

***REFACTORING DENGAN PENDEKATAN PRINSIP CLEAN
ARCHITECTURE PADA APLIKASI CHAKRA REWARDS BERBASIS
ANDROID***

SKRIPSI

Untuk Memenuhi Sebagian Persyaratan Mencapai Derajat Sarjana S-1

Program Studi Teknik Informatika



Disusun oleh:

A. MUH. FAKHRY MUBARAK

19106050028

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2023



PENGESAHAN TUGAS AKHIR

Nomor : B-154/Un.02/DST/PP.00.9/01/2023

Tugas Akhir dengan judul : REFACTORING DENGAN PENDEKATAN PRINSIP CLEAN ARCHITECTURE
PADA APLIKASI CHAKRA REWARDS BERBASIS ANDROID

yang dipersiapkan dan disusun oleh:

Nama : A. MUH. FAKHRY MUBARAK
Nomor Induk Mahasiswa : 19106050028
Telah diujikan pada : Senin, 09 Januari 2023
Nilai ujian Tugas Akhir : A

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR



Ketua Sidang
Dr. Agung Fatwanto, S.Si., M.Kom.
SIGNED

Valid ID: 63c6641d95d3



Penguji I
Ir. Sumarsono, S.T., M.Kom.
SIGNED

Valid ID: 63c6578c79393



Penguji II
Muhammad Didik Rohmad Wahyudi, S.T.,
MT.
SIGNED

Valid ID: 63c600ba90115



Yogyakarta, 09 Januari 2023
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi
Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

Valid ID: 63c7421a00122

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : A. Muh. Fakhry Mubarak

NIM : 19106050028

Program Studi : Teknik Informatika

Fakultas : Sains dan Teknologi

menyatakan bahwa skripsi saya yang berjudul “*Refactoring dengan Pendekatan Prinsip Clean Architechure pada Aplikasi Chakra Rewards Berbasis Android*” merupakan hasil penelitian saya sendiri, tidak terdapat pada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi, dan bukan plagiasi karya orang lain kecuali yang secara tertulis diacu dalam nasikah ini dan disebutkan dalam daftar pustaka.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 28 Desember 2022

Yang menyatakan,



A. Muh. Fakhry Mubarak

NIM. 19106050028

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamualaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : A. Muh. Fakhry Mubarak

NIM : 19106050028

Judul Skripsi : *Refactoring* dengan Pendekatan Prinsip *Clean Architecture* pada Aplikasi Chakra Rewards Berbasis Android

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Teknik Informatika.

Dengan ini kami berharap agar skripsi/tugas akhir Saudara dapat segera di-*munaqasyah*-kan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamualaikum Wr. Wb.

Yogyakarta, 28 Desember 2022

Pembimbing,



Dr. Agung Fatwanto, S. Si., M. Kom.

NIP. 19770103 200501 1 003

MOTTO

“Make big plans. Enjoy little things.”

“Whatever you do, do it 100%. When you work, work.

When you laugh, laugh. When you eat, eat like it's your last meal,”

– Tony Lip, dalam Film Green Book (2018)



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

HALAMAN PERSEMBAHAN

Saya persembahkan tugas akhir ini kepada kedua orang tua saya yang tak henti-hentinya mendoakan saya, serta selalu mencari jalan untuk mendidik dan merawat saya hingga detik ini.



KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah SWT yang telah memberikan kemampuan, kekuatan serta pengetahuan kepada penulis sehingga penulis dapat menyelesaikan penelitian dan penyusunan skripsi dengan judul “*Refactoring* dengan Pendekatan Prinsip *Clean Architecture* pada Aplikasi Chakra Rewards Berbasis Android” dengan baik. Sholawat dan salam selalu tercurahkan kepada junjungan kita Rasulullah SAW, yang telah menuntun kita menuju ke zaman yang terang benderang, dan semoga kita semua mendapatkan syafaatnya di hari kiamat kelak.

