**Name : Muhamad Fahraz Firdaus**

**NIM : 20220040131**

**Class : TI22 I**

**Course : platform-Based Programming**

**Assignment Session 10**

Here's a breakdown of the studentController.js file:

Purpose:

* It's a controller in a Node.js application using Express.js.
* It handles API requests related to student data (mahasiswa).
* It interacts with a database to retrieve, create, update, and delete student records.

Structure:

1. Imports:
   * express: Core library for creating web servers and handling requests.
   * db: A database module (not shown, but likely handles database connections and queries).
2. Router:
   * Creates an Express.js router to define specific routes and their handlers.
3. Routes and Handlers:
   * GET /mahasiswa:
     + Fetches all student records from the database.
     + Returns a JSON response containing the fetched students.
   * GET /mahasiswa/:nim:
     + Fetches a single student record based on the provided NIM (student ID).
     + Returns a JSON response containing the student data if found, otherwise a 404 Not Found error.
   * PUT /mahasiswa/:nim:
     + Updates an existing student record with provided data in the request body.
     + Returns a success message if the update is successful.
   * POST /mahasiswa:
     + Creates a new student record with data from the request body.
     + Returns a success message if the creation is successful.
   * DELETE /mahasiswa/:nim:
     + Deletes a student record with the specified NIM.
     + Returns a success message if the deletion is successful.
4. Error Handling:
   * Each route handler includes error handling to catch potential database errors or other issues.
   * If errors occur, appropriate status codes (500 Internal Server Error or 404 Not Found) and JSON error messages are returned.

Key Points:

* It follows a common pattern for RESTful API controllers using Express.js.
* It makes direct SQL queries using the db module, which might be less secure and flexible than using an Object-Relational Mapper (ORM).
* It doesn't implement input validation or authentication, which are important for security in production environments.