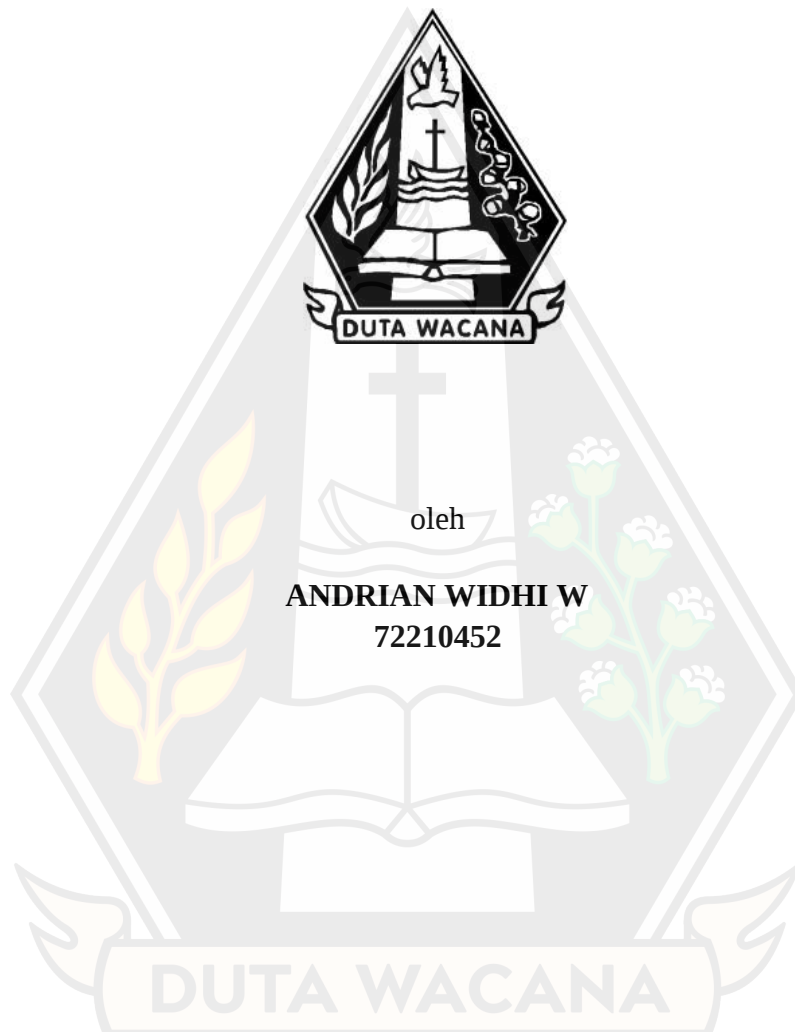


**APLIKASI EKSTRAKSI FITUR LONG METHOD PADA KODE PROGRAM
BERBASIS MOBILE DENGAN METRICS NOLV, CYCLO, DAN NCNDC**

Skripsi



oleh

**ANDRIAN WIDHI W
72210452**

**PROGRAM STUDI SISTEM INFORMASI FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
TAHUN 2024**

**APLIKASI EKSTRAKSI FITUR LONG METHOD PADA KODE PROGRAM
BERBASIS MOBILE DENGAN METRICS NOLV, CYCLO, DAN NCNDC**

Skripsi



Diajukan kepada Program Studi Sistem Informasi Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh
Gelar Sarjana Komputer

Disusun oleh

ANDRIAN WIDHI W
72210452

PROGRAM STUDI SISTEM INFORMASI FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
TAHUN 2024

PERNYATAAN PENYERAHAN KARYA ILMIAH

Saya yang bertanda tangan di bawah ini :

Nama : Andrian Widhi Wicaksana
NIM/NIP/NIDN : 72210452
Program Studi : Sistem Informasi
Judul Karya Ilmiah : APLIKASI EKSTRAKSI FITUR LONG METHOD
PADA KODE PROGRAM BERBASIS MOBILE
DENGAN METRICS NOLV, CYCLO, DAN NCNDC

dengan ini menyatakan:

- a. bahwa karya yang saya serahkan ini merupakan revisi terakhir yang telah disetujui pembimbing/promotor/reviewer.
- b. bahwa karya saya dengan judul di atas adalah asli dan belum pernah diajukan oleh siapa pun untuk mendapatkan gelar akademik baik di Universitas Kristen Duta Wacana maupun di universitas/institusi lain.
- c. bahwa karya saya dengan judul di atas sepenuhnya adalah hasil karya tulis saya sendiri dan bebas dari plagiasi. Karya atau pendapat pihak lain yang digunakan sebagai rujukan dalam naskah ini telah dikutip sesuai dengan kaidah penulisan ilmiah yang berlaku.
- d. bahwa saya bersedia bertanggung jawab dan menerima sanksi sesuai dengan aturan yang berlaku berupa pencabutan gelar akademik jika di kemudian hari didapati bahwa saya melakukan tindakan plagiasi dalam karya saya ini.
- e. bahwa Universitas Kristen Duta Wacana tidak dapat diberi sanksi atau tuntutan hukum atas pelanggaran hak kekayaan intelektual atau jika terjadi pelanggaran lain dalam karya saya ini. Segala tuntutan hukum yang timbul atas pelanggaran dalam karya saya ini akan menjadi tanggung jawab saya pribadi, tanpa melibatkan pihak Universitas Kristen Duta Wacana.
- f. menyerahkan hak bebas royalti noneksklusif kepada Universitas Kristen Duta Wacana, untuk menyimpan, melestarikan, mengalihkan dalam media/format lain, mengelolanya dalam bentuk pangkalan data (*database*), dan mengunggahnya di Repositori UKDW tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan pemilik hak cipta atas karya saya di atas, untuk kepentingan akademis dan pengembangan ilmu pengetahuan.
- g. bahwa saya bertanggung jawab menyampaikan secara tertulis kepada Universitas Kristen Duta Wacana jika di kemudian hari terdapat perubahan hak cipta atas karya saya ini.

h. bahwa meskipun telah dilakukan pelestarian sebaik-baiknya, Universitas Kristen Duta Wacana tidak bertanggung jawab atas kehilangan atau kerusakan karya atau metadata selama disimpan di Repositori UKDW.

i. mengajukan agar karya saya ini: *(pilih salah satu)*

- ☒ Dapat diakses tanpa embargo.
- ☐ Dapat diakses setelah 2 tahun.*
- ☐ Embargo permanen.*

Embargo: penutupan sementara akses karya ilmiah.

*Halaman judul, abstrak, dan daftar pustaka tetap wajib dibuka.

Alasan embargo *(bisa lebih dari satu)*:

- ☐ dalam proses pengajuan paten.
- ☐ akan dipresentasikan sebagai makalah dalam seminar nasional/internasional.**
- ☐ akan diterbitkan dalam jurnal nasional/internasional.**
- ☐ telah dipresentasikan sebagai makalah dalam seminar nasional/internasional ... dan diterbitkan dalam prosiding pada bulan ... tahun ... dengan DOI/URL ... ***
- ☐ telah diterbitkan dalam jurnal ... dengan DOI/URL artikel ... atau vol./no. ... ***
- ☐ berisi topik sensitif, data perusahaan/pribadi atau informasi yang membahayakan keamanan nasional.
- ☐ berisi materi yang mengandung hak cipta atau hak kekayaan intelektual pihak lain.
- ☐ terikat perjanjian kerahasiaan dengan perusahaan/organisasi lain di luar Universitas Kristen Duta Wacana selama periode tertentu.
- ☐ Lainnya (mohon dijelaskan)

**Setelah diterbitkan, mohon informasikan keterangan publikasinya ke repository@staff.ukdw.ac.id

***Tuliskan informasi kegiatan atau publikasinya dengan lengkap.

Yogyakarta, 2 Februari 2025

Mengetahui,

Yang menyatakan,



Argo Wibowo, S.T., MT.
Tanda tangan & nama terang pembimbing
NIDN/NIDK 0516118902



Andrian Widhi Wicaksana
Tanda tangan & nama terang pemilik karya/penulis
NIM 72210452

HALAMAN PENGESAHAN

APLIKASI EKSTRAKSI FITUR LONG METHOD PADA KODE PROGRAM BERBASIS MOBILE DENGAN METRICS NOLV, CYCLO, DAN NCNDC

Oleh: ANDRIAN WIDHI WICAKSANA / 72210452

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Sistem Informasi Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal
22 Januari 2025

Yogyakarta, 28 Januari 2025
Mengesahkan,

Dewan Penguji:

1. Argo Wibowo, S.T., M.T.
2. Katon Wijana, S.Kom., M.T.
3. Umi Probeykti, S.Kom., M.L.I.S.



Dekan

(RESTYANDITO, S.Kom., MSIS., Ph.D.)