Tidak dapat disangkal bahwa butuh usaha yang keras dalam penyelesaian skripsi ini, akan tetapi dalam proses penyelesaiannya tidak lepas dari bantuan dan dukungan berbagai pihak, baik secara langsung maupun tidak langsung. Secara spesifik, penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua, bapak, Drs. Anwar, M.Pd. dan mama, Dra. Raja Sangngang yang telah membesarkan dan merawat penulis hingga dapat mengenyam pendidikan tinggi dan menjadi pribadi yang lebih baik setiap waktunya, serta kakak, Andi Rafiqa Faradiyah yang selalu memberikan dukungan langsung maupun tidak langsung setiap waktunya,
2. Bapak Dr. Agung Fatwanto, S. Si., M. Kom., selaku Dosen Pembimbing Skripsi penulis yang senantiasa sabar dalam memberikan masukan dan arahan selama penulisan dan penyusunan skripsi,

3. Bapak Ir. Aulia Faqih Rifa'i, M.Kom., selaku Dosen Penasihat Akademik penulis selama kuliah di UIN Sunan Kalijaga Yogyakarta,
4. Ibu Ir. Maria Ulfah Siregar, S. Kom., MIT., Ph. D., selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta, serta seluruh dosen dan karyawan program studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta yang telah memberi ilmu dan bantuan selama masa studi kuliah penulis,
5. Mas Dimas Prasetyo Tegar Asmoro, S.Kom, M.Kom. yang senantiasa menjadi mentor penulis baik dalam dunia kerja maupun dalam kehidupan, serta rekan-rekan Departemen IT di PT Cakra Radha Mustika yang terus memberi dukungan selama proses penulisan skripsi,
6. Andi Shafa Salsabila Sukri yang selalu memberikan dukungan dan semangat kepada peneliti pada momen-momen yang mudah maupun yang sulit,
7. Teman-teman The Konts, yaitu Irfan, Shamil, Cholish, serta para tamu istimewa The Konts, yaitu Galih, Nawwab dan Uqi yang senantiasa berbagi cerita, pengalaman, rezeki serta dukungan terhadap satu sama lain di daerah perantauan,
8. Teman-teman SMAN 17 Makassar, seperti Dio, Ceri, Eva, Nuning, Bhagas, serta teman-teman Harlot IPA 4 yang tetap mendukung dan menemani masa perkuliahan,
9. Teman-teman seperjuangan di Teknik Informatika lintas angkatan yang telah menjadi rekan selama masa perkuliahan, terutama Ramadhan, Laras, Syauqi,

dan seluruh teman-teman angkatan 2019 yang telah menjadi teman diskusi dan selalu memberikan dukungan selama masa perkuliahan.

10. Serta semua pihak yang tidak bisa disebutkan satu-persatu.

Penulis menyadari masih banyak kekurangan dalam skripsi ini, sehingga kritik dan saran yang membangun dari pembaca sangat terbuka bagi penulis. Semoga tugas akhir ini dapat memberikan manfaat untuk mendorong penelitian-penelitian selanjutnya.

Yogyakarta, Desember 2022

Penulis

A. Muh. Fakhry Mubarak

19106050028

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR ISI

PENGESAHAN TUGAS AKHIR.....	i
PERNYATAAN KEASLIAN SKRIPSI.....	ii
SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR.....	iii
MOTTO	iv
HALAMAN PERSEMBAHAN.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiv
INTISARI.....	xv
ABSTRACT.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	4
1.4 Batasan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	6
2.1 Tinjauan Pustaka.....	6
2.2 Landasan Teori.....	13
2.2.1. PT Chakra Radha Mustika.....	13

