BAB I PENDAHULUAN

I.1 Latar Belakang

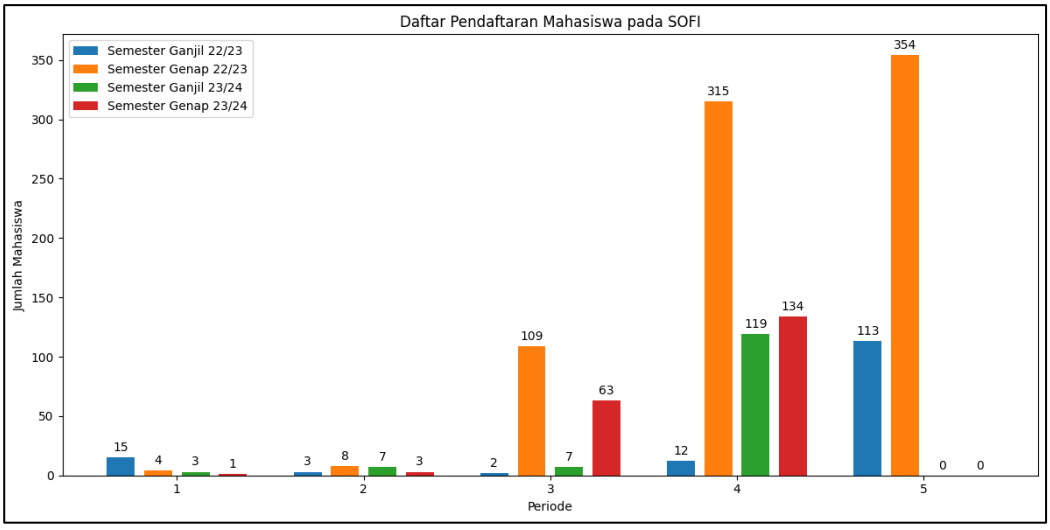
Perkembangan transformasi digital membawa banyak perubahan pada industri saat ini, bukan hanya industri bisnis saja yang mengalami transformasi digital yang sangat pesat, namun pendidikan juga banyak yang beralih pada transformasi digital atau bisa disebut pendidikan 4.0. Pendidikan 4.0 sendiri memiliki tujuan untuk memberikan siswa kemampuan kognitif, interpersonal, sosial, teknis agar mereka siap menghadapi tuntutan revolusi industri 4.0 dan tantangan global (Katyeudo & de Souza, 2022). Di era digital saat ini, pendidikan harus diperbarui dengan materi dan metode inovatif, serta meningkatkan aksesibilitas sumber daya pendidikan tanpa terkendala waktu dan tempat. Transformasi digital dalam pendidikan adalah bagian penting dari perubahan sosial yang terjadi saat ini (Morze & Strutynska, 2023). Hal ini juga dipicu karena pandemi covid pada tahun 2020 lalu, yang memaksa transformasi digital harus berkembang pesat dikarenakan pembelajaran jarak jauh yang sudah menjadi standar umum, serta sumber belajar yang tersedia dalam format digital (Kim HYe-Jin, 2021).

Universitas Telkom merupakan salah satu perguruan tinggi swasta yang terkena dampak dari transformasi digital tersebut. Salah satu dampak dari perubahan ini adalah munculnya sebuah aplikasi inovatif yang dikenal dengan nama "SOFI". SOFI (Sidang *Online* Fakultas Rekayasa Industri) adalah perangkat lunak berbasis web yang mendukung kegiatan akademik Fakultas Rekayasa Industri (FRI) di Universitas Telkom. Aplikasi SOFI digunakan oleh mahasiswa, dosen, dan pihak terkait di lingkungan universitas untuk mengelola proses sidang Tugas Akhir (TA). Fitur-fitur dalam SOFI mencakup pendaftaran, penjadwalan, pelaksanaan sidang, dan proses revisi TA. Pengembangan aplikasi SOFI dilakukan dengan menggunakan arsitektur monolitik.

