

Pengembangan Aplikasi Presentia Menggunakan ReactJS dengan Pendekatan Single Page Application (SPA) dan Penerapan Lazy Loading

Muhammad Zaky Fathurahim¹, Novian Anggis Suwastika², Isa Mulia Insan³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹mzakyf@student.telkomuniversity.ac.id, ²anggis@telkomuniversity.ac.id,

³isamuliainsan@telkomuniversity.ac.id

Abstrak

Efektivitas pengelolaan presensi siswa merupakan aspek penting dalam kelancaran proses pendidikan. Metode absensi manual masih umum digunakan di sekolah dan rentan terhadap kehilangan data serta penyalahgunaan. Untuk mengatasi hal tersebut, dikembangkan aplikasi Presentia sebagai solusi digital berbasis web. Presentia dibangun menggunakan pendekatan *Single Page Application* (SPA) dengan ReactJS untuk mendukung interaksi pengguna yang dinamis tanpa perlu memuat ulang halaman. Namun, SPA memiliki tantangan berupa waktu muat awal yang tinggi, sehingga diterapkan teknik *Lazy Loading* menggunakan “lazy()” dan “Suspense”. Karena aplikasi digunakan oleh berbagai *role* dengan hak akses berbeda, diterapkan juga mekanisme *Role-Based Access Control* (RBAC) melalui logika *conditional rendering*. Penerapan ini terbukti membatasi akses pengguna sesuai *role* yang dimiliki, dan telah memenuhi kebutuhan NFR-04. Sebanyak 125 dari 127 kebutuhan fungsional berhasil direalisasikan berdasarkan UI/UX, sementara dua lainnya tidak memiliki antarmuka pengguna. Pengujian fungsional menggunakan *End-to-End testing* dengan Cypress dan pengujian *load time* menggunakan DevTools. Karena keterbatasan waktu, *role* yang memiliki fungsi serupa digabung ke dalam satu skenario untuk mempercepat pelaksanaan pengujian fungsional menggunakan pendekatan *Equivalence Partitioning* (EQP), sehingga sebanyak 48 *test case* berhasil dikembangkan dan dijalankan. Selain itu, *Lazy Loading* terbukti menurunkan *load time* sebesar 24.09% dengan waktu muat konsisten di bawah 1 detik sehingga memenuhi kebutuhan NFR-01.

Kata kunci: presensi siswa, *single page application*, reactjs, *lazy loading*, pengembangan *frontend*, pengujian *frontend*,
