1. Pendahuluan

1.1. Latar Belakang

Perkembangan Smartphone di era sistem informasi dan komunikasi modern, masyarakat sudah terbiasa menggunakan komputer dan perangkat lunaknya [1]. Selanjutnya penggunaan Aplikasi Seluler merupakan sektor baru dan berkembang pesat karena dapat digunakan secara fleksibel dan praktis [1]. Menurut Alisara Hincheranan dan Wanchai Rivepiboon [5], dikarenakan pertumbuhan aplikasi *mobile* yang cepat, dibutuhkan kinerja sistem yang tinggi untuk beradaptasi dengan perubahan lingkungan dan evolusi teknologi. Kinerja termasukaspek terpenting dalam keberhasilan sebuah aplikasi karena berkaitan dengan seberapa cepat sistem berjalan dan memuat *data* [2] [3]. Disebutkan pada standar ISO/IEC 25010 [4], bahwa untuk menghasilkan performansi yang baik pada aplikasi *mobile*, diperlukannya efisiensi terhadap elemen resource *CPU usage*, *memory usage* dan *execution time* atau seberapa cepat program dapat dieksekusi. Salah satu faktor yang mempengaruhi kinerja sistemperangkat lunak yaitu penerapan architecture pattern [6].

Dalam mengembangkan aplikasi mobile dengan kinerja yang baik, pola arsitektur merupakan hal penting yang harus diperhatikan karena aplikasi pasti memiliki karakteristik dan pendekatan yang berbeda-beda [6] [13]. Sehingga perlu adanya penelitian untuk membandingkan beberapa aspek dari arsitektur perangkat lunak untuk mengetahui kelebihan dan kekurangan masing-masing arsitektur [13].

Terdapat beberapa architecture pattern yang banyak muncul dalam pengembangan aplikasi *mobile* dan banyak digunakan dalam aplikasi berat GUI, diantaranya terdapat tiga arsitektur yaitu *Model View Controller* (MVC), *Model View Presenter* (MVP) dan *Model View ViewModel* (MVVM) [6]. Dari penelitian sebelumnya [7] menyatakan, penerapan design pattern dengan cara memilih architecture pattern yang sesuai dapat meningkatkan performansi aplikasi menjadi lebih baik berdasarkan penggunaan *CPU* dan penggunaan memory. Pada pengujian performansi yang telah dilakukan sebelumnya [6], arsitektur MVVM dan MVP memiliki kelebihan pada aspek testability, modifiability, dan performance dibandingkan arsitektur MVC. Arsitektur MVVM dan MVP jauh lebih unggul pada metrik *memory usage* dan *execution time* dibandingkan dengan arsitektur MVC [6].

Dari penelitian sebelumnya [8] menyatakan bahwa, arsitektur MVVM yang merupakan *framework* baru dari pembentukan arsitektur MVP, salah satu perbedaannya adalah jika MVVM memiliki Databinding yang berperan mengurangi kompleksitas kode. Arsitektur MVVM juga memiliki komponen LiveData yang terdapat pada ViewModel [8]. LiveData ini berguna untuk menghindari adanya crash yang menyebabkan perangkat lunak berhenti berfungsi dengan benar dan menyebabkan *force quit* yang disebabkan karena Activity yang berhenti [8].

Sehingga tujuan penelitian ini adalah menerapkan arsitektur MVVM pada aplikasi yang sebelumnya telah menerapkan arsitektur MVP dengan menggunakan Covid-19 API sebagai *media* pertukaran *data*. Lalu aplikasi akan diuji fungsionalitas dengan menggunakan metode Black Box Testing untuk menguji sistem apakah aplikasi berjalan lancar tanpa error ataupun crash. Selanjutnya akan dilakukan analisis menggunakan pendekatan Goal Question Metric (GQM Method) untuk mengetahui pengaruh penerapan arsitektur MVVM terhadap kinerja aplikasi.

1.2. Topik dan Batasannya

Topik dan batasan masalah pada penelitian ini adalah melakukan analisis pengaruh penerapan *architecture pattern* MVVM terhadap kinerja pada aplikasi *mobile*. Untuk pengujian fungsionalitas menggunakan metode Black Box testing dan untuk analisis metrik kinerja menggunakan metode Goal Question Metrik atau (GQM Method). Adapun batasan masalah pada penelitian ini, diantaranya pengujian kinerja hanya mencakup metrik *CPU usage*, *memory usage*, *execution time* dan *load time data* dari API dan dalam mengukur kinerja aplikasi dilakukan pada *emulator* yang ada di fitur Android Profiler yang ada di *platform* Android Studio. Selanjutnya batasan lainnya yaitu aplikasi menggunakan *dataset* JSON yang diambil dari COVID-19 API.