

# app.listen(port, () => {

# console.log(`Server berjalan di http://localhost:${port}`);

# });

1. Jalankan server di terminal dengan menulis “**node index.js”**

# Untuk simulasi dengan postman

## Untuk memulai simulasi anda dapat mengunduh postman terlebih dahulu di link ini : [Download Postman | Get Started for Free](https://www.postman.com/downloads/)

1. Buat request baru dengan method yang diinginlkan (GET, POST, PUT, DELETE, PACTCH, HEAD) ethod :

##  Buat request baru dengan method GET.

 Masukan URL yang ada di terminal yang telah di jalanan dengan code “**node index.js**” URL saya ***http://localhost:3000.***

##  Klik send yang ada di kiri dan periksa response.