Ketua Program Studi

(Halim Budi Santoso, S.Kom., MT., MBA.,
Ph.D.)

HALAMAN PERSETUJUAN

Judul Skripsi : Aplikasi Ekstraksi Fitur Long Method Pada Kode
Program Berbasis Mobile Dengan Metrics NOLV,
CYCLO, dan NCNDC

Nama Mahasiswa : ANDRIAN WIDHI WICAKSANA

N I M : 72210452

Matakuliah : Skripsi

Kode : SI4046

Semester : Gasal

Tahun Akademik : 2024/2025

Telah diperiksa dan disetujui di Yogyakarta,
Pada tanggal 28 Januari 2025

Dosen Pembimbing I



Argo Wibowo, S.T., M.T.

Dosen Pembimbing II



Erick Kurniawan, S.Kom., M.Kom.

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

Aplikasi Ekstraksi Fitur Long Method Pada Kode Program Berbasis Mobile Dengan Metrics NOLV, CYCLO, dan NCNDC

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Sistem Informasi Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar kesarjanaan saya.

Yogyakarta, 28 Januari 2025



ANDRIAN WIDHI WICAKSANA
72210452

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yesus Kristus, atas segala berkat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "APLIKASI EKSTRAKSI FITUR LONG METHOD PADA KODE PROGRAM BERBASIS MOBILE DENGAN METRICS NOLV, CYCLO, DAN NCNDC" dengan baik sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana. Penulis menyadari bahwa hasil yang dicapai tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yesus Kristus yang telah mengkaruniakan berkat, ilmu, hikmat, pertolongan, kekuatan, dan perlindungan-Nya kepada penulis sehingga mampu menyelesaikan skripsi ini.
2. Bapak Argo Wibowo, S.T., M.T. dan Bapak Erick Kurniawan, S.Kom., M.Kom., selaku dosen pembimbing yang dengan sabar dan penuh perhatian telah memberikan bimbingan, wawasan, dan arahan yang sangat berarti selama penyusunan skripsi ini.
3. Bapak Katon Wijana, S.Kom., M.T. dan Ibu Umi Proboyekti, S.Kom., M.L.I.S. selaku dosen penguji yang dengan sabar dan penuh perhatian telah memberikan masukan, wawasan, dan arahan yang sangat berarti selama penyusunan skripsi ini.
4. Keluarga besar tercinta, terkhusus Ibunda tercinta dan saudari serta orang yang saya sayangi yang menjadi sumber kekuatan dan inspirasi selama perjalanan pendidikan ini. Terima kasih atas kasih sayang, doa, serta dukungan secara emosional.
5. Teman-teman terkasih yang sudah memberikan semangat, motivasi, dan bantuan yang tak ternilai.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, penulis terbuka untuk kritik dan saran yang membangun demi pembuatan skripsi yang lebih baik di masa mendatang. Semoga skripsi ini dapat

memberikan manfaat dan kontribusi positif bagi pengembangan ilmu pengetahuan, khususnya dalam bidang teknologi informasi.

Demikian kata pengantar ini penulis sampaikan. Terima kasih.

Andrian Widhi Wicaksana



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x
ABSTRAK	xi
ABSTRACT	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Spesifikasi Sistem.....	2
1.5 Tujuan dan Manfaat Penelitian	3
1.6 Tahapan Penelitian	3
1.6.1 Pengumpulan Data.....	3
1.6.2 Analisis Data	4
1.6.3 Desain Aplikasi	4
1.7 Metodologi Penelitian.....	4
1.8 Sistematika Penulisan	5
BAB 2 LANDASAN TEORI	6
2.1 Tinjauan Pustaka	6
2.2 Teori.....	8

2.2.1	NOLV_METHOD (Number of Local Variables)	8
2.2.2	CYCLO_METHOD (Cyclomatic Complexity)	8
2.2.3	NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD	9
2.2.4	Variabel	9
2.2.5	Long Method	10
2.2.6	Regex	10
BAB 3	PERANCANGAN SISTEM	12
3.1	Data Penelitian	12
3.2	Flowchart	14
BAB 4	PENERAPAN SISTEM	16
4.1	Implementasi Sistem	16
4.2	Analisis dan Pembahasan.....	25
4.2.1	Implementasi NOLV_METHOD (Number of Local Variables) ..	25
4.2.2	Implementasi CYCLO_METHOD (Cyclomatic Complexity).....	28
4.2.3	Implementasi NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHO D	32
BAB 5	PENUTUP	36
5.1	Kesimpulan	36
5.2	Saran	37
DAFTAR PUSTAKA		38
LAMPIRAN		40
A.	Listing Program.....	40
B.	Kartu Konsultasi.....	78
C.	Form Revisi Skripsi	80

