

1. Pendahuluan

1.1. Latar Belakang

Fraud atau kecurangan adalah suatu tindakan penipuan yang dilakukan secara sengaja dan direncanakan dengan tujuan memperoleh keuntungan dengan cara merampas hak orang lain baik secara langsung maupun tidak langsung [1]. Kecurangan dapat terjadi ketika seseorang memiliki kesempatan untuk melakukan hal tersebut. Semakin tinggi kesempatan untuk melakukan kecurangan, maka semakin tinggi juga peluang seseorang melakukan kecurangan [2].

Association of Certified Fraud Examiners (ACFE) membagi *fraud* kedalam 3 kategori utama yaitu, kecurangan laporan keuangan, penyimpangan aset dan korupsi [3]. Berdasarkan survei yang dilakukan oleh ACFE Indonesia *Chapter*, terungkap bahwa jenis *fraud* yang paling sering terjadi di Indonesia adalah korupsi dengan persentase 64,4%, diikuti oleh penyalahgunaan aset atau kekayaan negara dan perusahaan sebesar 28,9%, sementara *fraud* laporan keuangan hanya mencapai 6,7% [4]. Walaupun persentase terjadinya kecurangan laporan keuangan paling kecil, namun tetap saja kecurangan dapat menyebabkan kerugian finansial dan reputasi bagi perusahaan, skenario terburuknya adalah bangkrutnya perusahaan tersebut.

Melihat besarnya kerugian yang dapat disebabkan oleh kecurangan laporan keuangan, penting adanya sebuah sistem untuk mendeteksi dan mencegah tindakan tersebut pada suatu instansi/perusahaan. Maka dari itu, dilakukan pengembangan aplikasi Fraud Deterrence Propeller (FDP) didasarkan pada dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) dimana terdapat *Functional Requirement* (FR) dan *Non-functional Requirement* (NFR) yang harus diimplementasi [5]. Aplikasi FDP dibangun dengan pendekatan berbasis *website* sehingga memiliki kemampuan lintas *platform*, memungkinkan pengguna dapat mengaksesnya dari berbagai perangkat [6]. Hal ini memungkinkan tim Internal untuk secara proaktif memantau, menganalisis dan menindaklanjuti aktivitas yang mencurigakan secara *real-time* [7].

Kemudian penting untuk mempertimbangkan *volume* data yang terlibat dalam analisis keuangan. Laporan keuangan sering kali melibatkan data dengan jumlah yang besar, sehingga *load time* halaman pada *website* menjadi krusial. Oleh karena itu, peran seorang *Frontend Developer* menjadi krusial untuk memastikan tampilan yang responsif dan performa *website* yang optimal sehingga nyaman digunakan oleh pengguna.

Sebagai solusi untuk mengatasi masalah tersebut, metode yang digunakan pada penelitian ini adalah *Client-Side Rendering* pada *Framework Next.js* [8]. *Next.js* adalah kerangka kerja atau *framework JavaScript* yang digunakan untuk membangun aplikasi web berbasis *React* [9]. *Next.js* mempunyai fitur seperti *Client-Side Rendering* (CSR) dan *Server-Side Rendering* (SSR), meskipun begitu CSR dipilih karena dapat memberikan pengalaman lebih cepat saat *load* halaman dan dinamis saat berinteraksi dengan pengguna [10], [11]. Berdasarkan penelitian H. A. Jartarghar, CSR terbukti lebih cepat dibandingkan dengan SSR saat memuat halaman *Category*. Dalam hasilnya, halaman yang menggunakan CSR mendapatkan *load time* sebesar 250ms sedangkan pada SSR sebesar 593ms [12].

Optimisasi *load time* menjadi sangat penting mengingat aplikasi ini digunakan untuk menangani data sensitif seperti laporan keuangan. Pengalaman pengguna yang lambat dalam memuat halaman dapat berdampak pada efektivitas pekerjaan. *Load time test* dilakukan untuk mengukur seberapa cepat waktu yang dibutuhkan untuk memuat seluruh konten kedalam halaman *website* yang menggunakan CSR [13]. Menurut Pourghassemi, berdasarkan penelitian yang dilakukan oleh *Google* menunjukkan bahwa, 53% pengguna *website* pada perangkat *mobile* meninggalkan halaman yang membutuhkan *load time* lebih dari 3 detik [13].

Load time test dilakukan dengan menggunakan fitur *Network* pada *DevTools* yang tersedia pada *browser Google Chrome* dan *Microsoft Edge*. *Google Chrome* dipilih karena menurut data dari *website Similarweb*, *Google Chrome* menempati urutan pertama sebagai *browser* yang paling populer di seluruh dunia pada bulan Juni 2024 dengan *market share* sebesar 64,43% [14]. Hal ini diperkuat oleh penelitian Mohamed yang menyatakan bahwa *Google Chrome* merupakan *browser* yang paling banyak digunakan dengan tingkat penggunaan lebih dari 65% [15].

1.2. Topik dan Batasannya

Berdasarkan latar belakang yang telah dipaparkan, topik atau rumusan masalah yang dibahas pada tugas akhir ini yaitu :

1. Bagaimana penerapan *Client-Side Rendering* pada proses pengembangan aplikasi Fraud Deterrence Propeller (FDP) menggunakan *framework Next.js*?
2. Bagaimana pengaruh penerapan *Client-Side Rendering* terhadap performa aplikasi FDP?

Adapun beberapa batasan masalah yang perlu difokuskan pada penelitian ini, yaitu:

1. Penelitian ini dilaksanakan secara berkelompok sebagai bagian dari TA *Capstone* dengan peneliti berperan sebagai *Frontend Developer*.
2. Pembangunan aplikasi FDP hanya dari sisi *Frontend* atau tampilan depan.
3. Pembangunan aplikasi FDP mengacu pada dokumen SKPL.
4. Pembangunan aplikasi FDP menggunakan *Framework JavaScript* yaitu *Next.js*.

5. Jenis pengujian yang dilakukan adalah *load time testing* pada *environment staging*, termasuk waktu yang diperlukan dari *user* melakukan *request API* hingga halaman *website* tampil sepenuhnya.
6. Pengujian dilakukan pada *browser Google Chrome* dan *Microsoft Edge*.

1.3. Tujuan

Tujuan dilakukannya penelitian ini adalah sebagai berikut :

1. Mengembangkan aplikasi Fraud Deterrence Propeller dari sisi *Frontend* menggunakan *Framework Next.js* dengan menerapkan *Client-Side Rendering*.
2. Melakukan pengukuran *performance* dengan cara *load time test* antar halaman terhadap penerapan *Client-Side Rendering* pada aplikasi Fraud Deterrence Propeller.