Implementasi arsitektur monolitik ini memberikan beberapa keunggulan terutama ketika aplikasi mempunyai *scope* yang kecil dan hanya memiliki beberapa fungsi saja. Aplikasi tersebut memiliki kelebihannya tersendiri, seperti kemudahan dalam pengembangan dan pengujian (De Lauretis, 2019). Meskipun memiliki beberapa keunggulan, arsitektur monolitik juga memiliki kelemahan, seperti

ketidakmampuan layanan beroperasi secara independen. Hal ini dapat menyebabkan seluruh sistem terhenti jika satu layanan mengalami masalah (Trichur Ramachandran dkk., 2021). Kekurangan lainnya adalah terdapat tingginya duplikasi kode program dalam arsitektur monolitik (Miika, 2018). Selain itu seiring dengan meningkatnya kompleksitas aplikasi, modifikasi fitur menjadi sangat sulit, yang pada akhirnya memperlambat pengembangan aplikasi dan susahnya melakukan penskalaan aplikasi yang sudah kompleks di arsitektur monolitik, yang dimana mengakibatkan penambahan seluruh sumber daya yang ada serta meningkatnya kompleksitas yang lebih tinggi (De Lauretis, 2019). Pengimplementasian aplikasi monolitik sendiri sesuai ketika, aplikasi yang dikembangkan bersifat ringan dan sederhana. Namun, arsitektur ini kurang sesuai ketika aplikasi yang dikembangkan sangat kompleks dan memerlukan pengembangan yang terus-menerus (Makris dkk., 2022).

Aplikasi sidang fakultas SOFI mengalami masalah skalabilitas yang sangat terbatas. Hal ini terbukti dalam sesi wawancara dengan pengembang aplikasi SOFI, yang dimana pengembang kesulitan dalam melakukan skalabilitas aplikasi seperti horizontal *scaling* pada layanan tertentu yang memiliki *load* pengguna yang sangat tinggi dan tidak menentu.



Gambar I.1 Daftar Pendaftaran Mahasiswa pada SOFI

Terlebih lagi, berdasarkan data dari Layanan Administrasi Akademik Fakultas Rekayasa Industri (LAA FRI) memperlihatkan fluktuasi jumlah pendaftaran mahasiswa pada SOFI yang tidak menentu pada berbagai periode dan semester

yang digambarkan pada Gambar I.1, dengan puncaknya mencapai 315 pendaftaran pada periode ke-4 dan 354 pendaftaran pada periode ke-5 di Semester Genap 22/23. Keadaan ini menunjukkan bahwa layanan tertentu mengalami beban yang sangat tinggi secara tiba tiba, sehingga menyulitkan pengembang untuk meningkatkan layanan. Hal ini dikarenakan pada arsitektur monolitik, skalabilitas aplikasi hanya dapat dilakukan dengan menambah spesifikasi server yang sedang berjalan, ini berarti skalabilitas dilakukan pada satu kesatuan aplikasi yang ada (Al-Debagy & Martinek, 2018).

Oleh karena itu, solusi yang diharapkan untuk mengatasi permasalahan ini adalah dilakukannya migrasi aplikasi *backend* dari arsitektur monolitik ke arsitektur *microservice*. Dengan dilakukannya migrasi ke arsitektur *microservice* tentunya menghasilkan aplikasi server yang memiliki tingkat skalabilitas yang tinggi (Munawar & Hodijah, 2018). Tentunya langkah migrasi ini diharapkan dapat memberikan manfaat lainnya yang signifikan, seperti meningkatnya ketersediaan (Salii dkk., 2023), peningkatan performa aplikasi (Prasandy dkk., 2020), serta mengurangi beban biaya dan pemeliharaan yang mudah (Kuryazov dkk., 2020).

Meskipun begitu, proses migrasi aplikasi yang memiliki arsitektur monolitik diubah menjadi *microservice* merupakan proses yang kompleks, karena setiap aplikasi monolitik memiliki keunikan tersendiri dan dapat menghadirkan berbagai rintangan selama proses migrasi arsitektur (Trichur Ramachandran dkk., 2021). Oleh karena itu, dalam proses pengembangan, penulis memilih pendekatan *Iterative Incremental Development* karena pendekatan ini cocok dengan kondisi tim, ukuran proyek, dan memungkinkan perbaikan dilakukan selama tahap pengembangan aplikasi, hal ini guna memastikan kesesuaian pengembangan. Selain itu juga penggunaan prinsip atau pola *Domain Driven Design* (DDD) pada analisis sistem juga diadopsi. Pembuatan *microservice* tanpa menerapkan prinsip atau pola DDD dapat menimbulkan kesulitan dalam menentukan pembagian sistem dan definisi seberapa kecil seharusnya layanan tersebut. (Rahmatulloh dkk., 2021).

