

1. Pendahuluan

Bagian pendahuluan ini memberikan gambaran umum tentang pentingnya penelitian, latar belakang masalah dan batasan penelitian, ruang lingkup topik, dan tujuan penelitian.

Latar Belakang

Pada Kehawatiran yang semakin meningkat bagi para pemilik merek dan konsumen muncul akibat maraknya barang palsu, menyebabkan dampak negatif pada pengalaman pengguna dan citra merek. Dampak ini dapat berujung pada ketidakpuasan konsumen [1]. Pemerintah Indonesia telah melakukan berbagai upaya pencegahan terhadap pemalsuan barang dengan melindungi merek dagang. Oleh karena itu, pemalsuan barang termasuk dalam pelanggaran Hak Kekayaan Intelektual (HAKI) pada merek dagang [2].

Perlindungan dan penegakan merek diatur dalam Undang-Undang Nomor 15 Tahun 2001. Terdapat beberapa platform atau aplikasi dapat digunakan untuk memeriksa keaslian suatu barang, seperti Depatu, Legit App, dan Barcodelookup. Namun, setiap aplikasi atau platform tersebut memiliki kelebihan dan kelemahan yang berbeda-beda. Depatu dan Legit App, sebagai contoh, memungkinkan pengecekan keaslian barang dengan cara pengguna mengunggah foto produk, yang kemudian diperiksa oleh tim ahli dari platform tersebut. Kekurangan dari kedua platform tersebut adalah ketidakmampuannya melakukan pengecekan secara otomatis. Di sisi lain, Barcodelookup memfasilitasi pemeriksaan otomatis melalui QR code, kelemahannya adalah ketidakmampuannya dalam mendeteksi keaslian suatu barang. Berdasarkan kekurangan-kekurangan tersebut, diciptakanlah LegitCheck yang dirancang khusus untuk mengatasi kelemahan yang ada pada beberapa aplikasi tersebut.

LegitCheck merupakan aplikasi yang dirancang dalam melakukan verifikasi keaslian suatu produk. Pada aplikasi LegitCheck mencakup fitur-fitur yang lebih luas seperti kemampuan pembuatan QR code oleh pemilik merek, pengelolaan QR code, verifikasi otomatis keaslian barang melalui QR code, pemilihan barang setelah pemilik klaim QR code tersebut, serta perpindahan kepemilikan QR code dari suatu barang. Fitur ini memungkinkan pemilik untuk memiliki kendali atas QR Code dan akses langsung ke informasi terkait produk tersebut. Fokus utama aplikasi LegitCheck adalah melakukan perpindahan kepemilikan QR code dari pemilik pertama ke pemilik berikutnya. Oleh karena itu, dengan adanya aplikasi LegitCheck dapat membantu mereka yang tidak ingin risiko dalam memiliki barang palsu.

Saat ini sistem perangkat lunak semakin rumit dan canggih untuk memenuhi kebutuhan bisnis yang baru. Arsitektur perangkat lunak (*software architecture*) cocok dalam mengatasi kompleksitas dan perubahan perangkat lunak yang semakin meningkat [3]. Dalam melakukan perancangan dan pembangunan aplikasi mobile LegitCheck dengan kinerja yang baik, pola arsitektur merupakan hal penting yang harus diperhatikan, karena aplikasi memiliki karakteristik dan pendekatan yang berbeda beda.

Dalam memilih pola arsitektur, penting untuk mempertimbangkan kerangka kerja atau platform yang digunakan. Terdapat beberapa arsitektur yang digunakan dalam pengembangan aplikasi mobile seperti MVC (*Model-View-Controller*), VIPER (*View-Interactor-Presenter-Entity-Router*), MVVM (*Model-View-ViewModel*), dan lainnya, untuk pengembangan aplikasi *mobile* [4]. MVVM kerap kali digunakan dalam melakukan pengembangan pada aplikasi berbasis *mobile* [5]. Perancangan dan pembangunan aplikasi akan menggunakan kerangka kerja (*framework*) dengan menerapkan konsep pola desain MVVM sebagai landasan untuk mengatur struktur dan aliran data dalam aplikasi. MVVM adalah salah satu pola arsitektur (*architecture pattern*) yang mengangkat pada pemisahan kepentingan, hal tersebut memungkinkan dalam pemisahan antara grafis antarmuka (*graphical user interface*) dengan proses logika bisnis (*business logic*) atau logika halaman belakang (*back-end logic*) [3].