2.2.2.	Chakra Rewards	13
2.2.3.	Refactoring.....	14
2.2.4.	The Clean architecture	17
2.2.5.	Android Architecture Component (AAC)	21
2.2.6.	MVVM (Model-View-View Model).....	24
2.2.7.	McCall's <i>Software Quality Metrics</i>	26
2.2.8.	API (<i>Application Programming Interface</i>).....	26
BAB III	METODE PENELITIAN	28
3.1	Alat dan Bahan Penelitian.....	28
3.2	Objek Penelitian.....	28
3.3	Langkah-Langkah Penelitian	28
3.3.1.	Observasi dan Evaluasi Struktur Kode	28
3.3.2.	<i>Refactoring</i> Aplikasi Chakra Rewards	30
3.3.3.	Pengujian Aplikasi Sebelum dan Setelah <i>refactoring</i>	31
3.3.4.	Analisis Hasil dan Pembahasan	32
BAB IV	HASIL DAN PEMBAHASAN	33
4.1	Observasi dan Evaluasi Struktur Kode Aplikasi Chakra Rewards	33
4.1.1.	<i>Software Architecture</i>	33
4.1.2.	<i>Software Design Pattern</i>	35
4.1.3.	<i>Package Structure</i>	36
4.1.4.	Components	40
4.2	<i>Refactoring</i> Aplikasi Chakra Rewards	45
4.2.1.	Implementasi Clean Architecture	45

4.2.2.	Implementasi <i>Software Design pattern</i> MVVM.....	47
4.2.3.	<i>Refactoring package Structure</i>	49
4.2.4.	<i>Refactoring Components</i>	53
4.3	Pengujian Aplikasi Sebelum dan Setelah <i>Refactoring</i>	61
4.3.1.	Conciseness.....	62
4.3.2.	Consistency	62
4.3.3.	Generality.....	63
4.3.4.	<i>Instrumentation</i>	68
4.3.5.	Self-Documentation	71
4.3.6.	Simplicity	71
4.3.7.	Auditability	73
4.3.8.	Expandability	74
4.3.9.	Modularity	76
4.4	Analisis Hasil Pengujian Sebelum dan Setelah <i>Refactoring</i>	77
BAB V	PENUTUP.....	82
5.1	Kesimpulan	82
5.2	Saran	83
DAFTAR PUSTAKA	84
LAMPIRAN	87
CURRICULUM VITAE	97

DAFTAR GAMBAR

Gambar 2.1 Diagram Sistem <i>Membership</i> Chakra Rewards	14
Gambar 2.2 Diagram Perbandingan Perancangan Perangkat Lunak	15
Gambar 2.3 Diagram Clean Architecture.....	20
Gambar 2.4 Layer-Layer Pengembangan Aplikasi Android.....	22
Gambar 2.5 Diagram UI Layer pada Android	23
Gambar 2.6 Diagram Data Layer pada Android	24
Gambar 2.7 Diagram MVVM pada Android	25
Gambar 4.1 Diagram Relasi antar Komponen Sebelum <i>Refactoring</i>	33
Gambar 4.2 Diagram <i>Architecture</i> Aplikasi Chakra Rewards v1.3.16.....	34
Gambar 4.3 File <i>Presenter</i> pada Chakra Rewards v1.3.16	35
Gambar 4.4 Diagram <i>Design pattern</i> MVP	36
Gambar 4.5 Struktur <i>Package</i> Utama Sebelum <i>Refactoring</i>	37
Gambar 4.6 <i>Class</i> <i>BrandPresenter</i>	42
Gambar 4.7 File <i>BrandPresenter.java</i>	43
Gambar 4.8 File <i>PhotoPreviewPresenter.java</i>	43
Gambar 4.9 <i>Class</i> <i>ListRedeemPresenter</i>	45
Gambar 4.10 Diagram Relasi antar Komponen Setelah <i>Refactoring</i>	46
Gambar 4.11 Implementasi <i>Clean Architecture</i> di Aplikasi Android.....	47
Gambar 4.12 View dan ViewModel pada Fitur Login	48
Gambar 4.13 Modularisasi Aplikasi	49
Gambar 4.14 <i>Class</i> <i>HomeChakraViewModel</i>	54
Gambar 4.15 <i>Class</i> <i>GetCardUseCase</i>	54