DAFTAR GAMBAR

Gambar 3.1 <i>Flowchart</i> Sistem.....	14
Gambar 4.1 Halaman Summary Report.....	16
Gambar 4.2 Halaman Detailed Report.....	17
Gambar 4.3 Halaman Complexity Report.....	17
Gambar 4.4 Halaman Download Report 1	18
Gambar 4.5 Halaman Download Report 2.....	18
Gambar 4.6 Halaman CSV Hasil Download Report	19
Gambar 4.7 Kode Program NOLV	25
Gambar 4.8 Hasil Implementasi Kode Program NOLV	25
Gambar 4.9 Kode Program CYCLO_METHOD	28
Gambar 4.10 Hasil Implementasi CYCLO_METHOD.....	29
Gambar 4.11 Kode Program NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD..	32
Gambar 4.12 Hasil Implementasi NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD..	33
Gambar 4.11 Kode Program NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD..	23
Gambar 4.12 Hasil Implementasi NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD..	23

DAFTAR TABEL

Tabel 4.1 Tabel Perbandingan Sampel NOLV.....	20
Tabel 4.2 Tabel Perbandingan Sampel CYCLO_METHOD	22
Tabel 4.3 Tabel Perbandingan Sampel NUMBER_CONSTRUCTOR_	
NOTDEFAULTCONSTRUCTOR_METHOD	23



ABSTRAK

Skripsi ini bertujuan untuk mendeteksi dan melakukan seleksi code smell pada program berbasis mobile dengan menggunakan bahasa pemrograman Kotlin. Code smell adalah istilah yang menggambarkan indikasi bahwa suatu bagian kode memiliki kualitas yang buruk, yang dapat menyebabkan kesulitan dalam pemeliharaan, pengembangan program, dan kemungkinan terjadinya bug. Penelitian ini memfokuskan pada ekstraksi tiga metrik utama yang sering muncul dalam kode program Kotlin, yaitu NOLV_METHOD (Number of Local Variables), CYCLO_METHOD (Cyclomatic Complexity), dan NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD. Tujuan dari penelitian ini adalah untuk menganalisis seberapa buruk kualitas suatu program dengan menganalisis 3 metrik tersebut dan menyediakan alat analisis berbasis web yang dapat menampilkan hasilnya dalam format yang dapat diunduh. Proyek HyperSD yang dikembangkan dalam penelitian ini, menggunakan bahasa pemrograman Python dan framework Streamlit untuk membangun aplikasi yang interaktif dan efisien dalam mengekstrak serta menghitung metrik-metrik tersebut. Aplikasi ini menerapkan metode regex untuk menganalisis kode dan mengekstraksi informasi terkait jumlah variabel lokal, kompleksitas siklomatik, dan konstruktor non-default pada proyek Kotlin yang dianalisis. Hasil penelitian menunjukkan bahwa aplikasi ini mampu mengekstraksi data metrik secara akurat, dan menyajikan laporan yang dapat membantu pengembang dalam mengidentifikasi dan memperbaiki code smell dalam kode sumber. Selain itu, aplikasi ini juga dapat menghasilkan laporan dalam format CSV yang memudahkan pengguna dalam mengelola dan menganalisis data lebih lanjut.

Kata Kunci: CYCLO_METHOD, Kotlin, NOLV_METHOD, NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD, Streamlit.

ABSTRACT

This thesis aims to detect and select code smell in mobile-based programs using Kotlin programming language. Code smell is a term that describes an indication that a section of code is of poor quality, which can lead to difficulties in maintenance, program development, and possible bugs. This research focuses on extracting three main metrics that often appear in Kotlin program code, namely NOLV_METHOD (Number of Local Variables), CYCLO_METHOD (Cyclomatic Complexity), and NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD.

