

# BAB I PENDAHULUAN

## I.1 Latar Belakang

Perkembangan teknologi di era sekarang mempengaruhi berbagai bidang salah satunya adalah pada bidang pendidikan, yang mulai beradaptasi dengan aplikasi-aplikasi berbasis teknologi. Perkembangan teknologi merupakan salah satu proses yang penting di era sekarang dan dikenal sebagai “*knowledge society*” hal ini meliputi pengenalan *Information and Communication Technologies* (ICT) dalam berbagai konteks masyarakat seperti di dalam pendidikan (Daniel dkk., 2021). Perkembangan teknologi juga terjadi saat pandemi, banyak institusi pendidikan yang mulai menggunakan aplikasi-aplikasi berbasis teknologi. Covid-19 memberi dampak pada sekolah, perguruan tinggi, dan universitas di seluruh dunia untuk menunda kelas tatap muka dan menerapkan transformasi digital disemua sistem pendidikan (Banerjee dkk., 2021).

Universitas Telkom menjadi salah satu yang menggunakan kesempatan itu untuk melakukan perkembangan, khususnya pada Fakultas Rekayasa Industri yang memiliki beberapa aplikasi berbasis digital. Satu dari beberapa aplikasi yang digunakan untuk membantu produktivitas Fakultas Rekayasa Industri yaitu aplikasi SOFI yang merupakan aplikasi manajemen sidang TA (Tugas Akhir). Aplikasi tersebut digunakan untuk registrasi sidang, penjadwalan sidang, penilaian sidang, revisi sidang, dan laporan sidang.

Aplikasi SOFI saat ini menggunakan arsitektur monolitik. Arsitektur monolitik merupakan pendekatan tradisional yang dimana sebuah aplikasi dibangun dengan satu kode dasar yang mencakup beberapa layanan (Blinowski dkk., 2022). Pada aplikasi SOFI yang menggunakan arsitektur monolitik memiliki kekurangan seiring dengan pertumbuhan sebuah aplikasi dan kompleksitas pada aplikasi tersebut. Arsitektur monolitik memiliki beberapa kekurangan yaitu aplikasi sulit dipahami atau dimodifikasi yang menghambat pengembangan, perubahan kecil memerlukan pembuatan ulang seluruh aplikasi, dan skalabilitas hanya dapat dilakukan secara horizontal (Ponce et al., n.d. 2019 ).

Dalam beberapa kasus, migrasi ke arsitektur *microservice* merupakan pilihan yang tepat untuk menyelesaikan masalah yang ada dan dapat meningkatkan

kemampuan dalam pemeliharaan sebuah sistem (Ponce et al., n.d. 2019). *Microservice* merupakan modul-modul yang berdiri sendiri dan memiliki batasan fungsi bisnis terdefinisi dengan baik (Trichur Ramachandran, n.d.2021). Melakukan migrasi dari monolitik ke *microservice* membuat aplikasi lebih fleksibel, modular, mudah untuk berkembang dan memiliki skalabilitas yang baik. Arsitektur *microservice* memungkinkan layanan-layanan untuk diimplementasikan secara *independent* (Blinowski dkk., 2022).

Di sisi *frontend*, aplikasi SOFI juga masih menggunakan *Multi-Page Application*. *Multi Page Application* (MPA) merupakan aplikasi web terdiri dari banyak halaman yang berbeda dan halaman web sepenuhnya dimuat ulang dari server setiap kali pengguna berinteraksi dengan aplikasi dan memuat konten baru. Meskipun MPA memiliki keunggulan dalam navigasi dan struktur yang lebih tradisional, ada beberapa kelemahan yang dapat mengurangi pengalaman pengguna. Salah satu kelemahan yaitu mengalami error saat memuat data pengguna. Ketika aplikasi perlu memproses data dalam jumlah besar, ada kemungkinan besar terjadi eror. Hal ini dapat mengganggu fungsionalitas aplikasi dan mengurangi pengalaman pengguna. Jeng et al., (2021) menyatakan bahwa MPA dapat mengalami respon yang lemah dan keterlambatan karena arsitektur tradisional yang merugikan dalam pengumpulan data baru setiap kali dibutuhkan. Hal ini dapat menjadi masalah, terutama jika aplikasi perlu mengelola data pengguna dalam jumlah besar secara berkala. Selain itu, adanya potensi hilangnya data yang telah diinput oleh pengguna sebelum error terjadi. Pemuatan ulang yang berulang juga dapat mengakibatkan kehilangan data, jika pengguna secara tidak sengaja meninggalkan halaman atau mengalami kesalahan selama proses pemuatan ulang, informasi yang sudah di isi mungkin tidak akan tersimpan (Ghasemzadeh dkk., 2023). Ini menyebabkan pengguna harus menginput ulang dari awal, yang sangat menghambat dan menyulitkan pengguna. Dalam pengguna aplikasi SOFI dengan rentang pengguna yang dari mahasiswa dan dosen junior hingga dosen senior, terdapat peluang besar untuk meningkatkan *intuitive* aplikasi dan meningkatkan kepuasan pengguna. Selain itu, memperbaiki *error handling* dan meningkatkan *response time* dapat sangat membantu dalam menyelesaikan

masalah yang muncul, sehingga menghasilkan pengalaman pengguna yang lebih responsif dan efisien.

