Tugas Harisenin

Make a post service in Node.js

Oleh: Guido Gusthi Abadi.

Sebelumnya, model dari Post ini sudah disediakan oleh mentor di kodenya. Model ini adalah representasi struktur data untuk postingan, mencakup atribut seperti **user\_id**, **title**, dan **body**. Maka, langkah berikutnya adalah sebagai berikut:

1. Membuat Controller untuk Post

Ini adalah instruksi dari mentor di sesi terakhir Node.js, yaitu buat dahulu controllernya. Sehingga saya melakukannya dengan cara seperti ini:

Saya membuat sebuah **controller** yang akan berfungsi sebagai perantara antara rute dan model. Di dalam **controller** ini, saya akan mengimplementasikan fungsi-fungsi untuk setiap operasi CRUD:

* **createPost**: Ini untuk membuat postingan baru berdasarkan data yang dikirimkan dalam permintaan. Jika berhasil, nanti akan mengirimkan respons dengan status 201 (Created).
* **getAllPosts**: Bertujuan untuk mengambil daftar semua postingan yang ada dalam database.
* **getPostById**: Mengambil postingan berdasarkan ID yang diberikan dalam permintaan.
* **updatePost**: Memperbarui postingan berdasarkan ID dan data yang dikirimkan dalam permintaan.
* **deletePost**: Menghapus postingan berdasarkan ID yang diberikan dalam permintaan.

Setiap fungsi akan mengelola respons yang sesuai, termasuk respons sukses atau ada pesan error jika terjadi masalah.

2. Buat Route untuk Post

Setelah membuat controller, tentu saja hal yang kita lakukan adalah membuat rute atau **routes**. untuk menghubungkan permintaan HTTP dengan fungsi-fungsi yang telah kita definisikan dalam **controller**. Rute ini akan menentukan bagaimana permintaan HTTP akan diproses:

* **POST /api/posts**: Untuk membuat postingan baru.
* **GET /api/posts**: Untuk mengambil daftar semua postingan.
* **GET /api/posts/:postId**: Untuk mengambil postingan berdasarkan ID.
* **PUT /api/posts/:postId**: Untuk memperbarui postingan berdasarkan ID.
* **DELETE /api/posts/:postId**: Untuk menghapus postingan berdasarkan ID.

Berikut adalah implementasinya dalam VS Code :

const express = require("express");

const router = express.Router();

const postController = require("../controllers/post.controller");

// Create a new post

router.post("/posts", postController.createPost);

// Read all posts

router.get("/posts", postController.getAllPosts);

// Read a specific post

router.get("/posts/:postId", postController.getPostById);

// Update a post

router.put("/posts/:postId", postController.updatePost);

// Delete a post

router.delete("/posts/:postId", postController.deletePost);

module.exports = router;

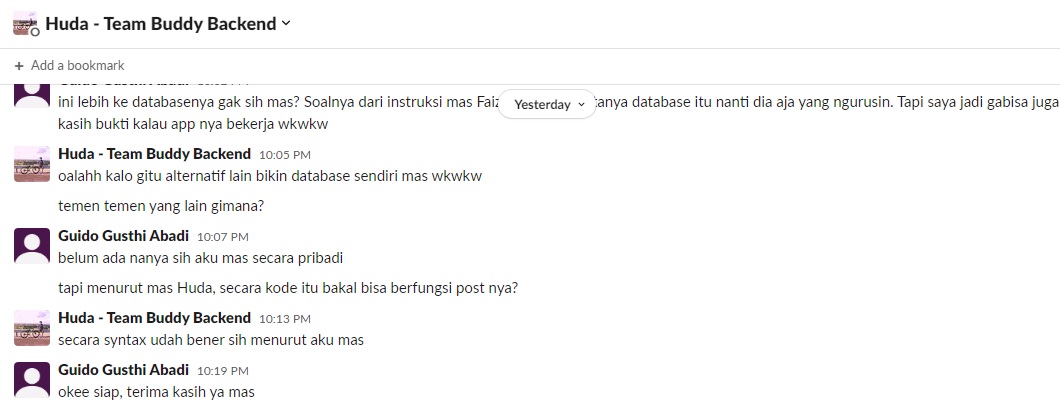
Bisa dilihat, disini kita memanggil **controller** yang sebelumnya sudah kita buat untuk bisa membuat **routes** ini.

3. Hubungkan Route ke Aplikasi Utama kita

Akhirnya, saya akan menghubungkan **route** post ke aplikasi utama menggunakan Express. Dalam kode utama (**index.js**), kita akan menggunakan rute yang telah kita buat untuk menangani permintaan terkait postingan. Hal ini akan memastikan bahwa setiap permintaan ke rute yang sesuai akan diteruskan ke fungsi yang tepat dalam **controller**.

Catatan kecil :

Saya sudah mencoba untuk tes aplikasi ini, tapi selalu ada error. Dan akhirnya, saya menghubungi team buddy saya kak Huda untuk minta bantuan. Kak Huda menjelaskan, bahwa dari segi kode tidak ada error dan mengiyakan kalau seharusnya kode ini akan berjalan lancar nantinya. Beriktu adalah buktinya :



Saya sudah sekitar 2 harian mencari solusinya, baik dari youtube ataupun searching di google. Tapi tetap saja selalu ada error. Semoga nanti ada solusi dan nilai yang terbaik ☺