

BAB I PENDAHULUAN

1.1 Latar Belakang

Outcome Based Education (OBE) adalah pendekatan pendidikan yang menitikberatkan pada pencapaian hasil belajar yang jelas dan terukur. Dalam OBE, institusi pendidikan merancang kurikulum, metode pengajaran, dan evaluasi yang secara langsung mendukung tujuan pembelajaran yang spesifik. Pendekatan ini memungkinkan pemantauan progres mahasiswa dengan lebih transparan dan objektif, mendorong mereka untuk lebih terlibat dalam pencapaian kompetensi yang relevan dengan kebutuhan dunia nyata. OBE juga mendukung institusi dalam menjaga tanggung jawab kepada para pemangku kepentingan, serta menjamin kualitas pendidikan yang lebih baik dan berkesinambungan [1].

Universitas Telkom telah mengimplementasikan sistem *Outcome Based Education (OBE)* yang difasilitasi oleh sebuah platform bernama OBE, yang terintegrasi dalam *superapps* iGracias dan dikembangkan menggunakan Bahasa Pemrograman PHP. Platform ini merupakan pengembangan dari aplikasi iGadis dan dirancang untuk memudahkan pengguna dalam melakukan input nilai dengan sistem penilaian berbasis *Course Learning Outcome (CLO)*. Aplikasi OBE ini mendukung pengelolaan *Program Educational Objectives (PEO)*, *Program Learning Outcome (PLO)*, *Course Learning Outcome (CLO)*, Mata Kuliah, Rencana Pengajaran Semester (RPS), Sub-CLO, serta manajemen *assessment tools*. Dengan mempertimbangkan bobot SKS tiap mata kuliah, aplikasi ini mengukur capaian PLO dan CLO mahasiswa secara akurat, serta menyediakan fitur tambahan seperti monitoring RPS dan pemetaan pengembangan pembelajaran [2].

Dengan meningkatnya lalu lintas penggunaan aplikasi iGracias yang didukung oleh lebih dari 30 ribu mahasiswa dan 1.000 dosen serta staf, seiring dengan semakin berkembangnya Telkom University, kebutuhan akan verifikasi, validasi, dan sinkronisasi data yang komprehensif menjadi sangat mendesak. Oleh karena itu dilakukanlah pembaruan pada aplikasi iGracias dengan mengembangkan platform ini menggunakan Angular framework. Pembaruan ini menghadirkan pembaruan teknologi untuk mendukung kinerja yang lebih optimal serta antarmuka pengguna yang lebih intuitif dan modern. Dengan demikian, pengguna dapat menikmati pengalaman yang lebih baik, dengan kecepatan akses dan efisiensi pengolahan data yang meningkat, sesuai dengan visi Telkom University dalam memberikan layanan digital berkualitas tinggi di lingkungan pendidikan.

Dalam proyek ini, penulis bertugas sebagai pengembang *Frontend* yang bertanggung jawab dalam membangun antarmuka pengguna berdasarkan desain yang telah disepakati. Proses ini dimulai dengan melakukan *slicing UI*, yaitu mengubah desain menjadi komponen-komponen *Frontend* yang interaktif dan responsif. Setelah itu,

penulis juga mengintegrasikan antarmuka tersebut dengan sistem backend melalui API untuk memastikan seluruh fungsionalitas dapat berjalan sesuai kebutuhan. Selain itu, penulis melakukan pengujian terhadap tampilan dan fungsionalitas yang telah dikembangkan guna memastikan sistem bekerja dengan baik, bebas dari bug, dan sesuai dengan kebutuhan pengguna.

1.2 Rumusan Masalah dan Solusi

Berdasarkan latar belakang yang diatas, maka rumusan masalah yang akan dibahas adalah sebagai berikut:

1. Bagaimana pendekatan desain antarmuka dalam migrasi sistem OBE dari PHP ke Angular dapat diterapkan untuk menghasilkan tampilan yang lebih konsisten, modern, dan sesuai dengan standar visual saat ini?

