

TUGAS AKHIR

**RANCANG BANGUN EKOSISTEM MANAJEMEN PENGETAHUAN
KODE SUMBER BERBASIS WEB DAN CLI MENGGUNAKAN METODE
EXTREME PROGRAMMING**



DISUSUN OLEH:

SOFYAN FARROS

NIM. 22106050033

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2026

LEMBAR PENGESAHAN



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-1033/Un.02/DST/PP.00.9/05/2026

Tugas Akhir dengan judul : Rancang Bangun Ekosistem Manajemen Pengetahuan Kode Sumber Berbasis Web dan CLI Menggunakan Metode Extreme Programming

yang dipersiapkan dan disusun oleh:

Nama : SOFYAN FARROS
Nomor Induk Mahasiswa : 22106050033
Telah diujikan pada : Senin, 11 Mei 2026
Nilai ujian Tugas Akhir : A-

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

TIM UJIAN TUGAS AKHIR



Valid ID: 6a1505fb8ab70

Ketua Sidang

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



Valid ID: 6a147e1547774

Penguji I

Dr. Ir. Aulia Faqih Rifa'i, M.Kom.
SIGNED



Valid ID: 6a140fab2f62

Penguji II

Ir. Maria Ulfah Siregar, S.Kom., MIT., Ph.D.
SIGNED



Valid ID: 6a1546e913a10

Yogyakarta, 11 Mei 2026
UIN Sunan Kalijaga

Dekan Fakultas Sains dan Teknologi

Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

LEMBAR PERNYATAAN

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Sofyan Farros
NIM : 22106050033
Program Studi : Informatika
Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya, bahwa tugas akhir saya yang berjudul **Rancang Bangun Ekosistem Manajemen Pengetahuan Kode Sumber Berbasis Web dan CLI Menggunakan Metode Extreme Programming** merupakan hasil pekerjaan penulis sendiri sepanjang pengetahuan penulis, bukan duplikasi atau saduran dari karya orang lain kecuali bagian tertentu yang penulis ambil sebagai bahan acuan. Apabila terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab penulis.

Yogyakarta, 21 April 2026

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA



Sofyan Farros
NIM. 22106050033

LEMBAR PERSETUJUAN



Universitas Islam Negeri Sunan Kalijaga



FM-UINSK-BM-05-03/R0

SURAT PERSETUJUAN TUGAS AKHIR

Hal : Persetujuan Tugas Akhir
Lamp : -

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

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

Nama : Sofyan Farros
NIM : 22106050033
Judul Tugas Akhir : Rancang Bangun Ekosistem Manajemen Pengetahuan Kode Sumber Berbasis Web dan CLI Menggunakan Metode Extreme Programming

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

Dengan ini kami berharap agar tugas akhir Saudara tersebut di atas dapat segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 21 April 2026

Pembimbing

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

NIP. 19770103 200501 1 003

INTISARI

Tugas akhir ini mengembangkan Lumbung Kode, sebuah ekosistem *Code Knowledge Management* hibrida yang menggabungkan aplikasi Web dan *Command Line Interface* (CLI) untuk mengatasi hambatan *context switching* dan fragmentasi kode. Menggunakan metode *Extreme Programming* (XP), sistem ini dibangun dengan teknologi *React.js*, *Node.js*, dan *Supabase*. Fitur otomatisasi dokumentasi diintegrasikan melalui Google Gemini API, sementara keamanan aset privat pengguna dijamin oleh kebijakan *Row Level Security* (RLS). Hasil pengujian melalui 95 skenario uji menunjukkan tingkat keberhasilan 100%. Lumbung Kode terbukti cukup efektif sebagai repositori kode yang aman, cerdas, dan adaptif bagi alur kerja pengembang perangkat lunak modern.

Kata Kunci: *Code Knowledge Management, Hybrid Ecosystem, Extreme Programming, CLI, Snippet.*



ABSTRACT

This capstone develops Lumbung Kode, a hybrid Code Knowledge Management ecosystem integrating Web and Command Line Interface (CLI) platforms to minimize context switching and code fragmentation. Utilizing the Extreme Programming (XP) method, the system is built with React.js, Node.js, and Supabase. Documentation automation is powered by the Google Gemini API, while private data security is enforced through Row Level Security (RLS) policies. Evaluation across 95 testing scenarios achieved a 100% success rate. Lumbung Kode is proven effective enough as a secure, intelligent, and adaptive code repository for modern software development workflows.

Keywords: *Code Knowledge Management, Hybrid Ecosystem, Extreme Programming, CLI, Snippet.*



MOTTO

وَأَنْ لَّيْسَ لِلْإِنْسَانِ إِلَّا مَا سَعَى ○ وَأَنَّ سَعْيَهُ سَوْفَ يُرَى

“Dan manusia hanya memperoleh apa yang telah diusahakannya,
Dan sesungguhnya usahanya itu kelak akan diperlihatkan (kepadanya)”

(An-Najm : 39-40)



LEMBAR PERSEMBAHAN

Segala puji bagi Dzat Penenun Takdir yang paling sempurna. Atas segala rida, bentangan rahmat, dan kekuatan yang diizinkan-Nya mengalir pada diri ini, sehingga ribuan baris kode, kerumitan logika, dan lelahnya pikiran akhirnya bermuara pada sebuah titik penyelesaian. Tanpa campur tangan-Nya, mustahil langkah yang tertatih ini mampu mencapai garis akhir.