I.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dipaparkan, maka rumusan masalah yang ditemukan pada penelitian ini adalah sebagai berikut:

a. Bagaimana menghadapi tantangan dalam mengidentifikasi pembagian seberapa kecil modul pendaftaran dan penjadawalan yang akan dilakukan migrasi ke arsitektur *microservice* aplikasi SOFI yang dapat mempengaruhi keutuhan dan konsistensi domain bisnis?
b. Bagaimana pendekatan efektif yang dapat digunakan untuk menghadapi kompleksitas dalam pengembangan *backend* pada modul pendaftaran dan penjadwalan yang dipecah menjadi aplikasi *microservice* sehingga memastikan kesesuaian hasil pengembangan?

I.3 Tujuan Penelitian

Berdasarkan permasalahan yang terjadi maka tujuan yang ingin dicapai pada penelitian ini antara lain:

a. Mengimplementasikan *Domain Driven Design* dalam tahap perancangan sistem sehingga dapat mengetahui seberapa kecil modul pendaftaran dan penjadwalan yang sudah ada, hal ini dapat menjaga keutuhan serta konsistensi domain bisnis.

b. Mengimplementasikan metode *Iterative Incremental Development* dalam proses pengembangan *backend* pada modul pendaftaran dan penjadwalan yang dipecah menjadi aplikasi *microservice* untuk memastikan kesesuaian pengembangan.

I.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan memberikan manfaat, baik secara teoritis maupun secara praktis, diantaranya sebagai berikut:

a. Manfaat bagi Fakultas Rekayasa Industri adalah terciptanya aplikasi *backend* berbasis *microservice* yang mendukung penyesuaian sistem serta pemisahan fungsi-fungsi modul pendaftaran dan penjadwalan aplikasi SOFI.

b. Manfaat bagi mahasiswa tingkat akhir, dengan migrasi arsitektur *microservice* proses pendaftaran dan penjadwalan menjadi lebih tangguh.

c. Manfaat bagi peneliti, diharapkan proyek migrasi ini memberikan kesempatan dalam mengembangkan keterampilan teknis yang berharga dalam merancang serta mengimplementasikan arsitektur *microservice.*

I.5 Batasan Masalah

Untuk mencapai tujuan yang telah ditentukan, maka permasalahan dibatasi kepada hal-hal berikut:

a. Aplikasi *backend* dibangun menggunakan bahasa Golang dengan *framework* Fiber.
b. Integrasi layanan menggunakan REST API.
c. Aplikasi hanya menggunakan relasional *database* yaitu MySQL.
d. Proses migrasi hanya dilakukan pada modul pendaftaran dan penjadwalan.

I.6 Sistematika Penulisan

Penelitian ini diuraikan dengan sistematika penulisan sebagai berikut:

Bab I Pendahuluan

Pada bab ini berisi mengenai latar belakang penelitian, masalah yang ada pada penelitian, tujuan penelitian, batasan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini memuat penjelasan tentang studi literatur yang relevan dengan permasalahan yang dibahas serta penelitian-penelitian terdahulu yang berkaitan dengan penelitian ini.

Bab III Metodelogi Penelitian

Pada bab ini diuraikan metode penelitian, sistematika dan analisis yang digunakan pada penelitian ini.

Bab IV Analisis dan Perancangan

Bab ini mencakup identifikasi sistem yang dilakukan migrasi meliputi analisis proses bisnis, pembuatan diagram serta analisis *Domain Driven Design.*

Bab V Implementasi dan Pengujian

Bab ini mencangkup pengembangan migrasi aplikasi sidang fakultas SOFI, yang dimana diuraikan dengan beberapa iterasi, serta terdapat juga pengujian sampai dengan tahap *deployment.*

Bab VI Kesimpulan dan Saran

Pada bab ini memaparkan kesimpulan dari penelitian yang dilakukan. Saran penelitian dikemukakan pada bab ini untuk penelitian selanjutnya.