Solusi: Migrasi akan menggunakan pendekatan berbasis komponen diterapkan secara menyeluruh, melalui penggunaan Angular Component Architecture, setiap elemen antarmuka dirancang sebagai unit mandiri yang dapat digunakan kembali. Desain visual digabungkan dengan penerapan sistem *styling* berbasis SCSS dan penggunaan prinsip desain responsif, sehingga tampilan aplikasi tetap konsisten di berbagai perangkat. Selain itu, pengelolaan tema dan layout diatur dengan menggunakan warna utama Telkom University, yaitu merah dengan kode hexadecimal #E01A3F, untuk memastikan keseragaman visual di seluruh halaman aplikasi.

2. Bagaimana cara meningkatkan pengalaman pengguna dalam sistem OBE saat migrasi dari PHP ke Angular agar lebih intuitif, efisien, dan mudah digunakan?

Solusi: Untuk meningkatkan pengalaman pengguna, penerapan modularisasi fitur melalui Angular Modules membantu dalam membagi aplikasi menjadi bagian-bagian yang lebih ringan dan mudah diakses. Dengan dukungan fitur *dependency injection* dari Angular, layanan-layanan seperti autentikasi, navigasi, dan pengelolaan data dari api dapat diintegrasikan dengan cara yang lebih efisien dan fleksibel. Alur navigasi diperbaiki dengan memanfaatkan Angular Router yang memungkinkan pemetaan rute secara dinamis dan terstruktur. Performa aplikasi juga ditingkatkan dengan strategi optimasi pemuatan komponen secara *lazy loading*, sehingga mempercepat waktu akses pengguna ke fitur-fitur utama.

1.3 Tujuan

Berdasarkan rumusan masalah yang ada, tujuan yang akan dicapai adalah:

1. Untuk menerapkan pendekatan desain antarmuka yang sesuai dalam proses migrasi sistem OBE dari PHP ke Angular guna menghasilkan tampilan yang lebih modern, konsisten, dan sesuai dengan standar visual terkini.
2. Untuk meningkatkan kualitas pengalaman pengguna dalam sistem OBE melalui pengembangan fitur-fitur baru serta perbaikan alur interaksi, kemudahan navigasi, dan efisiensi penggunaan saat sistem dimigrasikan ke Angular.

1.4 Batasan Masalah

Batasan masalah dalam pembuatan aplikasi ini adalah:

1. Penulis hanya berfokus pada proses migrasi pada bagian Menu *Subject Coordinator* dari sistem lama ke sistem baru, tanpa membahas migrasi menu lain yang ada dalam sistem secara keseluruhan.
2. Lingkup pengembangan yang dilakukan penulis hanya mencakup pengembangan pada sisi *Frontend*, yaitu tampilan antarmuka pengguna dan interaksi pengguna dengan sistem, tanpa melakukan perubahan pada sisi backend atau database.
3. Aplikasi yang dikembangkan hanya ditujukan untuk platform web, sehingga tidak mencakup pengembangan aplikasi untuk perangkat mobile seperti Android atau iOS.

1.5 Penjadwalan Kerja

Berikut adalah timeline dalam migrasi aplikasi OBE yang dapat dilihat pada Tabel 1.1 dan 1.2:

Tabel 1. 1 Tabel timeline migrasi Aplikasi OBE

No	Deskripsi Kerja	Juli 2024				Agustus 2024				September 2024				Oktober 2024			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Onboarding																
2	Pengenalan Project OBE																
3	Mempelajari Angular framework																
4	Eksplorasi kode sumber																
5	Menganalisis kebutuhan pengembangan yang berkaitan dengan menu Sub-CLO serta RPS																

Tabel 1. 2 Tabel lanjutan timeline migrasi aplikasi OBE

No	Deskripsi Kerja	November 2024				Desember 2024				Januari 2025				Februari 2025			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Melakukan persiapan pengembangan dengan menginstal seluruh library yang diperlukan serta menyiapkan <i>environment variable</i> .																
2	<i>Slicing</i> menu Sub-CLO																
3	Integrasi API menu Sub-CLO																
4	<i>Slicing</i> menu Sub-CLO																
5	Integrasi API menu Sub-CLO																