Dengan segala kerendahan hati, karya sederhana ini saya persembahkan untuk:

1. Kedua orang tua tercinta

Sebagai bentuk bakti kecil untuk kedua pelindung kehidupan saya, Bapak dan Ibu. Untuk Bapak, Mujab (alm) yang sudah mendidik di masa kecil penulis dengan kasih sayang. Kemudian khususnya untuk Ibunda penulis, Rusyanti yang doanya senantiasa menjadi perisai penembus langit di setiap harinya.

2. Keluarga besar dan sahabat seperjuangan

Kepada seluruh keluarga besar dan rekan-rekan seperjuangan penulis di bangku perkuliahan. Dinamika kebersamaan kita, waktu yang dihabiskan untuk mengabdikan berorganisasi, dan setiap diskusi panjang telah mematangkan cara pandang saya. Kalian adalah ruang belajar kehidupan nyata yang tidak akan pernah saya temukan di dalam kelas.

3. Diri sendiri dan almamater

Dan untuk diri ini sendiri. Terima kasih sudah bertahan tegak, tetap disiplin, dan menolak untuk menyerah pada keadaan. Tugas akhir ini akan selalu menjadi saksi bisu atas waktu yang terkorbankan dan tekad yang terus diuji. Serta tak lupa, dedikasi ini saya haturkan untuk almamater tercinta Prodi Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta yang telah menjadi rahim bagi ilmu dan pemikiran saya.

4. Seluruh orang yang telah hadir di hidup dan mengajarkan kebaikan kepada penulis, walau hanya satu huruf.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “ Rancang Bangun Ekosistem Manajemen Pengetahuan Kode Sumber Berbasis Web dan Cli Menggunakan Metode *Extreme Programming*” dengan baik dan tepat waktu.

Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga Yogyakarta. Dalam proses penyusunan Tugas Akhir ini, penulis menyadari bahwa tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak.

Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Prof. Noorhaidi Hasan S.Ag., M.A., M.Phil., Ph.D. selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Prof. Dr. Dra. Hj. Khurul Wardati, M.Si. selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Bapak Dr. Muhammad Mustakim, S.T. M.T. selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Dr. Agus Mulyanto, S.Si., M.Kom., ASEAN Eng. selaku dosen penasihat akademik yang telah memberikan arahan dan bimbingan selama masa perkuliahan penulis.
5. Dr. Agung Fatwanto, S.Si., M.Kom., selaku pembimbing tugas akhir yang dengan sabar memberikan arahan, masukan, dan dukungan dalam proses penyusunan tugas akhir ini.
6. Seluruh dosen dan karyawan Program Studi Informatika UIN Sunan Kalijaga Yogyakarta yang telah memberikan ilmu, bantuan, serta pelayanan selama penulis menempuh pendidikan.
7. Kedua orang tua penulis, terutama ibunda tercinta Rusyanti, yang senantiasa memberikan doa, kasih sayang, dukungan, serta pengorbanan yang tiada henti demi keberhasilan penulis.

8. Kakak dan adik penulis, Muchammad Fahrurizal, Siti Nuzulurrohma Afthriyana, Sunan Muqorrobin, dan Laela Nisfiya Fitriana yang selalu memberikan semangat, dukungan, dan motivasi dalam setiap proses yang dilalui penulis.

9. Sahabat serta teman-teman seperjuangan yang telah memberikan dukungan, bantuan, serta kebersamaan selama proses perkuliahan hingga penyusunan Tugas Akhir ini.

10. Semua pihak yang tidak dapat disebutkan satu per satu yang telah memberikan bantuan, dukungan, dan doa dalam penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki berbagai keterbatasan, baik dari segi kualitas maupun kelengkapan materi yang disajikan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun guna menyempurnakan karya ini. Penulis juga berharap Tugas Akhir ini dapat memberikan manfaat serta menjadi dasar untuk pengembangan lebih lanjut di masa yang akan datang.

Yogyakarta, 21 April 2026

Penulis,



Sofyan Farros
22106050033

DAFTAR ISI

LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
LEMBAR PERSETUJUAN.....	iv
INTISARI.....	v
<i>ABSTRACT</i>	vi
MOTTO.....	vii
LEMBAR PERSEMBAHAN	viii
KATA PENGANTAR.....	ix
DAFTAR ISI	xi
Daftar Tabel.....	xiv
Daftar Gambar.....	xvi
Daftar Lampiran	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Permasalahan.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah	5
1.4 Tujuan	5
1.5 Manfaat	6
1.5.1 Manfaat Teoretis	6
1.5.2 Manfaat Praktis	6
BAB II KAJIAN PUSTAKA.....	8
2.1 Platform Berbasis Web untuk Berbagi Pengetahuan	8
2.2 Efisiensi Alur Kerja melalui <i>Command Line Interface</i> (CLI)	9

2.3	Kompleksitas Pencarian dan Pemahaman Kode.....	9
2.4	Pemanfaatan Kecerdasan Buatan dalam Reduksi Beban Dokumentasi Teknis.....	10
2.5	Penerapan Metode <i>Extreme Programming</i> (XP)	11
2.6	Analisis Kesenjangan (<i>Gap Analysis</i>).....	11
BAB III METODE PENGEMBANGAN SISTEM		14
3.1	Alat dan Bahan.....	14
3.1.1	Perangkat Keras	14
3.1.2	Perangkat Lunak	14
3.2	Waktu dan Tempat Pengembangan	15
3.3	Metode Pengembangan.....	15
3.3.1	Perencanaan (<i>Planning</i>)	16
3.3.2	Perancangan (<i>Designing</i>)	17
3.3.3	Implementasi (<i>Coding</i>)	17
3.3.4	Pengujian (<i>Testing</i>)	18
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM		20
4.1	Perencanaan (<i>Planning</i>)	20
4.1.1	Analisis Kebutuhan Fungsional	21
4.1.2	Analisis Kebutuhan Non Fungsional	22
4.1.3	<i>User Stories</i>	23
4.2	Iterasi Pertama	26
4.2.1	Perancangan (<i>Designing</i>).....	26
4.2.2	Implementasi (<i>Coding</i>)	39
4.3	Iterasi Kedua	42
4.3.1	Perancangan (<i>Designing</i>).....	43

