

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Bukalapak adalah perusahaan teknologi Indonesia yang memiliki misi menciptakan perekonomian yang adil untuk semua dengan melalui platform online dan offlinenya. Salah satu produk utama dari perusahaan ini adalah *E-Commerce* dimana terdapat pengguna yang dapat menjadi penjual dan pembeli. Kedua jenis pengguna ini dapat melakukan transaksi satu sama lain dari produk yang telah disimpan oleh penjual.

Setiap penjual yang ada akan mendaftarkan dan mengiklankan produk mereka pada *E-Commerce* Bukalapak dengan spesifikasi sedetail dan sejelas mungkin, sehingga pembeli dapat mencari barang dengan akurat. Begitu pula dengan peran Bukalapak sebagai penengah, setiap data yang telah penjual masukan pada masing-masing produk mereka akan diolah oleh aplikasi Bukalapak menjadi suatu informasi yang dapat diterima dengan baik oleh pembeli.

Ada banyak cara untuk mengolah data produk yang didaftarkan oleh penjual pada aplikasi, salah satunya adalah menampilkan *tag* produk. *Tag* merupakan suatu informasi pengelompokan suatu produk pada kriteria tertentu, sebagai contoh terdapat beberapa penjual yang mendaftarkan produknya agar dapat melakukan pembayaran dengan cara *Cash on Delivery (COD)*, agar penjual mengetahui bahwa produk tersebut dapat dibeli dengan cara pembayaran *COD*, maka diberikanlah *tag* dengan nama *COD* pada produk tersebut, sehingga berfungsi untuk mempermudah pembeli untuk mengambil informasi dari produk tersebut.

Namun saat penulis melakukan pengecekan kode dan wawancara terhadap salah satu pegawai yang bekerja pada *Squad* Pencarian / *Search*, diketahui bahwa sistem informasi dalam fitur *tag* yang ada pada fitur pencarian saat ini terdapat perbedaan informasi antara *platform* Android dan iOS. Alasan utama yang mendasari hal tersebut yakni karena sistem informasi yang dioperasikan cukup sulit untuk

dipelihara, selain itu pengembang juga perlu waktu untuk memahami *source code* yang terdapat pada fitur tersebut, fakta ini juga diperkuat setelah penulis melakukan pengecekan pada *source code* sistem informasi dari fitur tersebut pada *platform* Android, ditemukan sangat banyak penamaan variabel dan fungsi yang tidak informatif atau bahkan disingkat. Selain itu terdapat banyak *boilerplate* pada pengimplementasian fitur tersebut dan bahkan *dead code* yang sengaja tidak dihapus, hal ini tentu mempengaruhi *readability* dan *maintainability* fitur tersebut, belum lagi kasus ini terjadi pada *platform* satunya yaitu iOS, hal ini terjadi karena pada saat implementasi pembuatan fitur, *source code* diimplementasikan oleh orang yang berbeda. Selain itu, kebutuhan *Product Manager* akan fitur *tag* saat itu sangat dinamis, hal ini berdampak pada jenis-jenis *tag* yang muncul pada suatu produk. Karena terdapat limitasi secara teknik pada fitur *tag*, yaitu setiap ada informasi tag baru yang ingin ditampilkan, pengembang perlu mengubah kode pada fitur tersebut dan diperlukannya *deployment* ulang.

Berdasarkan pokok permasalahan yang ditemukan, penulis perlu memberikan solusi agar sistem informasi yang beroperasi pada fitur *tag* pada saat ini dapat menjadi dinamis dalam menampilkan *tag*, selain itu mengimplementasikan *Clean Architecture* pada sistem informasi agar lebih *extendable* dan *testable*[1]. Selain dari pada itu penulis memberikan solusi agar mencoba mengaplikasikan *source code* yang sudah diimplementasikan dengan *Clean Architecture* pada *Kotlin Multiplatform Mobile* sehingga dapat mempercepat proses *development*[2], mengurangi potensi *human error*, dan lebih mempermudah *maintainability* karena hanya ada satu *source code* yang diaplikasikan.

