

BAB V

PENUTUP

5.1 Simpulan

Simpulan dari penggunaan teknologi *React Js* dan *Vite* sebagai *build tool* terhadap pengembangan *Front End* pada sistem informasi manajemen kepagawaian adalah sebagai berikut :

1. *Rendering* data dengan *Virtual DOM*

Hasil pengujian menunjukkan bahwa *React.js* memanfaatkan teknologi *Virtual DOM* untuk mempercepat proses rendering data pada antarmuka *front-end* dan mengurangi beban pada kinerja browser. Keunggulan ini sangat signifikan, terutama ketika data yang ditampilkan bersifat kompleks atau dalam jumlah besar, sehingga dapat memastikan pengalaman pengguna yang responsif dan optimal.

2. *Reusable Component*.

Berdasarkan hasil pengujian, dapat disimpulkan bahwa penggunaan *reusable component* dalam pengembangan antarmuka berjalan dengan baik. Komponen-komponen yang dirancang untuk dapat digunakan kembali mampu berfungsi sesuai harapan tanpa menimbulkan *error* maupun bug. Pendekatan ini tidak hanya meningkatkan efisiensi pengembangan, tetapi juga memastikan konsistensi desain dan perilaku di seluruh aplikasi.

3. Performa Sistem

Hasil pengujian menunjukkan bahwa performa sistem tetap sangat baik meskipun jumlah data yang dirender cukup besar. Hal ini didukung oleh implementasi *lazy loading*, yang memungkinkan komponen atau data hanya dimuat saat dibutuhkan, sehingga mengurangi beban awal pada aplikasi.

5.2 Saran

Berdasarkan hasil pengujian dan simpulan terhadap pengembangan aplikasi kepegawaian menggunakan *react js* dan *vite* sebagai *build tool*, maka saran yang dapat diberikan adalah sebagai berikut:

1. Optimalisasi Penggunaan *Virtual DOM*

Untuk memaksimalkan kinerja yang diberikan oleh *Virtual DOM*, pengembang disarankan untuk mengelola *state* dengan lebih hati-hati. Gunakan *state management* seperti *Redux* atau *Zustand* untuk menghindari pembaruan *state* yang tidak diperlukan, sehingga proses *rendering* tetap optimal dan tidak mengganggu pengalaman pengguna.

2. Peningkatan Konsistensi *Reusable Component*

Dalam memanfaatkan *reusable component*, pengembang disarankan untuk membuat *design system* atau pustaka komponen yang terdokumentasi dengan baik. Hal ini akan mempermudah kolaborasi tim dalam pengembangan fitur baru dan memastikan standar desain yang seragam di seluruh sistem.

3. Performa Sistem yang Lebih Baik

Meskipun *Vite* memberikan performa unggul, pengembang disarankan untuk terus memonitor dan mengoptimalkan ukuran bundle aplikasi, misalnya dengan memanfaatkan fitur *code splitting* atau *lazy loading*. Selain itu, integrasi alat pemantauan performa seperti *Lighthouse* atau *Web Vitals* dapat membantu dalam mengidentifikasi area yang memerlukan perbaikan lebih lanjut.

4. Pengujian dan Pemeliharaan Sistem

Untuk memastikan keberlanjutan kualitas sistem, pengujian *end-to-end* (E2E) seperti *Cypress* atau *Playwright* disarankan diterapkan secara rutin. Selain itu, pemeliharaan sistem harus mencakup pembaruan berkala terhadap dependensi *React.js* dan *Vite* untuk memanfaatkan fitur terbaru serta memperbaiki potensi kerentanan keamanan.