4.3.2 Implementasi (<i>Coding</i>)	52
4.4 Iterasi Ketiga.....	59
4.4.1 Perancangan	59
4.4.2 Implementasi (<i>Coding</i>)	62
4.5 Pengujian.....	64
4.5.1 <i>Black Box Testing</i>	65
4.5.2 White Box Testing	78
4.5.3 <i>Security Testing</i>	87
4.5.4 Kesimpulan Pengujian	97
BAB V PENUTUP.....	99
5.1 Kesimpulan	99
5.2 Saran	100
DAFTAR REFERENSI.....	102
DAFTAR LAMPIRAN	I
Lampiran 1	I
Lampiran 2	II
Lampiran 3	III

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Terdahulu.....	12
Tabel 3.1 Spesifikasi Perangkat Keras	14
Tabel 3.2 Spesifikasi Perangkat Lunak	15
Tabel 4.1 Kebutuhan Fungsional.....	21
Tabel 4.2 Kebutuhan non-Fungsional	23
Tabel 4.3 <i>User Stories</i> Perancangan dan Pembangunan	24
Tabel 4.4 Konfigurasi Utama Iterasi Pertama	39
Tabel 4.5 Fungsi - Fungsi Utama `authStore.js`	40
Tabel 4.6 Fungsi - Fungsi Utama snippetStore.js	41
Tabel 4.7 Fitur di <i>Dashboard</i>	42
Tabel 4.8 Hasil Konfigurasi Iterasi Kedua	53
Tabel 4.9 Fungsi yang Menangani CRUD di Modul Koleksi	55
Tabel 4.10 Tabel Fungsi-fungsi Sosial	56
Tabel 4.11 Daftar Kombinasi Pintasan Papan Keyboard	58
Tabel 4.12 Daftar Rancangan Command untuk CLI.....	61
Tabel 4.13 Library untuk Mendukung Modul CLI	62
Tabel 4.14 <i>Blackbox Testing</i> Modul Autentikasi.....	65
Tabel 4.15 <i>Blackbox Testing</i> Modul CRUD.....	66
Tabel 4.16 <i>Blackbox Testing</i> Modul Eksplorasi dan Pencarian	68
Tabel 4.17 <i>Blackbox Testing</i> Modul Filter dan <i>Sort Dashboard</i>	69
Tabel 4.18 <i>Blackbox Testing</i> Modul Koleksi	70
Tabel 4.19 <i>Blackbox Testing</i> Modul Interaksi Sosial	71
Tabel 4.20 <i>Blackbox Testing</i> Modul AI Metadata <i>Autofill</i>	73
Tabel 4.21 <i>Blackbox Testing</i> Modul CLI Autentikasi	73
Tabel 4.22 <i>Blackbox Testing</i> Modul CLI <i>Upload</i>	74
Tabel 4.23 <i>Blackbox Testing</i> Modul CLI <i>List</i> dan <i>Get</i>	76
Tabel 4.24 <i>Blackbox Testing</i> Modul CLI <i>Search</i>	77
Tabel 4.25 Kriteria Fungsi untuk dilakukan <i>White Box Testing</i>	79
Tabel 4.26 Perhitungan <i>Cyclomatic Complexity</i> Fungsi <i>Push</i>	82

Tabel 4.27 Jalur Independen Fungsi <i>Push</i>	82
Tabel 4.28 Penghitungan <i>Cyclomatic Complexity</i> Fungsi <i>checkUser</i>	86
Tabel 4.29 Jalur Independen Fungsi <i>checkUser</i>	86
Tabel 4.30 Daftar <i>Tools</i> untuk Pengujian Keamanan	87
Tabel 4.31 Uji Isolasi Data via Postman (<i>Direct API Call</i>)	87
Tabel 4.32 Simulasi Langsung via Supabase SQL Editor.....	89
Tabel 4.33 Uji Keamanan Token via JWT.io	91
Tabel 4.34 Uji Keamanan Token CLI - <i>Expired</i> Token dan Penyimpanan	93
Tabel 4.35 Uji Keamanan Route dan Sesi <i>Browser</i> via <i>DevTools</i>	95
Tabel 4.36 Kesimpulan Pengujian.....	97

