## 1. Pendahuluan

### Latar Belakang

Perkembangan perangkat lunak berbasis android sangatlah cepat. Tercatat terdapat sebanyak 2.591.578 perangkat lunak yang tersedia di *play store* pada Maret 2022 [1]. Kerangka kerja ataupun *framework* untuk mengembangkan perangkat lunak berbasis android pun banyak bermunculan, salah satunya adalah Flutter. Flutter merupakan *framework multi-platform* yang dirilis oleh *google* dan bersifat *open source* [2]. Pengembangan perangkat lunak menggunakan Flutter terdiri dari beberapa komponen UI (*user interface*) yang disebut dengan *widget*. *Widget* akan memberikan luaran berupa tampilan yang sesuai dengan konfigurasi dan *state* saat ini. *State* merupakan informasi yang dapat dibaca secara sinkronus ketika *widget* dibangun dan dapat berubah selama *widget* tersebut aktif [17]. Pada saat *state* suatu *widget* berubah, maka *widget* akan membangun ulang untuk menampilkan kondisi yang sesuai dengan *state* terbaru [3].

Pada UI suatu perangkat lunak, biasanya terdiri dari beberapa *widgets* yang dapat menyebabkan banyak terjadinya perubahan *state*. Ketika terdapat banyak *widget* atau semakin kompleks suatu perangkat lunak, maka akan menyulitkan proses pelacakan dan pembaharuan *state*. Berdasarkan hal tersebut, para *developer* menyadari bahwa diperlukan suatu cara untuk mengelola *state* (*State Management*). *State management* merupakan cara untuk mengelola *state* dari UI *controls* seperti *text fields, radio button,* dan lainnya [17].

Menurut penelitian yang dilakukan sebelumnya [5], menunjukkan bahwa penggunaan *state management* yang berbeda dapat menghasilkan perbedaan performa pada suatu perangkat lunak. Performa pada perangkat lunak berbasis android merupakan aspek yang penting, karena dapat menggambarkan bagaimana cara kerja, efisiensi, dan kenyamanan dalam penggunaannya [14]. Performa perangkat lunak yang buruk dapat mempengaruhi *responsiveness, launch time,* CPU *consumption, memory consumption,* dan *energy consumption* pada suatu perangkat android [18].

Penerapan *state management* dilakukan di beberapa *framework*, terutama pada pengembangan android salah satunya adalah Flutter. Pada Flutter terdapat beberapa *state management library* yang paling banyak disukai oleh *developer*, antara lain, GetX, Provider, dan BloC [4]. Hal ini dapat dilihat dari jumlah *likes* dan *stars* pada laman resmi *library* tersebut. GetX dengan 8475 *likes* [6] dan 6300 *stars* [7], Provider dengan 6323 *likes* [8] dan 4200 *stars* [9], dan BloC dengan 3736 *likes* [10] dan 8800 *stars* [11].

Pada penelitian sebelumnya [5] dilakukan pengujian performa terhadap Provider dan BloC yang dibandingkan dengan setState() sebagai *state management default* pada Flutter. Penelitian tersebut menunjukkan bahwa penggunaan BLoC menghasilkan performa yang lebih baik dibandingkan dengan Provider dan setState(). Namun penelitian tersebut belum melakukan pengujian terhadap GetX sebagai *state management* yang paling banyak disukai oleh *developer*. Oleh karena itu perlu dilakukan pengujian terhadap GetX, sehingga dapat diketahui performansi dari ketiga *state management* yang paling banyak disukai oleh *developer*. Dikarenakan BLoC memiliki performa yang lebih baik jika dibandingkan dengan Provider [5], maka pada penelitian ini akan dilakukan pengujian performa terhadap GetX yang akan dibandingkan dengan Blo C *state management*.

GetX merupakan *state management* yang ringan, *powerful*, dan *high performance* [20]. Namun pada penelitian yang dilakukan oleh Dmitrii Slepnev [12] menyebutkan bahwa penggunaan GetX yang kurang tepat dapat menimbulkan masalah performansi pada perangkat lunak yang dikembangkan. Penelitian tersebut menyebutkan bahwa penggunaan GetX kurang cocok pada perangkat lunak yang kompleks, namun belum melakukan pengujian performansinya. Hal ini menimbulkan suatu permasalahan baru yaitu kelebihan dari GetX yang diberikan pada laman resmi GetX [20] bertolak belakang dengan penelitian yang dilakukan oleh Dmitrii Slepnev [12] dalam performansi GetX. Untuk mengatasi hal tersebut, perlu dilakukan pengujian dan analisis performansi GetX dan BLoC. Tingkat kompleksitas perangkat lunak yang dimaksud dilihat dari jumlah *widgets*, jumlah data yang diakses, dan alur kerjanya.

### Topik dan Batasannya

Topik pada penelitian ini adalah melakukan pengujian performa terhadap GetX dan BLoC *state management library* pada Flutter untuk perangkat lunak berbasis android menggunakan metode *Performance Testing*. Pengujian dilakukan dengan cara mengimplementasikan kedua *state management* tersebut pada perangkat lunak menggunakan *framework* Flutter dan bahasa pemrograman dart. Perangkat lunak yang dikembangkan menerapkan API (*application programming interface*) yang didapatkan dari laman api.tvmaze. Pada penelitian ini telah dilakukan pengujian sebanyak enam skenario, dimana setiap skenario memiliki jumlah data yang berbeda yaitu 50, 100, 240, 2400, 2.400.000, 4.800.000. Jumlah data ini dipilih agar dapat melihat performa perangkat lunak jika menerima jumlah data yang beragam. Kemudian pengujian dilakukan dengan cara mengukur *cpu usage*, *memory usage*, dan *energy consumption* pada perangkat lunak telah dikembangkan menggunakan *tools* Android Profiler pada Android Studio. Perangkat lunak diinstal pada *device* Samsung S9 (SM-G960F) dengan spesifikasi Android *System* versi 10 (Android Q), 4GB RAM, dan *chipset* Exynos 9810.

**Tujuan**
Tujuan dari penelitian ini adalah melakukan analisis performa GetX dan BLoC *state management library* jika diimplementasikan pada perangkat lunak berbasis android.

**Organisasi Tulisan**
Bagian selanjutnya yaitu bab 2, membahas mengenai studi yang berkaitan dengan penelitian yang telah dilakukan. Pada bab 3 membahas model dan sistem yang digunakan untuk penelitian. Pada bab 4 berisi mengenai hasil dan analisis hasil yang didapat, lalu kesimpulan akan ditampilkan pada bab 5.