

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis terkait penelitian yang telah dilakukan didapatkan kesimpulan yang menjawab pertanyaan dari rumusan masalah, yaitu sebagai berikut.

1. Penerapan *code splitting* dengan teknik *dynamic import* pada ReactJS dan VueJS dengan konfigurasi CSR dan SSR terletak pada level kerangka kerja. ReactJS mengandalkan ekosistemnya yang kaya dengan memanfaatkan *library* pihak ketiga yaitu *@loadable/component*. Sebaliknya, VueJS mengintegrasikan kemampuan ini sebagai fitur bawaan yang sudah lengkap dengan *defineAsyncComponent*. Meskipun caranya berbeda, pola dasarnya sama, yaitu dengan membungkus komponen dalam *dynamic import* dan menyediakan mekanisme *fallback*. Teknik sekunder seperti *chunk split* kemudian diterapkan bukan pada level kode, melainkan pada level konfigurasi *bundler* (Rsbuild) dengan mengatur ukuran maksimal *chunk* untuk memberikan kontrol yang lebih granular.
2. Teknik *dynamic import* dengan strategi *route-based code splitting* (R-CS) terbukti secara konsisten menghasilkan waktu muat awal halaman yang lebih cepat dibandingkan *component-based code splitting* dan tanpa *code splitting*. Penerapan R-CS mampu meningkatkan metrik FCP sebanyak 70%, metrik LCP sebanyak 46%, metrik TBT sebanyak 57% dan SI sebanyak 62% lebih cepat. R-CS bekerja pada level makro dengan

memisahkan kode berdasarkan rute, yang secara drastis mengurangi ukuran *file JavaScript* yang perlu dimuat di awal. Perbedaan ukuran *bundle* yang signifikan (rata rata 400 kB) membuat faktor seperti latensi jaringan tidak terlalu berpengaruh.

3. Teknik *chunk split* tidak menunjukkan satu strategi yang unggul secara konsisten. Perbedaan ukuran *chunk* yang dihasilkan antara variasi ukuran maksimal yang lebih besar sangatlah kecil (dibawah 1 kB) sehingga faktor seperti latensi jaringan dapat menjadi faktor dominan dalam menentukan hasil metrik.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dilakukan untuk penelitian selanjutnya, yaitu sebagai berikut.

1. Penelitian selanjutnya dapat menggunakan SSR dengan teknik *streaming* pada VueJS dan ReactJS yang dapat mengirimkan HTML secara langsung setiap *byte*-nya ke klien tanpa menunggu HTML siap secara penuh dari *server*.
2. Membandingkan kinerja *code splitting* dengan *custom server* yang dibuat dengan *bundler* Vite atau Rsbuild dengan *meta framework* matang yang menggunakan SSR sebagai *rendering* utamanya seperti NuxtJS dan NextJS.
3. Mengkombinasikan teknik optimisasi lain yaitu *preload* untuk mengunduh *chunk* secara pararel ketika ukuran *chunk* menjadi lebih banyak dengan mengurangi nilai *max size* pada teknik *chunk split* dengan *bundler* Rsbuild.