DAFTAR GAMBAR

Gambar 3.1 Siklus <i>Extreme Programming</i>	16
Gambar 4.1 Desain Arsitektur Sistem.....	27
Gambar 4.2 <i>Use Case</i> Autentikasi Sistem	28
Gambar 4.3 <i>Activity Diagram</i> Registrasi.....	29
Gambar 4.4 <i>Activity Diagram Login</i>	30
Gambar 4.5 <i>Use Case Diagram</i> Manajemen <i>Snippet</i>	31
Gambar 4.6 <i>Activity Diagram</i> Tambah <i>Snippet</i>	32
Gambar 4.7 <i>Activity Diagram Import Snippet</i>	33
Gambar 4.8 <i>Activity Diagram Edit Snippet</i>	34
Gambar 4.9 <i>Activity Diagram Delete Snippet</i>	35
Gambar 4.10 <i>Entity Relationship Diagram</i> Iterasi Pertama.....	36
Gambar 4.11 Wireframe Login Page.....	37
Gambar 4.12 Wireframe Dashboard Page	37
Gambar 4.13 Wireframe Modal <i>Popup</i> Tambah <i>Snippet</i> Baru.....	38
Gambar 4.14 Wireframe Edit <i>Snippet</i>	38
Gambar 4.15 <i>Command Line</i> Inisiasi Awal Project	39
Gambar 4.16 Hierarki File yang Dihasilkan di Iterasi Pertama	40
Gambar 4.17 <i>Use Case Diagram</i> Iterasi Kedua	43
Gambar 4.18 <i>Activity Diagram</i> Eksplorasi dan Pencarian <i>Snippet</i>	44
Gambar 4.19 <i>Activity Diagram</i> Manajemen Koleksi.....	45
Gambar 4.20 <i>Activity Diagram Like Snippet</i>	46
Gambar 4.21 <i>Activity Diagram Fork Snippet</i>	47
Gambar 4.22 <i>Activity Diagram</i> Analisis Kode dengan AI.....	48
Gambar 4.23 Perkembangan <i>Entity Relationship Diagram</i> Iterasi Kedua	49
Gambar 4.24 Wireframe <i>Explore</i> dengan Tombol Like, Copy, dan Download.....	50
Gambar 4.25 Wireframe <i>Snippet Card</i> di <i>Dashboard</i> dengan Ikon Folder.....	51
Gambar 4.26 Wireframe Modifikasi Modal <i>Popup</i> Tambah <i>Snippet</i>	51
Gambar 4.27 Wireframe Modifikasi Edit Form	52
Gambar 4.28 Representasi Fungsi <i>Push</i>	80

Gambar 4.29 <i>Flow Graph</i> Fungsi <i>Push</i>	81
Gambar 4.30 Hasil Pengujian Fungsi <i>Push</i> dengan <i>Vitest</i>	83
Gambar 4.31 Representasi Fungsi <i>checkUser</i>	84
Gambar 4.32 <i>Flow Graph</i> Fungsi <i>checkUser</i>	85
Gambar 4.33 Hasil Pengujian Fungsi <i>Push</i> dengan <i>Vitest</i>	86
Gambar 4.34 Uji <i>Request Delete Snippet User B</i> dengan Token <i>User A</i> via <i>Postman</i>	88
Gambar 4.35 Uji <i>Request Patch</i> via <i>curl</i>	88
Gambar 4.36 Uji <i>Request</i> Tanpa <i>Authorization Header</i> via <i>Postman</i>	89
Gambar 4.37 Uji Hapus <i>Snippet User B</i> dengan Klaim <i>JWT</i> Sebagai <i>User A</i>	90
Gambar 4.38 Uji <i>Insert Snippet</i> ke <i>User B</i> dengan Klaim <i>JWT</i> Sebagai <i>User A</i>	90
Gambar 4.39 Uji Menampilkan Seluruh <i>Snippet User</i> dengan Klaim <i>JWT</i> Sebagai <i>User A</i>	91
Gambar 4.40 Tidak Ada Data Sensitif pada Hasil Dekode Token Di <i>JWT.Io</i>	92
Gambar 4.41 Sistem API Menolak Token yang Telah Dimanipulasi Secara Ilegal	93
Gambar 4.42 Token Sudah <i>Expired</i>	93
Gambar 4.43 Penolakan Sistem Terhadap Penggunaan Token yang Telah <i>Expired</i>	93
Gambar 4.44 Penolakan Sistem Akibat Token yang Telah <i>Expired</i>	95
Gambar 4.45 Isolasi Kredensial di Direktori <i>Appdata</i>	95
Gambar 4.46 Kata Sandi disensor pada Terminal <i>CLI</i>	95

DAFTAR LAMPIRAN

Lampiran 1. Ringkasan Tugas Akhir

Lampiran 2. Daftar Singkatan

Lampiran 3. User Stories



BAB I

PENDAHULUAN

1.1 Latar Belakang Permasalahan

Sudah sejak lama, pemanfaatan kembali kode (*code reuse*) dipandang sebagai salah satu cara utama untuk meningkatkan kualitas dan efisiensi pengembangan perangkat lunak, karena memungkinkan pengembang untuk menggunakan kode yang sudah teruji, alih-alih menulis ulang seluruh kode dari awal [1]. Namun dalam praktik nyatanya, penggunaan kembali kode lama sering kali terhambat oleh berbagai tantangan seperti susahnya mencari kode yang relevan, terbatasnya modularitas, dan masalah keterbatasan pemeliharaan kode yang menyebabkan upaya *refactoring* dan perancangan ulang tidak selalu berdampak nyata pada peningkatan kualitas dan efisiensi pengembangan perangkat lunak [2]. Di sisi lain, fragmentasi informasi pengetahuan pemrograman, misalnya tersebar di berbagai repositori, dokumentasi, dan sumber daring, mendorong fenomena “*reinventing the wheel*”, yaitu ketika pengembang dalam tim atau individu berulang kali membangun solusi serupa karena tidak mampu menemukan atau menilai kembali artefak yang sudah ada secara efektif [1]. Kombinasi antara tantangan teknis dalam modularitas dan *reuse*, serta masalah manajemen pengetahuan dan pencarian semantik kode, menjadikan isu *code reuse* dan fragmentasi sebagai masalah strategis yang perlu ditangani secara sistematis dalam dunia pengembangan dan rekayasa perangkat lunak modern [1], [2].