Gambar 4.16 <i>Class</i> UserCardRepository	55
Gambar 4.17 <i>Class</i> PenggunaCardLocalDataSource	56
Gambar 4.18 <i>Class</i> RemoteDataSource()	56
Gambar 4.19 <i>Class</i> TopUpViewModel.....	57
Gambar 4.20 <i>Class</i> MemberRepository	58
Gambar 4.21 <i>Class</i> MemberRemoteDataSource	59
Gambar 4.22 <i>Data class</i> CatalogueFilter(...)	59
Gambar 4.23 <i>Class</i> RedeemCatalogViewModel	60
Gambar 4.24 <i>Class</i> RedeemRepository.....	61
Gambar 4.25 Firebase Console Aplikasi Sebelum <i>refactoring</i>	69
Gambar 4.26 Firebase Console Aplikasi Setelah <i>refactoring</i>	70
Gambar 4.27 <i>Cyclomatic Complexity</i> Sebelum <i>Refactoring</i>	72
Gambar 4.28 <i>Cyclomatic Complexity</i> Setelah <i>Refactoring</i>	72
Gambar 4.29 Pengecekan Standar Kode Konvensi dengan SonarQube.....	73
Gambar 4.30 Penambahan Fitur pada Aplikasi Sebelum <i>Refactoring</i>	75
Gambar 4.31 Penambahan Fitur pada Aplikasi Setelah <i>Refactoring</i>	75
Gambar 4.32 <i>Module</i> pada Aplikasi Sebelum <i>Refactoring</i>	76
Gambar 4.33 <i>Module</i> pada Aplikasi Setelah <i>Refactoring</i>	77
Gambar 4.34 <i>Overview</i> Pengujian <i>Software Quality</i> dengan SonarQube.....	81

DAFTAR TABEL

Tabel 2.1 Ringkasan Tinjauan Pustaka	9
Tabel 4. 1 Perbandingan Jumlah File dan Baris Kode	62
Tabel 4.2 Komponen pada Aplikasi Sebelum <i>Refactoring</i>	64
Tabel 4.3 Komponen pada Aplikasi Setelah <i>Refactoring</i>	65
Tabel 4.4 Perbandingan Total Komponen yang <i>General</i>	68
Tabel 4.5 Ringkasan Hasil Pengujian McCall <i>Software Quality Metrics</i>	78

INTISARI

PT Cakra Radha Mustika merupakan perusahaan yang sedang mengembangkan aplikasi *membership* bernama Chakra Rewards, akan tetapi pengembangan aplikasi ini hanya berfokus pada fitur-fitur dari segi bisnis tanpa memperhatikan kualitas kode dan performa aplikasi sehingga kode yang ditulis sulit untuk dibaca, dipertahankan, dan diperluas. Oleh karena itu, perlu dilakukan *refactoring* dengan mengimplementasikan prinsip *clean architecture* untuk meningkatkan kualitas kode dan mempermudah pengembangan dan pemeliharaan aplikasi ini. Penelitian ini dilakukan untuk mengetahui bagaimana proses *refactoring* dan implementasi prinsip *clean architecture*, dan apakah aplikasi yang telah melalui proses tersebut lebih baik dari sisi kualitas kode yang diuji menggunakan McCall *Software Quality Metrics* dan platform SonarQube. Proses *refactoring* dimulai dengan observasi *code smell* dan *design pattern* serta struktur *package* yang kemudian di-*refactor* dengan menggunakan teknik-teknik *refactoring* dan memperhatikan prinsip-prinsip utama *clean architecture*. Hasil penelitian ini menunjukkan bahwa dengan melakukan *refactoring* pada aplikasi Chakra Rewards dan mengimplementasikan prinsip *clean architecture*, dapat meningkatkan kualitas kode pada *maintainability*, *flexibility*, dan *testability* serta mengurangi jumlah *bugs*, *vulnerabilities*, *code smells*, dan *code duplications*.

Kata Kunci: *Refactoring*, *Clean Architecture*, Chakra Rewards, McCall *Software Quality Metrics*, SonarQube

ABSTRACT

PT Cakra Radha Mustika is a company that is developing a membership application called Chakra Rewards, but the development of this application only focuses on features from a business perspective without paying attention to code quality and application performance so the code written is difficult to read, maintain, and expand. Therefore, it is necessary to do refactoring by implementing clean architecture principles to improve code quality and facilitate the development and maintenance of this application. This research was conducted to find out how the refactoring process and the implementation of clean architecture principles, and whether the application that has gone through the process is better in terms of code quality tested using McCall Software Quality Metrics and the SonarQube platform. The refactoring process starts with observing the code smell and design patterns as well as the package structure which is then refactored by using refactoring techniques and paying attention to the main principles of clean architecture. The results of this study show that by refactoring the Chakra Rewards application and implementing clean architecture principles, it can improve code quality in maintainability, flexibility, and testability and reduce the number of bugs, vulnerabilities, code smells, and code duplications.

