

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai perbandingan performa antara React Native dan Flutter dalam pengembangan aplikasi kasir restoran berbasis mobile, dapat disimpulkan beberapa hal sebagai berikut:

1. Perbandingan performa React Native dan Flutter menunjukkan bahwa kedua framework memiliki karakteristik performa yang berbeda pada setiap metrik pengujian. Pada aspek *load time*, React Native memiliki waktu muat awal aplikasi yang lebih cepat dibandingkan Flutter. Pada perangkat emulator mendapatkan nilai 617,6 ms (React Native) dan 1.942,8 ms (Flutter) dan pada perangkat fisik adalah 1.489,4 ms (React Native) dan 2.625,5 ms (Flutter). Namun pada aspek penggunaan sumber daya saat aplikasi berjalan, Flutter cenderung lebih efisien. Hal ini terlihat dari rata-rata penggunaan CPU pada emulator sebesar 42,93% (Flutter) dibandingkan 57,1% (React Native), serta pada perangkat fisik 19,44% (Flutter) dan 28,16% (React Native). Dari sisi memori, Flutter juga menunjukkan konsumsi yang lebih rendah dengan rata-rata 187,39 MB pada emulator dan 358,09 MB pada perangkat fisik, sedangkan React Native mencapai 267,15 MB pada emulator dan 428,27 MB pada perangkat fisik. Perbedaan ini dipengaruhi oleh arsitektur internal framework, mekanisme rendering, serta ukuran hasil *build* aplikasi di mana React Native memiliki ukuran 172 MB dan Flutter 52 MB, yang turut berkontribusi terhadap kebutuhan sumber daya saat aplikasi dijalankan.
2. Framework yang lebih optimal untuk pengembangan aplikasi kasir restoran berbasis mobile dalam penelitian ini adalah Flutter, khususnya apabila prioritas utama terletak pada efisiensi penggunaan CPU dan memori selama aplikasi beroperasi. Flutter menunjukkan konsumsi sumber daya yang lebih rendah dan ukuran aplikasi yang lebih kecil

sehingga lebih ringan dalam distribusi dan instalasi. Meskipun demikian, React Native tetap menjadi alternatif yang baik apabila kebutuhan aplikasi lebih menekankan pada kecepatan waktu muat awal (*startup time*) dan fleksibilitas integrasi dengan ekosistem JavaScript. Dengan demikian, pemilihan framework tidak bersifat mutlak, melainkan perlu disesuaikan dengan prioritas kebutuhan sistem yang akan dikembangkan.

5.2. Saran

Berdasarkan hasil penelitian dan analisis performa yang telah dilakukan, maka beberapa saran yang dapat diberikan adalah sebagai berikut:

1. Analisis performa dapat ditingkatkan dengan menambahkan variasi skenario penggunaan aplikasi yang lebih beragam, seperti jumlah item transaksi yang lebih besar, frekuensi navigasi antar halaman yang lebih intensif, serta kombinasi aktivitas pengguna dalam satu sesi aplikasi. Hal ini bertujuan untuk memberikan gambaran performa yang lebih komprehensif terhadap kondisi penggunaan aplikasi kasir di lingkungan operasional nyata.
2. Pengujian performa pada penelitian ini telah dilakukan menggunakan dua jenis perangkat dengan spesifikasi berbeda. Untuk meningkatkan kedalaman analisis, penelitian selanjutnya dapat mempertimbangkan penggunaan lebih banyak variasi perangkat dengan tingkat spesifikasi yang berbeda, sehingga hasil perbandingan performa dapat merepresentasikan lebih luas kondisi perangkat pengguna di lapangan.
3. Penelitian selanjutnya dapat mempertimbangkan penerapan teknik optimasi yang sesuai dengan karakteristik masing-masing framework, seperti pengelolaan state, struktur komponen, dan manajemen rendering antarmuka, sehingga analisis performa tidak hanya bersifat perbandingan hasil akhir, tetapi juga mencerminkan penerapan praktik pengembangan yang optimal.