Pada kesempatan ini penulis akan mengangkat skripsi yang berjudul Metode Penerapan *Clean Architecture* dengan *Kotlin Multiplatform Mobile* pada Aplikasi Bukalapak.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas, masalah yang ditemukan yaitu:

1. Limitasi secara teknik yang menyebabkan informasi *tag* pada produk tidak dinamis sehingga selalu diperlukannya perubahan pada kode dan adanya *deployment* ulang.
2. Sulitnya *maintainability* dan *readability* dari *source code* yang ada pada sistem informasi di fitur *tag*.
3. *human error* yang terjadi karena proses pembuatan *source code* diimplementasikan oleh orang yang berbeda, sehingga terdapat celah untuk membuat *output* dari sistem informasi menjadi berbeda.

1.3 Maksud dan Tujuan

1.3.1 Maksud

Maksud dari penelitian ini adalah merubah suatu implementasi lama dari fitur *tag* di *Search* pada aplikasi Bukalapak dengan cara melakukan *reengineering* dengan cara menerapkan metode *Clean Architecture* menggunakan *Kotlin Multiplatform* (KMM) yang dapat diimplementasi pada platform Android dan iOS untuk menyelesaikan masalah yang dialami oleh *user* (*Mobile Engineer*) dan *Product Manager*.

1.3.2 Tujuan

Tujuan implementasi *design pattern* dengan metode *Clean Architecture* menggunakan KMM sebagai berikut :

1. Mengubah suatu proses bisnis pada sistem informasi fitur *tag* sehingga informasi *tag* yang dimunculkan pada produk dapat dinamis dan *configurable* berdasarkan *config*.
2. Mengimplementasikan *clean architecture* agar kode terbagi menjadi beberapa *layer*, sehingga kode menjadi tidak menumpuk pada salah satu *layer* implementasi, sehingga kode menjadi lebih *maintanable* dan *readble*.

3. Mengaplikasikan kode yang sudah diimplementasikan dengan *clean architecture* kedalam *Kotlin Multiplatform Mobile*, sehingga satu *source code* dapat digunakan oleh dua *platform mobile* sekaligus.

1.4 Batasan Masalah

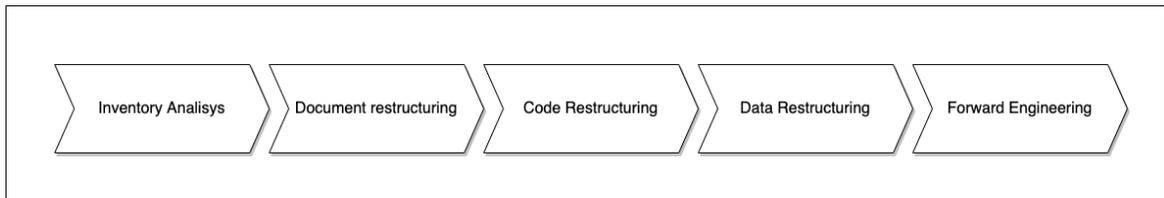
Adapun batasan masalah yang digunakan pada penelitian ini adalah sebagai berikut.

1. Proses *reengineering* dengan mengimplementasikan *clean architecture* dengan KMM yang bertujuan hanya untuk aplikasi *mobile*.
2. Platform yang didukung hanya sistem operasi Android dan iOS.
3. Kode akan diimplementasikan pada fitur *Search* di aplikasi Bukalapak.
4. Fokus pada penelitian ini yaitu pada *source code* fitur *Tag Search* saja, tanpa mengubah *behavior* utama terhadap pengguna (pembeli).