Selain itu, dari perspektif *knowledge management* dalam rekayasa perangkat lunak, fragmentasi informasi teknis dan artefak pengetahuan (kode, dokumentasi, *requirement*, dan keputusan desain) menjadi salah satu akar masalah rendahnya *reuse* dan seringkali tim “*reinventing the wheel*” dalam pengembangan perangkat lunak. Ketika pengetahuan tersebar di berbagai repositori dan tidak terindeks secara semantik, pengembang kesulitan menemukan solusi yang relevan dan cenderung menulis ulang fungsi serupa, alih-alih memanfaatkan kembali komponen yang sudah ada [1]. Dalam konteks metode agile, komponen seperti *user story* berperan

sebagai *boundary object* yang menjembatani perspektif bisnis dan teknis, namun juga dapat menjadi sumber fragmentasi informasi jika tidak dikelola dan ditransformasikan dengan baik ke artefak rancangan dan implementasi, sehingga pengetahuan yang tertuang di dalamnya sulit dilacak dan digunakan kembali pada siklus-siklus berikutnya [3].

Pendekatan pencarian kode berbasis semantik yang memetakan *query natural language* dan fragmen kode ke dalam representasi terstruktur berupaya mengurangi fragmentasi ini dengan menjembatani kesenjangan antara bahasa alami dan bahasa pemrograman, sehingga pengetahuan lama dapat lebih mudah ditemukan dan diintegrasikan dalam proyek baru [1]. Dengan demikian, pengelolaan pengetahuan yang efektif, pengurangan fragmentasi informasi, dan pemanfaatan artefak seperti *user story* secara terintegrasi menjadi prasyarat penting untuk mencegah praktik “*reinventing the wheel*” dan memaksimalkan manfaat *code reuse* dalam siklus hidup pengembangan perangkat lunak modern [1].

Di sisi lain, isu *context switching* antara terminal dan browser dapat diposisikan sebagai bentuk interupsi teknologi yang mengganggu keberlangsungan proses pekerjaan utama pengembang. Wiesche mendefinisikan interupsi secara luas sebagai segala sesuatu yang mengganggu atau menunda anggota tim atau organisasi saat sedang mengerjakan tugas, serta memasukkan di dalamnya “*technology-induced interruptions*” yang datang dari lingkungan kerja eksternal [4]. Perpindahan berulang dari terminal ke peramban untuk mencari informasi, membuka dokumentasi, atau mengecek status alat lain termasuk kategori “*interruptions imposed by the external environment*”, misalnya notifikasi, panggilan, atau praktik kerja harian yang memaksa pengembang meninggalkan tugas utama dan kemudian perlu waktu untuk kembali fokus [4]. Studi ini menunjukkan bahwa pengembang sampai harus menjadwalkan waktu tenang (*quiet time*) karena setelah jam tertentu, keberlangsungan alur kerja mereka terinterupsi oleh agenda sinkronisasi seperti *meeting*, jeda personal, dan interaksi sosial. Mereka melaporkan bahwa interupsi seperti ini memaksa mereka meninggalkan apapun aktivitas mereka saat ini dan membutuhkan waktu lagi untuk kembali ke aktivitas tersebut dengan fokus [4]. Dengan analogi yang sama, setiap kali pengembang

harus meninggalkan terminal, beralih ke *browser*, lalu kembali lagi, terjadi pola interupsi eksternal yang memecah alur kerja, mengganggu konsentrasi mendalam, dan menambah *overhead* untuk memulihkan konteks tugas. Hal ini menjadi dasar argumen perlunya fitur-fitur *Command Line Interface* (CLI) dasar (misalnya integrasi pencarian, dokumentasi, atau cek status) agar mengurangi perpindahan konteks terminal browser dan demi menjaga fokus.

Selain isu *code reuse*, *context switching*, ada juga isu dokumentasi manual yang dapat ditekan sebagai salah satu titik lemah klasik dalam pengembangan perangkat lunak modern dan data-science, yang membuat fitur *Artificial Intelligence* (AI) relevan sebagai solusi. Dalam konteks pengembangan perangkat lunak yang berkelanjutan, studi pemetaan sistematis menemukan bahwa dokumentasi sering kali tampak buruk, dengan tantangan seperti dokumentasi dianggap sebagai pemborosan, sulit dipahami, dan sering tidak sinkron dengan perangkat lunak karena fokus jangka pendek hanya pada *software* yang berjalan. Dokumentasi informal dan artefak pengembangan sering dibiarkan apa adanya, sehingga kualitas dan kelengkapannya rendah [5]. Padahal hal ini berdampak pada pemeliharaan, *reuse*, dan bahkan reliabilitas hasil ilmiah karena pengguna berisiko memakai perangkat lunak secara keliru [6]. Bahkan dalam ranah *data science*, studi pada *notebook* komputasional menemukan bahwa praktisi sering hanya memperhatikan kode dan mengabaikan pembuatan atau pembaruan dokumentasi ketika iterasi cepat, sehingga banyak bagian kerja tidak pernah terdokumentasi dengan baik [7]. Untuk menjawab masalah ini, beberapa karya empat tahun terakhir mengeksplorasi AI sebagai solusi otomatisasi dokumentasi menggunakan *deep learning*, pencarian *Application Programming Interface* (API), dan *prompts* untuk menghasilkan dokumentasi otomatis yang mengingatkan pengguna pada bagian yang akan mereka abaikan, mengurangi waktu penulisan dokumentasi, dan meningkatkan kepuasan terhadap kualitas dokumentasi [7].