The purpose of this research is to analyze how poor the quality of a program is by evaluating these three metrics and provide a web-based analysis tool that can display the results in a downloadable format. The HyperSD project, developed in this research, uses the Python programming language and Streamlit framework to build an interactive and efficient application to extract and calculate these metrics. The application applies the regex method to analyze the code and extract information related to the number of local variables, cyclomatic complexity, and non-default constructors in the Kotlin projects analyzed. The results show that the app is able to extract metric data accurately, and present reports that can help developers identify and fix code smell in the source code. In addition, this application can also generate reports in CSV format that make it easier for users to manage and analyze the data further.

Keywords: CYCLO_METHOD, Kotlin, NOLV_METHOD, NUMBER_CONSTRUCTOR_NOTDEFAULTCONSTRUCTOR_METHOD, Streamlit.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Proyek HyperSD merupakan salah satu penelitian yang bertujuan untuk mendeteksi dan melakukan seleksi *code smell*. *Code smell* merupakan istilah yang digunakan dalam rekayasa perangkat lunak untuk menggambarkan tanda-tanda dalam kode sumber yang mungkin menunjukkan adanya masalah yang lebih mendalam. Penelitian sebelumnya menganggap bahwa *code smell* sebagai salah satu yang memiliki frekuensi tertinggi, yang mungkin memiliki dampak negatif terbesar terhadap kualitas perangkat lunak, dan memiliki aturan pendeteksian yang didefinisikan dalam literatur atau diimplementasikan pada alat yang tersedia (Arcelli Fontana *et al.*, 2016; Li *et al.*, 2017; Olbrich *et al.*, 2010). Terdapat 13 *Code smell* terdiri dari *Inappropriate Intimacy*, *Long Method*, *Spaghetti Code*, *Speculative Generality*, *God Class*, *Complex Class*, *Refused Bequest*, *Message Chain*, *Feature Envy*, *CDSBP*, *LPL*, *Lazy Class*, *Middle Man*. *Long method* adalah salah satu jenis *code smell* yang paling umum ditemukan dalam perangkat lunak. *Long method* memiliki penyebaran (*diffuseness*) tinggi di berbagai proyek perangkat lunak, baik skala kecil maupun besar, karena itu *long method* menjadi topik yang di ambil dalam penelitian ini (Palomba *et al.*, 2018). Meskipun *code smell* tidak secara langsung menyebabkan *bug* atau kesalahan dalam program, mereka menandakan bahwa ada sesuatu yang mungkin salah dengan desain atau struktur kode, yang dapat mengakibatkan masalah di masa depan jika tidak diperbaiki.

Proyek HyperSD dijalankan agar dapat mengetahui seberapa buruk atau kurang berkualitasnya suatu program. Pendekatan hybrid ini menggabungkan *machine learning* dan *rule-based metric*. Rule-based metric memungkinkan identifikasi *code smell* yang jelas berdasarkan aturan yang telah ditetapkan, sehingga memudahkan pengembang dalam memahami dan memperbaiki masalah. Penelitian ini adalah sebagian kecil dari proyek HyperSD yang berfokus pada persiapan ekstraksi fitur dari kode program kotlin. Penelitian sebelumnya telah

menyeleksi fitur dan didapati 3 metrik yang paling sering muncul yaitu *NOLV_method*, *CYCLO_method*, dan *number_constructor_NotDefaultConstructor_methods*, karena merupakan urutan 3 teratas dari fitur yang paling relevan maka metrik ini dijadikan penelitian. Tujuan dari ekstraksi ini adalah untuk mendukung proses deteksi code smells menggunakan pendekatan hybrid. Manfaat dari aplikasi web ini adalah menyediakan data uji dan data latih yang diperlukan untuk mendukung proyek HyperSD secara keseluruhan.

1.2 Rumusan Masalah

Long method meningkatkan kompleksitas kode, membuatnya sulit dipahami dan dipelihara. Keberadaan *Long method* membutuhkan upaya lebih untuk menyederhanakan kode, mengurangi risiko bug, dan meningkatkan keterbacaan serta efisiensi proses pengembangan.

1.3 Batasan Masalah

Proyek ini agar dapat melakukan ekstraksi fitur maka diperlukan 3 metrik diatas. Bahasa pemrograman yang dianalisis berupa Program Kotlin, dan akan ditampilkan dari *website* berbasis PWA. Penelitian ini akan meneliti 3 metrik teratas diantaranya adalah *NOLV_method*, *CYCLO_method*, dan *number_constructor_NotDefaultConstructor_methods*. 3 metrik ini diambil berdasarkan 3 metrik dengan hasil terbanyak.