1.5 Metodologi Penelitian

Jenis penelitian ini berlaku untuk penelitian kualitatif. Penelitian terapan bertujuan untuk menemukan solusi atas permasalahan yang dihadapi masyarakat dan dunia usaha. Metodologi yang digunakan dalam penelitian ini memiliki beberapa tahapan. Dalam membantu proses *reengineering* pada aplikasi Bukalapak, peneliti mengadopsi suatu proses model *reengineering* pada buku *Software Engineering A Practitioners Approach* yang ditulis oleh Roger S Pressman yang diadaptasikan dengan kasus yang dihadapi peneliti. Pada proses model *Reengineering* terbagi menjadi 6 aktivitas, yaitu *Inventory Analysis*, *Document Restructuring*, *Reverse Engineering*, *Code Restructuring*, *Data Restructuring* dan *Forward Engineering*[3]. Akan tetapi peneliti hanya mengadopsi 5 proses dari proses model yang ada pada buku tersebut, peneliti tidak memerlukan proses *Reverse Engineering* karena peneliti sudah memiliki kode dari implementasi fitur lama, sehingga proses tersebut dapat dilewati. Selain itu proses yang dilakukan tidak repetitif karena ruang lingkup

penelitian ini hanya akan melakukan proses *reengineering* sebanyak satu kali. Sehingga didapatkannya proses *reengineering* seperti pada Gambar 1-1.



Gambar 1-1 Proses *reengineering* pada aplikasi Bukalapak

1. *Inventory Analysis*

Peneliti melakukan analisis bisnis dan juga kondisi teknis dari fitur yang ada, sehingga hasil *reengineering* dapat menghasilkan rancangan yang baik dan dapat mengatasi masalah yang ada.

2. *Document Restructuring*

Pada proses ini peneliti mendokumentasikan hasil dari analisis dan juga konsep rancangan dari proses sebelumnya. Dokumentasi dipakai sebagai acuan dalam proses *Code Restructuring* dan *Data Restructuring*, selain itu juga dokumentasi dapat dipakai sebagai petunjuk dari suatu fitur yang ada.

3. *Data Restructuring*

Perubahan struktur data dari fitur yang ada dilakukan peneliti agar fitur dapat memenuhi kebutuhan bisnis dan juga sebagai prasyarat untuk melakukan *Code Restructuring*.

4. *Code Restructuring*

Pada proses ini peneliti melakukan implementasi dari rancangan yang sudah dibuat, penulis melakukan proses *reengineering* dengan mengimplementasikan rancangan dengan konsep *Clean Architecture*.

5. *Forward Engineering*

Setelah melakukan proses *Code Restructuring*, hasil kode akan diadopsi dan dikompilasi menggunakan *Kotlin Multiplatform Mobile* sehingga kode dapat dipakai di kedua platform *mobile* dan juga memakai *framework* terbaru yang lebih futuristik.

1.6 Sistematika Penulisan

Sistematika penulisan disusun untuk memberikan gambaran umum mengenai penelitian yang dikerjakan. Adapun sistematika penulisan yang digunakan dalam penyusunan laporan skripsi ini yaitu sebagai berikut.

BAB 1 PENDAHULUAN

Bab 1 menguraikan latar belakang permasalahan, merumuskan inti permasalahan, mencari solusi atas masalah tersebut, menentukan maksud dan tujuan, pembatasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Bab 2 menguraikan sejarah perusahaan, visi & misi, dan struktur organisasi perusahaan dan bahan-bahan kajian, konsep dasar, dan teori dari para ahli yang berkaitan dengan penelitian. Meninjau permasalahan dan hal-hal yang berguna dari penelitian-penelitian serupa yang pernah dikerjakan sebelumnya dan menggunakannya sebagai acuan penyelesaian masalah pada penelitian ini.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab 3 menguraikan hasil analisis pengguna, memodelkan pengguna, menguraikan proses pembuatan desain dengan pendekatan *participatory* beserta hasil yang didapatkan berupa desain konseptual dan model desain dalam bentuk prototipe digital.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab 4 menguraikan implementasi berdasarkan hasil analisis sebelumnya. Pada bagian ini juga akan ditentukan bagaimana model desain interaksi dibangun, diuji dan disesuaikan dengan hasil penelitian.

BAB 5 KESIMPULAN DAN SARAN

Bab 5 menguraikan tentang kesimpulan dari hasil penelitian beserta saran untuk pengembangan selanjutnya.