Berdasarkan pemaparan di atas, terdapat urgensi untuk menghadirkan solusi terintegrasi yang mampu mengatasi masalah fragmentasi pengetahuan, inefisiensi akibat *context switching*, dan rendahnya kualitas dokumentasi. Diperlukan sebuah ekosistem yang tidak hanya berfungsi sebagai repositori kode (Web), tetapi juga

tertanam langsung dalam alur kerja pengembang melalui terminal (CLI) serta dilengkapi kecerdasan untuk mengotomatisasi dokumentasi. Untuk menjawab permasalahan inefisiensi dan kebutuhan adaptasi fitur yang cepat, pengembangan sistem ini membutuhkan metode yang responsif dan berorientasi pada kualitas kode. Metode pengembangan *Extreme Programming* (XP) dipilih karena karakteristiknya yang mengutamakan umpan balik cepat (*rapid feedback*), pengujian berkelanjutan (*continuous testing*), dan kesederhanaan desain (*simple design*) [8]. Pendekatan ini dinilai paling tepat untuk menangani dinamika pengembangan *tools* yang bersifat eksploratif seperti Lumbung Kode, di mana integrasi fitur cerdas dan antarmuka ganda (Web-CLI) perlu dikembangkan dalam siklus iterasi pendek guna menjamin keandalan sistem. Oleh karena itu, tugas akhir ini bertujuan untuk merancang dan membangun sebuah sistem *Code Knowledge Management* bernama Lumbung Kode. Pengembangan sistem ini akan menerapkan metode *Extreme Programming* (XP) untuk memastikan adaptabilitas terhadap kebutuhan pengembang yang dinamis serta pengiriman fitur yang cepat dan berkualitas[8].

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dapat dinyatakan sebagai berikut:

1. Bagaimana merancang arsitektur ekosistem *Code Knowledge Management* yang mengintegrasikan antarmuka Web dan CLI untuk mengatasi masalah fragmentasi aset kode serta mengurangi inefisiensi alur kerja *context switching* pengembang?
2. Bagaimana efektivitas integrasi *Generative AI* dalam mengotomatisasi pendokumentasian metadata aset kode guna meningkatkan efisiensi alur kerja dan mengurangi beban manual pengembang?
3. Bagaimana menerapkan metode *Extreme Programming* (XP) dalam perancangan dan pembangunan sistem "Lumbung Kode" untuk menjamin

kualitas perangkat lunak dan adaptabilitas terhadap kebutuhan fitur yang dinamis?

1.3 Batasan Masalah

Agar penelitian ini tetap terfokus pada tujuan awal dan tidak meluas dari ruang lingkup yang telah ditentukan, maka ditetapkan batasan masalah sebagai berikut:

1. Ekosistem dikembangkan hanya pada dua antarmuka, yaitu Web (SPA) dan CLI, tanpa mencakup ekstensi IDE maupun add-on peramban
2. Integrasi platform eksternal (GitHub) dibatasi pada pengambilan data melalui URL publik, bukan integrasi API secara menyeluruh.
3. Pemanfaatan Generative AI dibatasi pada otomasi metadata dan validasi integritas konten (AI Quality Gate).
4. Evaluasi sistem difokuskan pada pemenuhan kebutuhan fungsional, tanpa melakukan pengujian kegunaan (Usability Testing) secara formal.

1.4 Tujuan

Tugas akhir ini bertujuan untuk:

1. Merancang dan membangun ekosistem manajemen pengetahuan kode terpusat berbasis Web *Single Page Application* dan CLI yang memfasilitasi penyimpanan, pencarian, dan penggunaan kembali kode secara efisien guna meminimalisir hambatan alur kerja pengembang.
2. Mengimplementasikan dan menguji fitur otomasi metadata berbasis Generative AI untuk mempercepat proses dokumentasi aset kode dan meminimalisir hambatan kognitif saat menyimpan pengetahuan kode.
3. Menerapkan metode pengembangan *Extreme Programming (XP)* dengan siklus iteratif, pengujian berkelanjutan (*continuous testing*), desain sederhana (*simple design*), dan perbaikan struktur kode (*refactoring*) demi menghasilkan perangkat lunak berkualitas dan adaptif.

1.5 Manfaat

Tugas akhir ini diharapkan dapat memberikan manfaat baik secara teoretis maupun praktis, sebagai berikut:

1.5.1 Manfaat Teoretis

Kontribusi pada Rekayasa Perangkat Lunak: Memberikan wawasan ilmiah mengenai efektivitas arsitektur ekosistem hibrida (*hybrid ecosystem*) yang menggabungkan aplikasi Web dan *Command Line Interface* (CLI) dalam meningkatkan efisiensi manajemen pengetahuan kode.

Implementasi Fitur Cerdas dalam Pengembangan: Memberikan wawasan mengenai pola integrasi (*integration pattern*) antara aplikasi konvensional dan layanan *Generative AI* untuk mengotomatisasi dokumentasi teknis dan meningkatkan efisiensi pemeliharaan kode.

Validasi Metode XP: Memberikan bukti empiris terkait penerapan metode *Extreme Programming* (XP) dalam pengembangan proyek perangkat lunak skala kecil hingga menengah yang membutuhkan adaptabilitas tinggi terhadap fitur-fitur kompleks.

1.5.2 Manfaat Praktis

Bagi Pengembang Perangkat Lunak (*Developer*):

1. Menyediakan alat bantu (*tools*) yang dapat meminimalisir waktu pencarian kode berulang.
2. Mengurangi beban kognitif akibat perpindahan konteks (*context switching*) melalui integrasi alat CLI yang menyatu dengan alur kerja terminal.
3. Membantu menjaga kerapian dokumentasi kode secara otomatis tanpa menambah beban kerja manual.

