# MODUL PERKULIAHAN

**Mata Kuliah** : Pengembangan Aplikasi Web  
**Materi** : Melengkapi RESTful API (Update, Delete, Search) dengan Sequelize  
**Pertemuan** : 05  
**Fakultas** : Teknik  
**Program Studi** : Teknologi Informasi  
**Kode MK** : AntaraTI501P  
**Disusun Oleh** : Ir.Asroni, S.T., M.Eng

## DAFTAR ISI

1. [PEMBAHASAN](#pembahasan)
2. [PRAKTIKUM](#praktikum)
   * [A. Tujuan Praktikum](#a-tujuan-praktikum)
   * [B. Materi](#b-materi)
   * [C. Metode Praktikum](#c-metode-praktikum)
   * [D. Langkah-langkah Praktikum](#d-langkah-langkah-praktikum)
3. [TUGAS](#tugas)

## PEMBAHASAN

1. Pengenalan RESTful API:
   * Konsep dan prinsip REST
   * HTTP methods (GET, POST, PUT, DELETE, etc.)
   * Status codes dan best practices
2. Desain RESTful API:
   * Resource naming conventions
   * URL structure
   * Query parameters dan filtering
3. Implementasi RESTful API dengan Express.js:
   * Routing untuk berbagai HTTP methods
   * Request handling dan response formatting
4. Middleware dalam RESTful API:
   * Body parsing
   * CORS handling
   * Authentication middleware

## PRAKTIKUM

### A. Tujuan Praktikum

Mahasiswa akan melengkapi API Presensi dengan mengimplementasikan fungsionalitas **UPDATE**, **DELETE**, dan **SEARCH** menggunakan Sequelize, parameter rute, dan parameter kueri.

### B. Materi

1. Pembuatan RESTful API endpoints
2. Implementasi CRUD operations melalui API
3. Penggunaan middleware untuk validasi dan autentikasi
4. Penanganan error dan response formatting

### C. Metode Praktikum

1. Penjelasan materi oleh instruktur
2. Demonstrasi langsung pembuatan RESTful API
3. Praktik mandiri oleh mahasiswa
4. Diskusi dan tanya jawab

### D. Langkah-langkah Praktikum

1. **Implementasi DELETE (Menghapus Data Presensi)** Kita akan membuat *endpoint* baru untuk menghapus satu catatan presensi berdasarkan ID-nya.
   * Buat Controller **deletePresensi** di **presensiController.js:**

exports.deletePresensi = *async* (*req*, *res*) => {

try {

*const* { id: userId } = req.user;

*const* presensiId = req.params.id;

*const* recordToDelete = await Presensi.findByPk(presensiId);

if (!recordToDelete) {

return res

.status(404)

.json({ *message:* "Catatan presensi tidak ditemukan." });

}

if (recordToDelete.userId !== userId) {

return res

.status(403)

.json({ *message:* "Akses ditolak: Anda bukan pemilik catatan ini." });

}

await recordToDelete.destroy();

res.status(204).send();

} catch (error) {

res

.status(500)

.json({ *message:* "Terjadi kesalahan pada server", *error:* error.message });

}

};

* + Tambahkan Route delete di **routes/presensi.js:**

A screen shot of a computer program

AI-generated content may be incorrect.

* + **Uji Coba Postman:** Lakukan *request* DELETE ke http://localhost:3001/api/presensi/1 (ganti 1 dengan ID presensi yang ada di database Anda).

1. **Implementasi UPDATE (Mengubah Data Presensi)** Kita akan membuat *endpoint* untuk mengoreksi/mengubah catatan presensi.
   * Buat Controller **updatePresensi** di **presensiController.js:**

exports.updatePresensi = *async* (*req*, *res*) => {

try {

*const* presensiId = req.params.id;

*const* { checkIn, checkOut, nama } = req.body;

if (checkIn === undefined && checkOut === undefined && nama === undefined) {

return res.status(400).json({

*message:*

"Request body tidak berisi data yang valid untuk diupdate (checkIn, checkOut, atau nama).",

});

}

*const* recordToUpdate = await Presensi.findByPk(presensiId);

if (!recordToUpdate) {

return res

.status(404)

.json({ *message:* "Catatan presensi tidak ditemukan." });

}

recordToUpdate.checkIn = checkIn || recordToUpdate.checkIn;

recordToUpdate.checkOut = checkOut || recordToUpdate.checkOut;

recordToUpdate.nama = nama || recordToUpdate.nama;

await recordToUpdate.save();

res.json({

*message:* "Data presensi berhasil diperbarui.",

*data:* recordToUpdate,

});

} catch (error) {

res

.status(500)

.json({ *message:* "Terjadi kesalahan pada server", *error:* error.message });

}

};

* + TambahkanRoute di **routes/presensi.js:**

A screen shot of a computer code

AI-generated content may be incorrect.

* + **Uji Coba Postman:** Lakukan *request* PUT ke *http://localhost:3001/api/presensi/1*. Jangan lupa kirim *Body* JSON berisi data yang ingin diubah, contoh:

A screenshot of a computer

AI-generated content may be incorrect.

1. **Implementasi SEARCH (di reportController.js)** Kita akan memodifikasi getDailyReport agar bisa memfilter berdasarkan nama karyawan.
   * Impor Operator Sequelize:

*const* { Presensi } = require("../models");

*const* { Op } = require("sequelize");

* + Modifikasi **getDailyReport**:

exports.getDailyReport = *async* (*req*, *res*) => {

try {

*const* { nama } = req.query;

*let* options = { *where:* {} };

if (nama) {

options.where.nama = {

*[Op.like]:* `%${nama}%`,

};

}

*const* records = await Presensi.findAll(options);

res.json({

*reportDate:* new Date().toLocaleDateString(),

*data:* records,

});

} catch (error) {

res

.status(500)

.json({ *message:* "Gagal mengambil laporan", *error:* error.message });

}

};

* + - * + **Uji Coba Postman:** Lakukan *request* GET ke *http://localhost:3001/api/reports/daily?nama=User*. Ini akan mencari semua presensi yang namanya mengandung kata "User".

## TUGAS

1. **Lengkapi Fitur reportController**: Modifikasi **getDailyReport** agar bisa memfilter berdasarkan **rentang tanggal** (tanggalMulai dan tanggalSelesai) menggunakan *query parameters* dan Sequelize [Op.between].
2. **Validasi Input**: Terapkan express-validator pada *endpoint* PUT /api/presensi/:id untuk memastikan data waktuCheckIn dan waktuCheckOut yang dikirim adalah format tanggal yang valid.
3. Buat file Tugas5.md di dalam folder TUGAS dengan menyertakan gambar screenshot:
   * Endpoint update data presensi
   * Endpoint update jika format tanggal yang diisi tidak valid
   * Endpoint delete data
   * Enpoint search berdasarkan nama
   * Endpoint search berdasarkan tanggal
4. Kirim link file **Tugas5.md** dari repository GitHub ke dalam MyKlass praktikum PAW.