Keywords: *Refactoring, Clean Architecture, Chakra Rewards, McCall Software Quality Metrics, SonarQube*

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengembangan rekayasa perangkat lunak beberapa tahun terakhir telah berkembang dengan sangat pesat. Berdasarkan data (Statista, 2019), jumlah pengembang perangkat lunak pada tahun 2019 adalah 24.5 juta dan diprediksi akan terus meningkat menjadi 28.7 juta pada tahun 2024. Dari seluruh pengembang perangkat lunak tersebut, sering kita temui pengembang yang bekerja dalam tim untuk *developing* (mengembangkan) dan *maintaining* (mempertahankan) sebuah perangkat lunak.

PT Cakra Radha Mustika (CRM) merupakan salah satu anak perusahaan dari PT Sanghiang Perkasa atau terkenal sebagai Kalbe Nutritionals, sebuah perusahaan yang menghasilkan produk-produk nutrisi berkualitas seperti suplemen vitamin, susu, biskuit, dan lain-lain. PT CRM merupakan perusahaan berbasis *Total Business Solution* yang menyediakan layanan dengan membantu perusahaan lain menjalankan proyek atau kampanye dari proses awal hingga langkah akhir. Salah satu sistem informasi yang sedang dikembangkan oleh PT CRM adalah sistem informasi *membership* atau *loyalty* dalam bentuk aplikasi bernama Chakra Rewards berbasis android yang dapat diunduh di Google Play Store. *Membership program* atau program loyalitas pelanggan merupakan program yang dirancang dengan tujuan meningkatkan dan mempertahankan jumlah transaksi yang dilakukan oleh pelanggan (Wirtz & Kum, 2001). Selama ini, pengembangan aplikasi Chakra

Rewards hanya berfokus pada fitur-fitur dari segi bisnis tanpa memperhatikan kualitas kode dan performa aplikasi.

Pada Desember 2021, peneliti bergabung dengan perusahaan ini dan menemukan bahwa aplikasi ini memiliki beberapa masalah antara lain *system design* yang buruk, kode yang sulit dibaca, dan aplikasi yang sulit untuk dipertahankan. Hal ini selaras dengan jumlah *code smell* sebanyak 2.2k, yang mengindikasikan bahwa terdapat sesuatu dalam kode yang tidak benar, dan perlu diselidiki lebih lanjut (Fowler, 2019). Selain *code smell*, dilakukan pula pengujian kompleksitas *software* menggunakan *cyclomatic complexity*, yaitu *metrics* perangkat lunak yang digunakan untuk mengukur kompleksitas program atau fungsi tertentu dalam suatu program (McCabe, 1976). Pada aplikasi ini, ditemukan 8.799 *cyclomatic complexity* yang menunjukkan bahwa aplikasi memiliki kompleksitas yang tinggi. Untuk mengatasi hal tersebut, perlu dilakukan peningkatan kualitas kode serta membuat aplikasi lebih mudah dibaca, dipertahankan, dan diperluas maka perlu dilakukan *refactoring* pada aplikasi tersebut dengan mengimplementasikan prinsip *clean architecture*.

Refactoring dapat didefinisikan sebagai perubahan struktur internal pada sebuah perangkat lunak dengan tujuan peningkatan performa, kode yang lebih mudah dimengerti, dan mempermudah modifikasi kode tanpa mengubah fungsionalitas dari sebuah perangkat lunak. *Refactoring* dilakukan dengan tujuan untuk meningkatkan desain perangkat lunak, mempermudah kode untuk dibaca, mempermudah menemukan *bug*, dan meningkatkan kecepatan pengembangan perangkat lunak (Fowler, 2019).