Bagi Penulis:

1. Sebagai sarana implementasi dan pendalaman kompetensi teknis dalam membangun arsitektur *Full-Stack* (React, Node.js, Supabase) serta penerapan disiplin *Software Engineering* modern.

Bagi Akademisi dan Peneliti Selanjutnya:

1. Dapat dijadikan referensi dan atau rujukan bagi penelitian atau selanjutnya yang ingin mengembangkan sistem manajemen pengetahuan berbasis AI atau mengeksplorasi integrasi antarmuka pengembang (*Developer Experience/DX*).



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi, dan pengujian yang telah dilakukan terhadap ekosistem "Lambung Kode", dapat disimpulkan bahwa sistem yang dibangun telah menjawab permasalahan utama yang diangkat dari tugas akhir ini, yaitu kebutuhan akan sistem manajemen *snippet* kode terpusat yang terstruktur dan mudah diakses oleh pengembang.

Pertama, dari sisi pengembangan sistem, "Lambung Kode" berhasil diimplementasikan sebagai sebuah ekosistem manajemen *snippet* yang terintegrasi dengan dua antarmuka utama, yaitu aplikasi berbasis web dan *Command Line Interface* (CLI) yang terhubung melalui Supabase sebagai *Backend-as-a-Service* (BaaS). Integrasi ini memungkinkan pengembang untuk mengelola *snippet* dengan lebih mudah berkat dukungan dua *interface* utama, sehingga dapat mengurangi *context switching* dalam alur kerja sehari-hari.

Kedua, proses pengembangan yang dibagi ke dalam tiga iterasi telah berjalan sesuai dengan rencana. Iterasi pertama berfokus pada pembangunan fondasi inti, meliputi fitur autentikasi, *setup* basis data, dan operasi CRUD dasar. Iterasi kedua berfokus pada pengayaan fitur, seperti sistem koleksi (*folders*), *Full-Text Search* berbasis PostgreSQL (GIN Index), fitur sosial (*like, fork, copy*), notifikasi *real-time*, serta integrasi AI Google Gemini untuk memudahkan pengguna mengisi metadata *snippet* secara otomatis. Iterasi ketiga menitikberatkan pada pengembangan dan publikasi paket '*lambung-cli*' ke *registry* NPM, serta finalisasi proses *deployment* sehingga ekosistem dapat digunakan secara *end-to-end* oleh pengembang.

Ketiga, dari sisi kualitas perangkat lunak, hasil pengujian menunjukkan bahwa seluruh fungsi utama, baik pada aplikasi web maupun CLI, telah berjalan sesuai dengan spesifikasi. Dari total 95 skenario pengujian yang telah disusun dan dibagi ke tiga bagian pengujian, *Black Box Testing*, *White Box Testing*, dan *Security Testing* seluruhnya dinyatakan lulus 100%. Capaian ini mengindikasikan bahwa

sistem telah cukup andal dalam menangani alur fungsi yang dirancang, serta relatif aman dari celah manipulasi umum, seperti manipulasi *local storage*. Dengan demikian, dapat disimpulkan bahwa ekosistem "Lambung Kode" cukup layak digunakan sebagai solusi awal untuk manajemen dan pemanfaatan *snippet* kode secara terpusat bagi pengembang.

5.2 Saran

Meskipun "Lambung Kode" telah berhasil dibangun dan memenuhi kebutuhan fungsional yang dirumuskan dalam tugas akhir ini, tentu masih terdapat beberapa peluang pengembangan yang dapat dilakukan agar sistem dapat menjadi lebih matang, skalabel, dan memiliki nilai lebih tinggi bagi komunitas pengembang perangkat lunak.

1. Pengembangan Ekstensi IDE (*Integrated Development Environment*):

Disarankan untuk mengembangkan ekstensi atau *plugin* untuk IDE populer seperti *Visual Studio Code*. Dengan adanya integrasi langsung ke dalam editor kode, pengguna dapat lebih efisien dalam menyimpan, mencari, dan mengambil *snippet* tanpa perlu berpindah jendela aplikasi.

2. Fitur Kolaborasi dan Ruang Kerja Tim (*Organization Workspace*):

Saat ini Lambung Kode masih berfokus pada penggunaan individu dan berbagi *snippet* secara publik. Pengembangan fitur ruang kerja tim (*Team/Organization Workspace*) yang mendukung kolaborasi terkontrol, berbagi sumber daya internal, serta pengelolaan *snippet* privat pada level organisasi atau perusahaan akan memperluas cakupan penggunaan dan memperluas *value* sistem.

3. Peningkatan Kapabilitas AI Generatif:

Integrasi AI (Google Gemini) yang saat ini digunakan untuk otomatisasi pengisian metadata masih dapat dikembangkan lebih jauh. Pengembangan fitur *code refactoring*, *code translation* antar bahasa pemrograman, maupun *code generation* berbasis *natural language prompt* berpotensi untuk menjadikan ekosistem "Lambung Kode" sebagai asisten cerdas dalam pengelolaan dan penulisan kode. Bukan sekadar repositori *snippet*.

4. Dukungan Sinkronisasi dengan Layanan Eksternal

Pengembangan fitur sinkronisasi lanjutan dengan layanan pihak ketiga, seperti GitHub Gist, GitLab *Snippets*, atau layanan *cloud storage* lainnya, akan memberikan fleksibilitas tambahan bagi pengguna dalam menyimpan dan mencadangkan *snippet*. Integrasi ini juga dapat mempermudah migrasi data dan adopsi "Lumbung Kode" di lingkungan pengembangan yang sudah memiliki ekosistem tersendiri.

