

BAB I. PENDAHULUAN

1.1 Latar Belakang

Antrian adalah fenomena yang terjadi ketika permintaan akan suatu layanan pada suatu waktu melebihi kapasitas layanan pada waktu yang sama. Masalah antrian sering ditemui dalam kehidupan sehari-hari baik di industri maupun dalam perdagangan, keuangan, sosial, dan lainnya. Antrian yang sangat panjang dan waktu pelayanan yang terlalu lama sangat menjengkelkan[1]. Masalah antrian ini sudah menjadi banyak ditemui di berbagai sektor industri seperti kuliner, kesehatan, perbankan, dan lain sebagainya. Masalah antrian dapat menyebabkan kendala serius karena menunggu dalam antrian akan menghabiskan waktu pelanggan. Pelanggan yang tidak mampu menunggu mungkin meninggalkan antrian dan menyebabkan kerugian bagi penyedia layanan. Salah satu masalah antrian paling umum terjadi pada antrian makanan, yang biasanya terjadi di restoran[2]. Bagi banyak organisasi, ketidakpuasan pelanggan terhadap waktu tunggu adalah masalah serius yang membutuhkan solusi[3]. Harapan telah berubah terkait layanan dan "makanan cepat saji" orang-orang beralih ke tempat lain di mana mereka mendapatkan apa yang mereka inginkan, yaitu bentuk pemenuhan keinginan instan[3]. Dalam perkembangan teknologi saat ini, Saat ini sudah terdapat beberapa aplikasi penyedia layanan antrian virtual seperti Antrique, Qiwii, dan ExaQue. Meskipun aplikasi-aplikasi tersebut umum digunakan, namun masih memiliki kelemahan seperti antarmuka pengguna yang kurang memuaskan, sering mengalami *crash*, tidak adanya estimasi waktu antrian, serta belum berfokus pada sektor kuliner.

Berdasarkan masalah diatas perlu di kembangkan aplikasi yang dapat mengatasi kekurangan aplikasi tersebut. Aplikasi pelanggan Antria bertujuan untuk memfasilitasi dan menggantikan proses mengantri yang membosankan, tidak produktif, dan menghabiskan banyak waktu saat memesan makanan yang mereka inginkan. Ini akan membuat pengantri tetap dapat melakukan aktivitas atau kegiatan produktif lainnya sambil menunggu di lokasi tujuan dari mana pun. Aplikasi pelanggan antria akan dikembangkan menggunakan flutter, flutter adalah kerangka kerja sumber terbuka dari Google untuk membangun aplikasi yang indah, di kompilasi secara *native*, *multi-platform*, dari satu kode sumber[4].

Dalam melakukan pengembangan aplikasi pelanggan antria, aplikasi akan selalu melakukan pembaruan dan pengembangan sesuai dengan kebutuhan pengguna, yang dimana akan selalu dilakukan pemeliharaan dan juga pengujian. Untuk dapat memenuhi hal tersebut dibutuhkan arsitektur yang tepat dalam pengembangan aplikasi pelanggan antria. Oleh karena itu aplikasi pelanggan antria akan menerapkan *clean architecture* dalam pengembangannya, *clean architecture* menyediakan metode untuk membangun struktur arsitektur aplikasi dan menyelesaikan masalah manajemen state serta memisahkan logika bisnis dari implementasi platform spesifik [5]. Kelebihan dari *clean architecture* yaitu kemampuan untuk menerapkan pemisahan kepentingan terkait fungsi platform tertentu dan fungsionalitas yang independen. Meminimalkan masalah saat melakukan peningkatan tanpa mengubah elemen logika bisnis, alur data, atau interaksi. Selain itu, *clean architecture* memungkinkan pengujian aturan bisnis tanpa ketergantungan eksternal, yang memungkinkan perubahan antarmuka pengguna (UI) dan sumber data tanpa masalah. [6]. Dengan menggunakan *clean architecture* diharapkan proses pengembangan dan pengujian dapat lebih cepat dan terstruktur.

Untuk memastikan aplikasi berfungsi dengan tepat, maka perlu dilakukan pengujian untuk memastikan aplikasi berjalan dengan tepat. Dalam penelitian ini, *clean architecture* akan berfokus pada aspek *maintainability* dari fitur-fitur yang terdapat pada aplikasi[9]. Pengujian *maintainability* pada penelitian ini akan menggunakan SonarQube, yang dapat memberikan laporan terperinci terkait *maintainability* [17]. Pengujian pada aplikasi juga dilakukan dengan menggunakan *User Acceptance Testing* (UAT). *User Acceptance Testing* adalah proses untuk memeriksa apakah sistem menerima persyaratan pengguna atau tidak. Ini dilakukan pada saat sistem digunakan oleh pengguna sesungguhnya[14]. Hasil yang diperoleh dari *User Acceptance Testing* (UAT) ini sangat penting untuk melakukan perbaikan dan penyesuaian akhir sebelum aplikasi secara resmi digunakan oleh pengguna.

1.2 Topik dan Batasannya

Selama tahap pengembangannya, *clean architecture* akan digunakan untuk mengembangkan aplikasi pelanggan antrian. Maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana implementasi aplikasi pelanggan antrian dengan menggunakan *clean architecture*?
2. Bagaimana tingkat *maintainability* aplikasi mobile pelanggan antrian yang telah menerapkan *clean architecture*?
3. Apakah implementasi antarmuka aplikasi pelanggan antrian sudah sesuai dengan desain yang telah dibuat?

Batasan masalah dari perancangan aplikasi pelanggan antrian adalah sebagai berikut:

1. Implementasi antarmuka dan integrasi API pada aplikasi pelanggan antrian dengan menerapkan *clean architecture*.
2. Penggunaan flutter digunakan pada aplikasi berbasis android.

1.3 Tujuan

Tujuan yang ingin dicapai pada implementasi aplikasi pelanggan antrian adalah sebagai berikut :

1. Mengimplementasikan *clean architecture* pada pengembangan aplikasi pelanggan antrian.
2. Menilai tingkat *maintainability* dari aplikasi pelanggan antrian yang mengimplementasikan *clean architecture*.
3. Membangun antarmuka aplikasi pelanggan antrian sesuai dengan desain yang telah dibuat.