Sementara itu, *clean architecture* merupakan sebuah prinsip arsitektur sistem dalam pemrograman dengan tujuan *separation of concerns*, yaitu memisahkan perangkat lunak menjadi beberapa layer seperti *business rules* layer, *user interface* layer, dan layer lainnya yang diperuntukkan untuk pengguna dan sistem. Hasil yang diharapkan dengan mengimplementasikan *clean architecture* adalah meningkatkan kualitas desain *architecture* sehingga dapat meminimalkan sumber daya manusia yang diperlukan untuk membangun dan memelihara sistem yang diperlukan (Martin et al., 2018).

Tulisan ini akan berfokus pada *refactoring* aplikasi Chakra Rewards berbasis android yang dalam pengembangannya mengimplementasikan prinsip *clean architecture* dengan tujuan meningkatkan kualitas kode aplikasi sehingga aplikasi lebih mudah untuk dikembangkan dan dipertahankan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah yang akan diselesaikan dalam penelitian ini yaitu:

1. Bagaimana cara melakukan *refactoring* pada aplikasi Chakra Rewards ?
2. Bagaimana cara mengimplementasikan prinsip *clean architecture* pada aplikasi Chakra Rewards?
3. Apakah setelah *refactoring* dan mengimplementasikan prinsip *clean architecture* kualitas kode aplikasi Chakra Rewards meningkat?

1.3 Tujuan Penelitian

Berdasarkan latar belakang yang telah diuraikan, tujuan penelitian yang ingin dicapai yaitu untuk:

1. Mengetahui cara melakukan *refactoring* aplikasi Chakra Rewards.
2. Mengetahui cara mengimplementasikan prinsip *clean architecture* pada aplikasi Chakra Rewards.
3. Mengetahui apakah terjadi peningkatan kualitas kode pada aplikasi Chakra Rewards setelah dilakukan *refactoring* dan melakukan implementasi prinsip *clean architecture*.

1.4 Batasan Penelitian

Pada penelitian ini didefinisikan beberapa batasan sebagai berikut:

1. Versi aplikasi yang akan dijadikan sebagai objek penelitian adalah aplikasi v1.3.16 yang berjalan pada *environment development* (pengembangan).
2. Proses pengujian kualitas kode akan menggunakan McCall *Software Quality Metrics* dengan fokus *Product Revision*, serta menggunakan platform SonarQube.

1.5 Manfaat Penelitian

Penelitian ini akan mendukung baik peneliti, dan pembaca dalam mengetahui cara *refactoring* serta implementasi prinsip *clean architecture* dalam pengembangan sebuah aplikasi berbasis sistem operasi android. Bagi perusahaan tempat penelitian dilakukan, manfaat penelitian ini adalah mendapatkan aplikasi

dengan kualitas kode yang lebih baik. Diharapkan, apa saja yang tidak diterapkan pada penelitian ini dapat memberikan pelajaran bagi pembaca.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil *refactoring* dengan pendekatan *clean architecture* pada Chakra Rewards berbasis Android, dapat disimpulkan bahwa:

1. Untuk melakukan *refactoring* pada aplikasi Chakra Rewards perlu diawali dengan observasi *code smell* pada aplikasi, kemudian *code smell* tersebut di-*refactor* menggunakan teknik-teknik *refactoring* untuk meningkatkan kualitas kode.
2. Untuk mengimplementasikan prinsip *clean architecture* pada aplikasi Chakra Rewards, perlu diawali dengan observasi *design pattern* dan struktur *package* pada aplikasi. Setelah itu *design pattern* dan struktur *package* tersebut kemudian di-*refactor* dengan memperhatikan prinsip-prinsip utama *clean architecture*, yaitu *Single Responsibility*, *Open Closed*, dan *Dependency Inversion Principle*.
3. Hasil pengujian menggunakan McCall *Software Quality Metrics* menunjukkan bahwa dari 9 *metrics* pengujian, aplikasi setelah *refactoring* unggul pada 8 *metrics* pengujian dan tidak terjadi perubahan pada 1 *metrics* pengujian (*auditability*). Hal ini menunjukkan terdapat peningkatan kualitas kode dari sisi *maintainability*, *flexibility*, dan *testability*. Hasil pengujian menggunakan platform SonarQube juga menunjukkan terjadi peningkatan kualitas kode pada aplikasi dengan berkurangnya 29 *bugs*, 2 *vulnerabilities*, 1,996 *code smells*, dan 9.2% *code duplications*.