1.4 Spesifikasi Sistem

Spesifikasi sistem yang dibangun terbagi dalam 3 hal, yaitu:

1. Spesifikasi software yang digunakan:
 1. Sistem Operasi Windows 11
 2. Visual Studio Code
 3. Excel (CSV)
2. Spesifikasi aplikasi yang dibangun, yaitu:
 1. Sistem menganalisis dan mengeskraksi seberapa buruknya suatu project yang menggunakan bahasa pemrograman kotlin

2. Sistem dapat memberikan hasil *output* dalam web berupa tampilan dalam bentuk array
3. Sistem yang dibangun dapat menghasilkan suatu hasil dari program yang dilakukan input dan dapat disimpan dalam bentuk *file* CSV

1.5 Tujuan dan Manfaat Penelitian

Hasil akhir dari penelitian ini akan menghasilkan output berupa tampilan di web yang dapat diunduh berupa *file* CSV yang didalamnya berisi informasi mengenai *NOLV_method*, *CYCLO_method*, dan *number_constructor_NotDefaultConstructor_methods*, 3 metrik ini termasuk *code smell* yang paling sering ditemukan dalam pemrograman mobile dengan bahasa pemrograman kotlin, dan dapat menampilkan berapa banyak jumlah *file*, kelas, fungsi, dan variabel dari 1 *file* program dengan bahasa pemrograman kotlin. Penelitian ini tidak hanya bertujuan untuk mendeteksi dan menganalisis long method, tetapi juga memastikan bahwa hasil perhitungan atau analisis yang dilakukan menggunakan metrik tersebut sudah sesuai dengan rumus atau definisi yang telah ditentukan akurat atau belum untuk masing-masing metrik, karena jika masalah ini tidak segera ditangani, dapat mengarah pada peningkatan waktu pemeliharaan, pengurangan kinerja aplikasi, serta potensi kesalahan dalam pengembangan aplikasi di masa depan.

1.6 Tahapan Penelitian

Tahapan penelitian terdiri dari:

1.6.1 Pengumpulan Data

Metode terakhir dalam mengumpulkan data adalah melalui analisis dokumen. Dalam pendekatan ini, peneliti memanfaatkan berbagai sumber informasi yang merekam peristiwa masa lalu sebagai objek penelitian. Sumber-sumber ini dapat berupa catatan tertulis, representasi visual, atau karya-karya monumental individu. Cakupan dokumen yang dapat digunakan sangat luas, meliputi jurnal pribadi, riwayat hidup, tulisan biografi, teks peraturan, serta berbagai bentuk dokumentasi lainnya. Metode ini memungkinkan peneliti untuk menggali informasi berharga dari rekaman sejarah dalam berbagai format.

1.6.2 Analisis Data

Pada tahap analisis data, dilakukan identifikasi dan pengelompokan terhadap data yang telah dikumpulkan. Proses ini bertujuan untuk mengklasifikasikan dan memisahkan data berdasarkan kategori tertentu agar lebih terstruktur. Dengan demikian, risiko redundansi data dapat diminimalkan, yang akan sangat membantu dalam mengelola data secara lebih efisien. Selain itu, pengelompokan ini juga memudahkan dalam meninjau dan memahami data yang relevan, sehingga hanya data yang benar-benar diperlukan untuk input yang dipertimbangkan. Pada akhirnya, langkah ini membantu menghasilkan analisis yang lebih akurat dan informatif.

1.6.3 Desain Aplikasi

HyperSD dibuat menggunakan framework streamlit dan bahasa pemrograman Python untuk antarmuka yang interaktif dan efisien dalam melakukan analisis terhadap kode sumber Kotlin. Streamlit adalah framework *open-source* berbasis Python yang memungkinkan pembuatan aplikasi web secara cepat dan tanpa memerlukan keahlian dalam pengembangan frontend atau JavaScript. Aplikasi ini memungkinkan pengguna mengunggah file ZIP yang berisi proyek Kotlin, yang kemudian dianalisis untuk menghasilkan tiga metrik, yaitu NOLV (Number of Local Variables), CYCLO (Cyclomatic Complexity), dan NCNDC (Number of Non-Default Constructors). Python digunakan untuk mengelola analisis data dan perhitungan metrik tersebut dengan menggunakan pustaka yang luas dan mendukung pengolahan data serta penghitungan kompleksitas kode. Keunggulan penggunaan Streamlit dan Python terletak pada kemudahan pengguna, kecepatan pengembangan, dan kemampuan untuk melakukan analisis yang mendalam terhadap kualitas kode dengan hasil yang dapat diunduh dalam format CSV.