Pola desain yang dipilih adalah MVVM karena metode MVVM lebih baik digunakan jika melakukan efisiensi pada metrik saat waktu eksekusi dan load time atau waktu yang dibutuhkan untuk membuka halaman API data, dikarenakan adanya data binding yang berfungsi dalam meringkas kode, sehingga mempersingkat waktu eksekusi pada program, dan dengan diterapkannya LiveData yang berfungsi mempercepat proses pengambilan atau pemanggilan data dari API [6]. Arsitektur ini terdiri dari tiga layer, yaitu *model*, *view*, dan *ViewModel* [7]. Dalam arsitektur MVVM, *ViewModel* berfungsi untuk berkomunikasi dengan *Model*, dan mengelola interaksi dengan API. *ViewModel* mengambil data dari *model* melalui API, kemudian melakukan pengolahan data, dan mempersiapkan data untuk tampil di *View*. Dengan menggunakan MVVM dan API, aplikasi LegitCheck dapat optimal dalam pengelolaan data, komunikasi dengan *Model*, dan menampilkan informasi yang diperlukan kepada pengguna secara efisien dan terstruktur.

Setelah fitur atau fungsi telah selesai dikembangkan, langkah selanjutnya adalah melakukan pengujian dengan Black box dan White box, pengujian Black box untuk LegitCheck bertujuan untuk melakukan pengujian fungsional dikarenakan aplikasi LegitCheck merupakan aplikasi baru yang mana untuk menghindari *bug* ketika digunakan oleh user, maka pengujian Black box menjadi alasan digunakan dalam pengujian aplikasi LegitCheck. Sedangkan pengujian White box merupakan pengujian yang dikembangkan berdasarkan kode program [8]. Pengujian White box pada LegitCheck adalah pengujian untuk menguji semua jalur eksekusi guna membantu

mendeteksi kesalahan logika serta memastikan bahwa setiap output dari fungsi kode berjalan sesuai dengan ekspektasi, agar tidak terjadi bug yang tidak diinginkan ketika aplikasi LegitCheck digunakan oleh pengguna.

Topik dan Batasannya

Topik ini berfokus pada pengembangan aplikasi LegitCheck menggunakan arsitektur MVVM pada aplikasi LegitCheck dan pengujian pada aplikasi LegitCheck. Pembahasan ini berupa metode dan alat yang digunakan dalam melakukan pengujian fungsional. Metode pengujian yang dilakukan dengan cara dua pengujian yaitu dengan pengujian White box dan Black box. Pada pengujian White box dan Black box terdapat dua cara yaitu dengan manual dan otomatis menggunakan library testing yang sudah terdapat dalam flutter test. Selain itu melakukan pengujian Black box untuk melakukan pengujian fungsionalitas dari aplikasi yang sudah dibangun dengan menggunakan arsitektur MVVM. Pengembangan aplikasi mobile LegitCheck juga melibatkan penggunaan infrastruktur API yang terhubung dengan penyediaan data, memastikan aplikasi dapat memberikan informasi yang akurat dan *real-time* kepada pengguna, namun pada penelitian ini hanya terbatas dalam pengujian White box dan Black box, tidak melakukan pengujian load test pada API.

Tujuan

Berikut adalah tujuan yang akan dicapai:

1. Mengimplementasikan arsitektur MVVM dalam melakukan pengembangan aplikasi LegitCheck
2. Menerapkan pengujian White box secara manual dan otomatis menggunakan *library* Flutter test
3. Melakukan pengujian Black box secara manual dan otomatis menggunakan *library* Flutter `integration_test` untuk melakukan uji fungsionalitas aplikasi berdasarkan pandangan pengguna
4. Melakukan analisis hasil pengujian untuk mengidentifikasi *bug* atau kekurangan dalam implementasi dan untuk memastikan aplikasi dapat berjalan di multi platform yaitu Android dan iOS dan aplikasi dapat berjalan sesuai dengan harapan