RINGKASAN

ANALISIS PENGARUH *CODE SPLITTING* TERHADAP WAKTU MUAT AWAL HALAMAN WEB MENGGUNAKAN REACTJS DAN VUEJS

Muhammad Ilham Isfadhillah

Waktu muat awal halaman adalah elemen krusial dalam pengembangan aplikasi web, khususnya di bagian *front end*, penurunan kecepatan sekecil 100 ms secara signifikan dapat menurunkan kemampuan web dalam mempertahankan pengguna dan mencapai targetnya (tingkat retensi dan konversi). Di sisi lain, penggunaan kerangka kerja modern seperti ReactJS dan VueJS dapat menghasilkan ukuran *bundle* JavaScript yang relatif besar dan berdampak negatif pada waktu muat awal. Teknik *code splitting* hadir sebagai solusi untuk memecah *bundle* menjadi bagian-bagian yang lebih kecil dan dimuat sesuai kebutuhan.

Penelitian ini menganalisis dampak dua teknik *code splitting*, yaitu *dynamic import* (dengan strategi *route-based code splitting* dan *component-based code splitting*) dan *chunk split* (dengan strategi ukuran maksimal *chunk*) terhadap waktu muat awal halaman pada aplikasi FindSoed. Pengujian dilakukan pada aplikasi tersebut yang dibangun dengan kombinasi kerangka kerja ReactJS dan VueJS dalam mode *Client Side Rendering* (CSR) dan *Server Side Rendering* (SSR) menggunakan *bundler* Rsbuild dan Vite. Kinerja diukur menggunakan perangkat lunak Lighthouse berdasarkan metrik *First Contentful Paint* (FCP), *Largest Contentful Paint* (LCP), *Total Blocking Time* (TBT), dan *Speed Index* (SI). Untuk membandingkan efektivitas strategi secara keseluruhan, diterapkan sebuah sistem skoring yang menghitung jumlah total keunggulan pada setiap metrik di seluruh kombinasi pengujian.

Hasil penelitian menunjukkan bahwa strategi *route-based code splitting* pada teknik *dynamic import* secara konsisten menjadi yang paling unggul, mengurangi ukuran JavaScript awal secara dratis (selisih hingga 400 kB) dan meningkatkan semua metrik kinerja. Sebaliknya, variasi pada strategi *chunk split* tidak menunjukkan pola keunggulan yang konsisten dan perbedaan ukuran JavaScript awal yang sangat kecil (selisih di bawah 1 kB), sehingga membuat hasil pengujian dapat dipengaruhi oleh faktor lain seperti latensi jaringan.

Kata Kunci: *Code Splitting*, Kerangka Kerja *Front End,* Waktu Muat Awal, Unjuk Kerja Web

SUMMARY

IMPACT ANALYSIS OF CODE SPLITTING ON INITIAL LOAD TIME OF WEB PAGES USING REACTJS AND VUEJS

Muhammad Ilham Isfadhillah

Initial page load time is a crucial element in front-end web application development, where a speed decrease of as little as 100 ms can significantly lower the web's ability to retain users and achieve its targets (user retention and conversion rates). On the other hand, the use of modern frameworks such as ReactJS and VueJS produces relatively large JavaScript bundle sizes, which negatively impact initial load times. The code splitting technique emerges as a solution to break down the bundle into smaller parts that are loaded on demand.

This research analyzes the impact of two code splitting techniques, dynamic import (with route-based and component-based strategies) and chunk split (with maximal chunk size strategies) on the initial page load time of the FindSoed application. The tests were conducted on that application built with a combination of the ReactJS and VueJS frameworks in both Client-Side Rendering (CSR) and Server-Side Rendering (SSR) modes, using Rsbuild and Vite bundlers. Web Performance was measured using Lighthouse software based on First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT), and Speed Index (SI) metrics. To compare the overall effectiveness of the strategies, a scoring system was implemented that calculates the total number of wins for each metric across all test combinations.

The research results show that the route-based code splitting strategy for dynamic import technique is consistently superior, reducing the initial JavaScript size drastically(with differences up to 400 kB) and improving all performance metrics. Conversely, variations in chunk split strategy did not show a consistent winning pattern and the very small differences in initial JavaScript size (with differences under 1 kB), which made the test results susceptible to other factors such as network latency.

Keywords: *Code Splitting, Front End Framework, Initial Load Time, Web Performance*