1.7 Metodologi Penelitian

Penulis mengambil langkah-langkah berikut untuk menyelesaikan sistem yang akan dibangun:

- a. Kebutuhan sistem didapat melalui dataset dari proyek-proyek yang sudah dibuat yang menggunakan bahasa pemrograman kotlin

- b. Data tentang informasi perhitungan metrik di bangun berdasarkan penelitian-penelitian sebelumnya untuk dikembangkan dalam project ini.
- c. Menemukan perhitungan berdasarkan logika matematika dan statistika
- d. Pembangunan sistem sesuai kebutuhan pengguna agar dapat menghasilkan analisa dan ekstraksi yang akurat

1.8 Sistematika Penulisan

Penulisan skripsi diawali dengan pendahuluan dari Bab pertama. Bab 1 akan menjelaskan latar belakang pertanyaan penelitian, rumusan masalah ini akan menjadi dasar dari desain penelitian yang dilakukan, batasan masalah sistem, dan manfaat penelitian, tahapan penelitian dan sistematika penulisan laporan. Selain itu, Bab 2 berisi landasan teori dan tinjauan pustaka. Dalam tinjauan literatur, studi terkait akan dijelaskan aplikasi serupa yang masih terkait satu sama lain. Lalu dibagian landasan teori, akan menjelaskan teori-teori tentang visualisasi grafis yang nantinya akan diterapkan sebagai metodologi pengembangan aplikasi.

Bab 3 membahas analisis dan desain sistem. Bab ini akan dijelaskan bagaimana proses kerja sistem, data yang dibutuhkan sistem, dan gambaran basis data. Alur sistem akan dijelaskan menggunakan diagram alur. Kemudian, Bab 4 akan membahas Semua tentang aplikasi dan analisis sistem. Bab ini membahas bagaimana penerapan desain yang dikembangkan di Bab 3. Laporan ini ditutup di Bab 5, yaitu bagian kesimpulan dan saran penelitian. Bab ini menjelaskan hasil akhir dari sebuah penelitian dan saran dari penulis terkait pengembangan sistem yang lebih baik lagi kedepannya.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, penulis berhasil mengembangkan solusi untuk menghitung jumlah file, kelas, metode, variabel, dan package dalam proyek Kotlin. Tiga metrik yang sering muncul, yaitu jumlah *variabel local*(NOLV), jumlah percabangan logis(CYCLO), dan jumlah konstruktor non-default(NOCNDC), berhasil dihitung dan ditampilkan secara dinamis dalam web aplikasi HyperSD serta dapat diunduh dalam format CSV. Hal ini menunjukkan keberhasilan dalam menyediakan alat untuk monitoring dan analisis proyek.

Namun, dalam penelitian ini ditemukan adanya perbedaan antara hasil perhitungan metrik menggunakan kode program dengan hasil perhitungan manual (menggunakan rumus). Analisis error rate menggunakan metode MAPE menunjukkan hasil sebagai berikut:

1. Jumlah Variabel Lokal: MAPE sebesar 0.00%, mengindikasikan bahwa perhitungan menggunakan kode program sangat akurat dan sesuai dengan perhitungan manual.
2. Jumlah Percabangan Logis: MAPE sebesar 29.55%, mengindikasikan adanya perbedaan yang signifikan antara perhitungan kode program dan manual.
3. Jumlah Konstruktor Non-Default: MAPE sebesar 20.00%, mengindikasikan adanya perbedaan pada beberapa kasus antara perhitungan kode program dan manual.

Perbedaan *error rate* pada perhitungan jumlah percabangan logis dan konstruktor non-default menunjukkan adanya keterbatasan pada metode analisis yang digunakan dalam penelitian ini, terutama dalam menangani kompleksitas struktur kode dan perbedaan interpretasi dalam pendeteksian percabangan dan

konstruktor. Hal ini menekankan perlunya metode yang lebih akurat untuk mengurangi perbedaan ini dan memastikan hasil analisis yang lebih presisi.