Untuk menghadapi masalah tersebut, aplikasi SOFI dapat beralih ke *Single Page Application*. *Single Page Application* (SPA) merupakan aplikasi web yang terdiri dari komponen-komponen individu yang dapat di perbarui secara dinamis tanpa perlu memuat ulang seluruh halaman setiap permintaan dari pengguna. Kornienko et al., (2021) menambahkan jika SPA merupakan kombinasi solusi, membagi aplikasi web menjadi dua (*frontend* dan *backend*) dan mengatur interaksi di antara keduanya. Dengan menggunakan SPA, data yang diinput oleh pengguna tidak akan hilang sebelum eror terjadi. Hal ini dapat meningkatkan *intuitive* pengguna terhadap aplikasi agar lebih mudah digunakan, serta menyelesaikan permasalahan interaksi aplikasi dengan pengguna. Selain itu, saat mengalami error ketika memuat data pengguna karena terlalu banyak, SPA dapat mengurangi waktu respon. Dengan SPA memuat semua sumber daya sekaligus dan memperbarui konten secara dinamis, SPA memberikan pengalaman yang lebih mulus dan interaktif bagi pengguna (Ghasemzadeh dkk., 2023). Selain itu, SPA dapat meningkatkan responsivitas aplikasi. Masalah yang dialami pada aplikasi SOFI dapat diatasi dengan menggunakan SPA sebagai solusi untuk mengatasi permasalahan yang ada.

Dari penjelasan di atas, penelitian ini melakukan migrasi aplikasi SOFI dari MPA ke SPA dengan metode *iterative incremental*. Metode ini merupakan gabungan antara metode *iterative* dan metode *incremental*. Keduanya berfokus pada pengembangan bertahap dan berulang untuk meningkatkan kualitas dan menyesuaikan sistem sesuai dengan kebutuhan pengguna. Metode *iterative* memungkinkan revisi dan penyempurnaan berulang berdasarkan umpan balik dan pembelajaran dari iterasi dengan strategi penjadwalan pengerjaan ulang dimana waktu disisihkan untuk merevisi dan memperbaiki bagian sistem (Okesola et al., 2020). Metode ini mengembangkan aplikasi secara berulang dengan mengevaluasi dan melakukan perbaikan yang diperlukan pada setiap tahap, mengurangi risiko dan meningkatkan kualitas aplikasi.

## **I.2 Perumusan Masalah**

Berdasarkan latar belakang, rumusan masalah yang mendasari penelitian ini adalah:

- a. Bagaimana cara mengatasi error pada aplikasi SOFI saat memuat data pengguna karena terlalu banyak data yang harus diproses, menyebabkan kinerja aplikasi menurun dan menimbulkan tantangan dalam memastikan aplikasi tetap responsif dan efisien?
- b. Bagaimana mengatasi masalah error handling pada aplikasi SOFI, seperti saat pengguna ingin menginput penjadwalan, inputan tersebut bisa hilang sebelum error terjadi, semua data hilang menimbulkan tantangan signifikan dalam memberikan pengalaman pengguna yang memuaskan?

## **I.3 Tujuan Penelitian**

Berdasarkan rumusan masalah sebelumnya, Penelitian ini bertujuan untuk:

- a. Meningkatkan performa aplikasi SOFI saat memuat data pengguna dengan mentransisi ke SPA, untuk mengurangi waktu respon. Hal ini membuat aplikasi tetap responsif dan efisien.
- b. Mengembangkan aplikasi SOFI dengan menggunakan SPA untuk memastikan bahwa data yang di input masih ada dan tidak hilang sebelum error terjadi. Dengan SPA, ada peluang untuk meningkatkan intuitivitas aplikasi serta memberikan pengalaman pengguna yang lebih baik.

## **I.4 Batasan Penelitian**

Penelitian ini memiliki beberapa batasan, diantaranya sebagai berikut:

- a. Penelitian tidak mengubah tampilan atau menambah fitur pada aplikasi SOFI.
- b. Menggunakan React Js dalam mengembangkan aplikasi SOFI.
- c. Penelitian ini berfokus pada modul penjadwalan.
- d. Melakukan pengujian menggunakan UAT dengan metode black box testing.

## **I.5 Manfaat Penelitian**

Manfaat penelitian ini:

1. Bagi Fakultas Rekayasa Industri, penelitian ini bermanfaat dalam meningkatkan produktifitas Fakultas Rekayasa Industri yang dimana

memberikan pengalaman pengguna yang lebih baik lebih responsif dan efisien.

2. Bagi peneliti, penelitian ini bermanfaat menambah wawasan dan pengetahuan baru tentang penelitian ini dan tentang migrasi dari MPA menjadi SPA untuk masa yang akan datang.