

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Berdasarkan hasil penelitian dan analisis komparasi yang telah dilakukan antara *framework* Flutter dan Ionic Angular dalam pengembangan aplikasi *mobile* "Lelang Agro", dapat ditarik kesimpulan sebagai berikut untuk menjawab rumusan masalah:

##### 1. Komparasi Performa (*Quantitative Performance*)

Secara umum, *framework* Flutter menunjukkan superioritas performa dibandingkan Ionic Angular pada mayoritas parameter pengujian:

- a. Responsivitas (TTI): Flutter memiliki waktu inisialisasi yang lebih cepat dengan rata-rata TTI sebesar 602 ms, sedangkan Ionic Angular membutuhkan waktu 859 ms. Selisih sebesar 257 ms (29,9%) ini disebabkan oleh arsitektur *native* Flutter yang dapat langsung *me-render* UI tanpa melalui lapisan WebView seperti pada Ionic.
- b. Kelancaran Visual (FPS): Flutter mampu mempertahankan stabilitas frame rate yang lebih baik dengan rata-rata 56,66 FPS, sedikit lebih unggul dibandingkan Ionic Angular yang mencatatkan 54,64 FPS. Selisih 2,02 FPS (3,7%) ini menunjukkan keunggulan Skia Flutter dalam *me-render* animasi *scroll* tanpa bergantung pada performa WebView sistem.
- c. Efisiensi Sumber Daya: Flutter terbukti lebih hemat sumber daya dengan rata-rata penggunaan memori (RAM) sebesar 156,28 MB dan penggunaan CPU sebesar 17,61%. Sebaliknya, Ionic Angular mencatatkan konsumsi RAM rata-rata 184,80 MB (lebih tinggi 18,3%) dan penggunaan CPU 28,06% (lebih tinggi 59,3%) akibat *overhead* dari proses *bridging* JavaScript ke *native layer*.

- d. Ukuran Aplikasi: Ionic Angular unggul signifikan dalam efisiensi penyimpanan dengan ukuran berkas APK sebesar 8,1 MB, sedangkan Flutter menghasilkan berkas sebesar 23,5 MB (hampir 3x lebih besar). Perbedaan ini dikarenakan Ionic memanfaatkan komponen sistem WebView yang sudah tertanam di perangkat, sementara Flutter harus menyertakan Skia dan Dart *Runtime* secara mandiri di dalam paket aplikasi.

Tabel 21. Ringkasan Perbandingan Performa

Parameter	Flutter	Ionic Angular	Unggul	Selisih
TTI	602 ms	859 ms	Flutter	29,9% lebih cepat
FPS	56,66	54,64	Flutter	3,7% lebih tinggi
Memori	156,28 MB	184,80 MB	Flutter	15,4% lebih hemat
CPU	17,61%	28,06%	Flutter	37,2% lebih hemat
APK size	23,5 MB	8,1 MB	Ionic	65,5% lebih kecil

## 2. Tingkat Kustomisasi

Ionic Angular memiliki keunggulan dalam aspek struktur proyek dan konfigurasi dengan tingkat kompetensi *Mature*. Standarisasi struktur yang ketat menggunakan konvensi Angular serta penggunaan CLI (*Command Line Interface*) memudahkan manajemen proyek dan konfigurasi sistem tanpa perlu menyentuh lapisan *native* secara mendalam.

Di sisi lain, Flutter memiliki tingkat kompetensi *Capable* pada struktur dan *Aware* pada konfigurasi. Pendekatan ini memberikan fleksibilitas tinggi namun menuntut pengembang untuk melakukan pengaturan manual pada level *native* (Android/iOS) dan membangun arsitektur folder secara mandiri.

Untuk aspek komponen UI, kedua *framework* menunjukkan kemampuan setara pada level *Mature* dengan pendekatan yang berbeda, Angular menggunakan *Component-Based Architecture*, sementara Flutter menggunakan *Widget Composition*.

### 3. Dukungan Pengembang (*Developer Support*)

Flutter memiliki momentum pertumbuhan ekosistem yang lebih positif dengan tren penggunaan yang meningkat (42%) dan dokumentasi terpusat yang sangat terstruktur, memudahkan proses pembelajaran dan pemecahan masalah. Sementara itu, Ionic Angular mengalami tren penurunan (12%), namun tetap memiliki dukungan komunitas yang luas berkat integrasinya dengan ekosistem besar pengembangan *web* (NPM dan JavaScript).

### 4. Rekomendasi Implementasi

- a. Flutter direkomendasikan jika prioritas utama pengembangan adalah performa aplikasi yang tinggi dan pengalaman pengguna (*User Experience*) yang mulus. Framework ini unggul signifikan dalam kecepatan, stabilitas visual, serta efisiensi penggunaan CPU dan memori, menjadikannya pilihan terbaik untuk aplikasi yang menuntut interaksi intensif dan responsivitas tinggi.
- b. Ionic Angular direkomendasikan jika prioritas utama adalah efisiensi penyimpanan dan kemudahan konfigurasi. Framework ini terbukti jauh lebih ringan dengan ukuran aplikasi lebih kecil dibandingkan Flutter, serta menawarkan struktur proyek yang lebih mapan (*Mature*) berbasis standar *web*. Hal ini menjadikan Ionic solusi yang lebih relevan untuk perangkat dengan penyimpanan

terbatas atau untuk tim pengembang yang ingin mempercepat siklus pengembangan memanfaatkan ekosistem *web* yang sudah ada.

## 5.2. Saran

Berdasarkan keterbatasan penelitian dan temuan, peneliti mengajukan beberapa saran untuk pengembangan selanjutnya:

1. Pengujian Lintas Platform (iOS): Penelitian ini membatasi pengujian performa hanya sistem operasi Android. Disarankan bagi peneliti selanjutnya untuk melakukan pengujian komparasi pada perangkat iOS untuk memvalidasi klaim konsistensi performa lintas platform kedua *framework*.
2. Variasi Spesifikasi Perangkat: Pengujian dalam penelitian ini menggunakan perangkat *mid-range* (Snapdragon 732G). Mengingat target pengguna aplikasi pertanian yang beragam, disarankan untuk melakukan pengujian pada perangkat *low-end* (spesifikasi rendah) untuk melihat batas toleransi performa kedua *framework* dalam kondisi sumber daya yang sangat terbatas.
3. Optimasi Ukuran Aplikasi Flutter: Bagi pengembang yang memilih Flutter, disarankan untuk menerapkan teknik optimasi ukuran lanjutan seperti penggunaan *Android App Bundle* (.aab) atau *proguarding* yang lebih agresif untuk meminimalisir kesenjangan ukuran aplikasi dengan solusi *hybrid*.