5.2 Saran

Meskipun penelitian ini berhasil mengimplementasikan sistem untuk analisis metrik kode Kotlin dan mengidentifikasi area yang perlu ditingkatkan dalam proyek, masih terdapat ruang untuk pengembangan lebih lanjut. Maka dari itu, untuk penelitian selanjutnya disarankan untuk mempertimbangkan penggunaan metode Abstract Syntax Tree (AST) dalam analisis kode. Metode AST dapat memberikan representasi struktur kode yang lebih akurat dan detail dibandingkan dengan pendekatan berbasis regex yang digunakan dalam penelitian ini. Dengan menggunakan AST, diharapkan akurasi perhitungan metrik, terutama jumlah percabangan logis dan konstruktor non-default, dapat ditingkatkan secara signifikan dan *error rate* dapat diminimalkan.



DAFTAR PUSTAKA

- Dewangan, S., Rao, R. S., Mishra, A., & Gupta, M. (2022). Code Smell Detection Using Ensemble Machine. *applied sciences*, 1-22.
- Emden, E. v., & Moonen, L. (2002). Java Quality Assurance by Detecting Code Smells. 1-11.
- Fontana, F. A., Mäntylä, M. V., Zanoni, M., & Marino, A. (2015). Comparing and Experimenting Machine Learning Techniques. *Empirical Software Engineering*, 1143–1191.
- Lorenz, M., & Kidd, J. (1994). *Metrics of Object-Oriented Software*. United States: Prentice-Hall, Inc. Division of Simon and Schuster One Lake Street Upper Saddle River, NJ.
- Marinescu, R. (2005). Measurement and Quality in Object-Oriented Design. *Computer Society*, 1-4.
- MCCABE, T. J. (1976). A Complexity Measure. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 308-320.
- Sa'adah, U., Permatasari, D. I., Hardiansyah, F. F., Yunanto, A. A., Hasim, J. A., Wulandari, I., . . . Shihab, D. Q. (2022). TOOL REFACTORING OTOMATIS UNTUK MENANGANI LAZY CLASS CODE . *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 743- 749 .
- Syawanodya, I., Anggraini, D., Al-Hijri, F. M., & Ardimansyah, M. I. (2024). An Empirical Evaluation on the Effect of Refactoring Code Smells . *International Journal on Advanced Science Engineering Information Technology*, 214-223.
- Arcelli Fontana, F., Mäntylä, M. V., Zanoni, M., & Marino, A. (2016). Comparing and experimenting machine learning techniques for code smell detection. *Empirical Software Engineering*, 21(3), 1143–1191. <https://doi.org/10.1007/s10664-015-9378-4>

- Li, J., Wang, Z. L., Zhao, H., Gravina, R., Fortino, G., Jiang, Y., & Tang, K. (2017). Networked human motion capture system based on quaternion navigation. *BodyNets International Conference on Body Area Networks*. <https://doi.org/10.1145/0000000.0000000>
- Lorenz, M., & Kidd, J. (1994). *Object-oriented Software Metrics: A Practical Guide*. PTR Prentice Hall.
- Marinescu, R. (2005). Measurement and quality in object-oriented design. *IEEE International Conference on Software Maintenance, ICSM, 2005*, 701–704. <https://doi.org/10.1109/ICSM.2005.63>
- Mccabe, T. J. (1976). A Complexity Measure. In *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING* (Issue 4).
- Mitra, S. (2021). Regular expressions: A detailed study for the understanding of their role and methods for efficient application Soham Mitra. ~ 71 ~ *International Journal of Research in Circuits, Devices and Systems*, 2(2), 71–76. www.circuitsjournal.com
- Olbrich, S. M., Cruzes, D. S., & Sjøberg, D. I. K. (2010). Are all code smells harmful? A study of God Classes and Brain Classes in the evolution of three open source systems. *IEEE International Conference on Software Maintenance, ICSM*. <https://doi.org/10.1109/ICSM.2010.5609564>
- Palomba, F., Bavota, G., Penta, M. Di, Fasano, F., Oliveto, R., & Lucia, A. De. (2018). On the diffuseness and the impact on maintainability of code smells: a large scale empirical investigation. *Empirical Software Engineering*, 23(3), 1188–1221. <https://doi.org/10.1007/s10664-017-9535-z>
- Wicaksono, N. G. C., & Leong, H. (2021). PERBANDINGAN PERFORMA BAHASA PEMROGRAMAN JAVA DAN KOTLIN PADA ANDROID APPS. *PROXIES*, 4(2), 85–96.