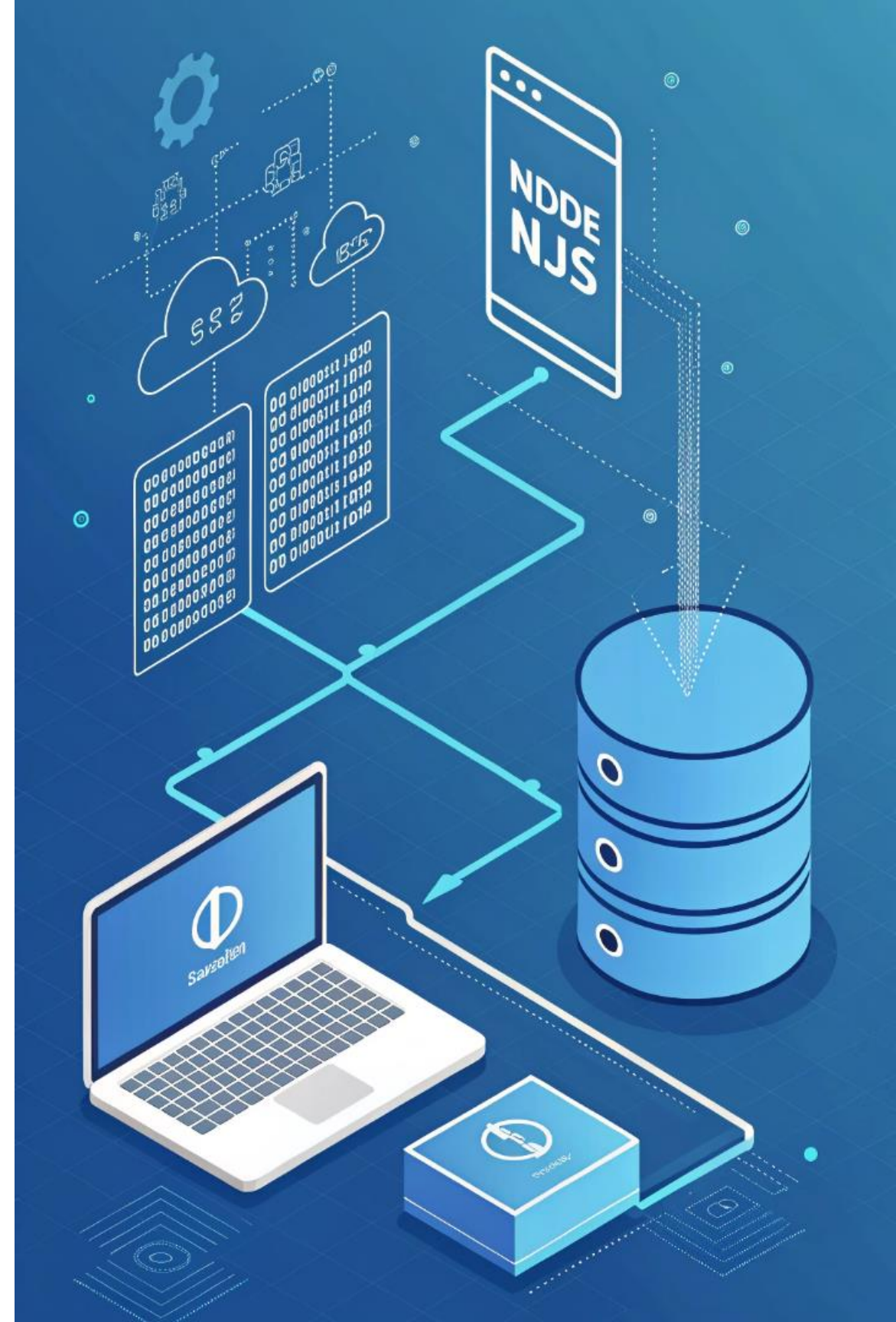


Integrasi Node.js dengan Basis Data Menggunakan Sequelize ORM

- Mahasiswa mampu menghubungkan aplikasi Node.js dengan basis data menggunakan ORM Sequelize (untuk SQL) atau Mongoose (untuk MongoDB).
- Mahasiswa mampu membuat model data dan tabel basis data menggunakan ORM.
- Mahasiswa mampu melakukan operasi CRUD (Create, Read, Update, Delete) pada basis data melalui aplikasi web menggunakan ORM.

Jeffry, S.kom.,M.T





Teori Singkat ORM dan Sequelize

ORM (Object Relational Mapping) adalah teknik untuk mengakses dan memanipulasi data dalam database relasional menggunakan objek dalam bahasa pemrograman. Teknik ini menjembatani perbedaan antara paradigma pemrograman berorientasi objek dengan model data relasional.

Sequelize adalah ORM berbasis JavaScript untuk Node.js yang mendukung berbagai jenis database seperti MySQL, PostgreSQL, SQLite, dan MSSQL. Keunggulan Sequelize meliputi dukungan untuk relasi antar tabel, operasi CRUD, otomatisasi pembentukan query SQL, dan kemampuan migrasi skema database.