5.2 Saran

Pada proses *refactoring* aplikasi yang dilakukan, proses testing masih terbatas pada manual testing pada tahap *system testing* dengan *environment development* yang dilakukan oleh *Software quality Assurance* (QA) di PT Cakra Radha Mustika. Oleh karena itu, peneliti menyarankan untuk menuliskan *automation testing* berupa *unit testing* yang dilakukan oleh Android Engineer dengan harapan, *bug* akan terdeteksi lebih cepat jika dilakukan *refactoring*.



DAFTAR PUSTAKA

- Aflah Taqiu Sondha, Umi Sa'adah, Fadilah Fahrul Hardiansyah, & Maulidan Bagus Afridian Rasyid. (2020). Framework dan Code Generator Pengembangan Aplikasi Android dengan Menerapkan Prinsip Clean Architecture. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 9(4), 327–335. <https://doi.org/10.22146/jnteti.v9i4.572>
- Arribe, E., Aryanto, & Asrianto, R. (2021). APLIKASI E-MARKETPLACE MENGGUNAKAN ARSITEKTUR MVVM (MODEL-VIEW-VIEWMODEL) BERBASIS ANDROID. *JURNAL FASILKOM*, 11(2), 75–78. <https://doi.org/10.37859/jf.v11i2.2762>
- Christina Juliane, Rizal Dzulkarnaen, & Windi Susanti. (2019). Metode McCall's untuk Pengujian Kualitas Sistem Informasi Administrasi Tugas Akhir (SIATA). *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 3(3), 488–495. <https://doi.org/10.29207/resti.v3i3.1170>
- Daniel Jacobson, Greg Brail, & Dan Woods. (2012). *APIs: A Strategy Guide* (Mary Treseler, Ed.). O' Relly Media.
- Fowler, M. (2019). *Refactoring: Improving the Design of Existing Code* (K. Beck, Ed.; Second Edition). Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- I Made Sukarsa, I Nyoman Piarsa, & I Gede Bagus Premana Putra. (2020). Application of MVP Architecture in Developing Android-Based Seminar

- Ticket Booking Applications. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(3), 513–520. <https://doi.org/10.29207/resti.v4i3.1396>
- Khairullah, Soedijono, B., & al Fatta, H. (2017). Pengukuran Kualitas Sistem Informasi Inventaris Aset Universitas Muhammadiyah Bengkulu Menggunakan Metode Mccall. *Jurnal Informasi Interaktif*, 2(2), 84–92.
- Lou, T. (2016). *A comparison of Android Native App Architecture MVC, MVP and MVVM*. Aalto University.
- Martin, R. C., Grenning, J., Brown, S., Henney, K., & Gorman, J. (2018). *Clean architecture: a craftsman's guide to software structure and design* (Issue s 31). Prentice Hall.
- McCabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering*, SE-2(4), 308–320. <https://doi.org/10.1109/TSE.1976.233837>
- Prabowo, G., Suryotrisongko, H., & Tjahyanto, A. (2018). A Tale of Two Development Approach: Empirical Study on The Maintainability and Modularity of Android Mobile Application with Anti-Pattern and Model-View-Presenter Design Pattern. *2018 International Conference on Electrical Engineering and Informatics (ICELTICS)(44501)*, 149–154. <https://doi.org/10.1109/ICELTICS.2018.8548784>
- Statista. (2019). *Global Developer Population 2024*. <https://www.statista.com/statistics/627312/worldwide-developer-population>
- Wirtz, J., & Kum, D. (2001). Determinants of Membership Marketing Programs. *Der Markt*, 40(1), 40–49. <https://doi.org/10.1007/BF03033971>

Wisnuadhi, B., Munawar, G., & Wahyu, U. (2020). Performance Comparison of Native Android Application on MVP and MVVM. *Proceedings of the International Seminar of Science and Applied Technology (ISSAT 2020)*. <https://doi.org/10.2991/aer.k.201221.047>