Mendukung Relasi

Memudahkan pembuatan dan pengelolaan hubungan antar tabel seperti one-to-many, many-to-many, dan one-to-one.

Operasi CRUD

Menyediakan metode sederhana untuk operasi Create, Read, Update, dan Delete tanpa menulis query SQL kompleks.

Otomatisasi Query

Menghasilkan query SQL yang optimal secara otomatis berdasarkan definisi model dan relasi.

Migrasi Skema

Memungkinkan perubahan struktur database secara terprogram dan terkontrol.

Instalasi dan Setup Awal

Untuk memulai proyek integrasi Node.js dengan basis data menggunakan Sequelize, kita perlu melakukan inialisasi proyek dan menginstal paket-paket yang diperlukan. Langkah pertama adalah menjalankan perintah **npm init -y** untuk membuat file `package.json`.

Selanjutnya, kita perlu menginstal paket-paket pendukung dengan perintah **npm install express sequelize mysql2 body-parser cors**. Paket-paket ini akan membantu kita dalam membangun aplikasi web dan mengintegrasikannya dengan basis data.

Node.JS

◇ Config

◇ Models

◇

◇ Routes



Inisialisasi Proyek

Jalankan perintah `npm init -y` untuk membuat file `package.json`



Instalasi Paket

Instal `express`, `sequelize`, `mysql2`, `body-parser`, dan `cors`



Struktur Direktori

Buat struktur direktori yang terdiri dari `config`, `models`, `routes`, dan file `index.js`



Praktikum Langkah per Langkah

Dalam praktikum ini, kita akan mempelajari cara menghubungkan Node.js dengan basis data, membuat model, dan melakukan operasi CRUD menggunakan Sequelize ORM. Langkah pertama adalah membuat file konfigurasi database di **config/db.js** yang berisi informasi koneksi ke database.

Selanjutnya, kita akan membuat model Mahasiswa di **models/mahasiswa.js** dan mengimplementasikan operasi CRUD di **routes/mahasiswa.js**. Kita juga akan mempelajari cara menyusun relasi antar tabel, seperti hubungan antara Mahasiswa dan MataKuliah.

Menghubungkan dengan Basis Data

Buat file **config/db.js** untuk mengonfigurasi koneksi ke database MySQL menggunakan Sequelize

Membuat Model dan Tabel

Definisikan model Mahasiswa dengan atribut nama, nim, dan jurusan di **models/mahasiswa.js**

Implementasi Operasi CRUD

Buat **routes/mahasiswa.js** dengan endpoint untuk operasi GET, POST, PUT, dan DELETE

Menyusun Relasi Antar Tabel

Definisikan model MataKuliah dan atur relasi one-to-many dengan model Mahasiswa

Menjalankan Aplikasi



Setelah mengimplementasikan semua komponen yang diperlukan, langkah selanjutnya adalah menjalankan aplikasi. File utama **index.js** berperan sebagai entry point aplikasi yang mengintegrasikan semua komponen yang telah kita buat sebelumnya.

Dalam file ini, kita mengimpor modul-modul yang diperlukan seperti express, cors, body-parser, dan sequelize. Kita juga mengonfigurasi middleware dan menentukan rute API untuk endpoint mahasiswa. Terakhir, kita menjalankan sinkronisasi database dan memulai server pada port 5000.



Impor Modul

Mengimpor express, cors, body-parser, sequelize, dan rute mahasiswa



Konfigurasi Middleware

Mengatur cors dan body-parser untuk menangani permintaan JSON



Definisi Rute

Menentukan '/api/mahasiswa' sebagai endpoint untuk operasi CRUD mahasiswa



Sinkronisasi Database

Menjalankan sequelize.sync() untuk menyinkronkan model dengan database



Menjalankan Server

Memulai server pada port 5000 dan menampilkan pesan konfirmasi

Penutup dan Rangkuman

Modul praktikum ini telah membahas cara mengintegrasikan Node.js dengan basis data menggunakan Sequelize ORM. Mahasiswa telah mempelajari konsep ORM dan keunggulan Sequelize, serta mengimplementasikan aplikasi web yang terintegrasi dengan basis data relasional.

Melalui praktikum ini, mahasiswa diharapkan mampu menghubungkan Node.js dengan basis data, membuat model dan tabel, melakukan operasi CRUD, mengonfigurasi koneksi basis data, dan menyusun relasi antar tabel menggunakan ORM.

Koneksi Database

Menghubungkan Node.js dengan basis data MySQL menggunakan Sequelize

1



Pemodelan Data

Membuat model dan tabel basis data dengan pendekatan berorientasi objek

Relasi Antar Tabel

Menyusun hubungan one-to-many dan relasi lainnya antar model



Operasi CRUD

Mengimplementasikan Create, Read, Update, dan Delete menggunakan ORM