Dengan mengimplementasikan saran-saran tersebut pada penelitian dan pengembangan selanjutnya, diharapkan "Lumbung Kode" dapat berkembang menjadi ekosistem manajemen pengetahuan kode sumber yang lebih komprehensif, adaptif, dan siap diintegrasikan ke dalam praktik *software development* profesional pada skala yang lebih luas.

DAFTAR REFERENSI

- [1] N. Bibi, A. Maqbool, T. Rana, F. Afzal, A. Akgul, and S. M. Eldin, "Enhancing Semantic Code Search With Deep Graph Matching," *IEEE Access*, vol. 11, pp. 52392–52411, 2023, doi: 10.1109/ACCESS.2023.3263878.
- [2] S. Rochimah, T. R. Hadiningrum, B. D. Mardiana, D. O. Siahaan, J. A. Rizky, and M. S. Ary, "A Maintainability Framework to Ensure the Software Quality in Object-Oriented Programming," *IEEE Access*, vol. 13, pp. 195796–195821, 2025, doi: 10.1109/ACCESS.2025.3633265.
- [3] T. Sporseem, T. Dingsøyr, and K.-J. Stol, "User stories as boundary objects in agile requirements engineering: A theoretical literature review," *Journal of Systems and Software*, p. 112693, Mar. 2025, doi: 10.1016/j.jss.2025.112693.
- [4] M. Wiesche, "Interruptions in Agile Software Development Teams," 2021.
- [5] T. Theunissen, U. van Heesch, and P. Avgeriou, "A mapping study on documentation in Continuous Software Development," *Inf. Softw. Technol.*, vol. 142, Feb. 2022, doi: 10.1016/j.infsof.2021.106733.
- [6] M. Vidoni and Z. Codabux, "Towards a taxonomy of Roxygen documentation in R packages," *Empir. Softw. Eng.*, vol. 28, no. 4, Jul. 2023, doi: 10.1007/s10664-023-10345-4.
- [7] A. Y. Wang *et al.*, "Documentation Matters: Human-Centered AI System to Assist Data Science Code Documentation in Computational Notebooks," *ACM Transactions on Computer-Human Interaction*, vol. 29, no. 2, Apr. 2022, doi: 10.1145/3489465.
- [8] D. Jean Cross Sihombing, "Development of Academic Dashboard Application with Extreme Programming Method to Optimize Decision Making in Higher Education." [Online]. Available: <http://ejournal.seaninstitute.or.id/index.php/InfoSains>

- [9] S. Ranjan Dwivedi, A. Kumar, M. Singh, and V. Krishna Mishra, "WEB BASED KNOWLEDGE SHARE PLATFORM," *International Research Journal of Engineering and Technology (IRJET)*, vol. 12, no. 4, 2025, [Online]. Available: www.irjet.net
- [10] K. Rao, "Code Snippet Manager Using Flask and SQLite," *Scientific Journal of Artificial Intelligence and Blockchain Technologies*, vol. 2, no. 3, Sep. 2025, doi: 10.63345/sjaibt.v2.i3.301.
- [11] Dr. B. Rama Mohan, Abhed Chainani, A. Krishna Kartheek, Ch. A. S. Praneeth, and Kaustubh Shakelli, "IntelliDev: AI-Powered Command Line Tool for Developers," *international journal of engineering technology and management sciences*, vol. 8, no. 3, pp. 161–165, 2024, doi: 10.46647/ijetms.2024.v08i03.019.
- [12] W. Sun *et al.*, "A Survey of Source Code Search: A 3-Dimensional Perspective," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 6, Jun. 2024, doi: 10.1145/3656341.
- [13] B. Zhang, L. Fu, P. Liang, J. Yu, and C. Wang, "Demystifying Code Snippets in Code Reviews: A Study of the OpenStack and Qt Communities and A Practitioner Survey," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Apr. 2024, [Online]. Available: <http://arxiv.org/abs/2307.14406>
- [14] Aarti, "Generative AI in Software Development: An Overview and Evaluation of Modern Coding Tools," *International Journal for Multidisciplinary Research*, vol. 6, no. 3, 2024, [Online]. Available: www.ijfmr.com
- [15] D. Yang *et al.*, "DocAgent: A Multi-Agent System for Automated Code Documentation Generation," *ArXiv*, May 2025, [Online]. Available: <http://arxiv.org/abs/2504.08725>
- [16] D. Jean Cross Sihombing, "Development of Academic Dashboard Application with Extreme Programming Method to Optimize Decision Making in Higher Education," *Jurnal Info Sains :*

- Informatika dan Sains*, vol. 13, 2023, [Online]. Available: <http://ejournal.seaninstitute.or.id/index.php/InfoSains>
- [17] Chandra Ramadhan, Mamok Andri Senubekti, and Dien Amalia, "Penerapan Metodologi Agile dalam Pengembangan Perangkat Lunak," *Router : Jurnal Teknik Informatika dan Terapan*, vol. 3, no. 2, pp. 10–15, Apr. 2025, doi: 10.62951/router.v3i2.411.
 - [18] A. Shrivastava, I. Jaggi, N. Katoch, D. Gupta, and S. Gupta, "A Systematic Review on Extreme Programming," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jul. 2021. doi: 10.1088/1742-6596/1969/1/012046.
 - [19] M. F. Azhar, S. Munir, and Z. Imaduddin, "Multi-Tenant Platform for Web-Based Library Applications Using Extreme Programming Methods," *The Indonesian Journal of Computer Science*, vol. 14, no. 3, Jun. 2025, doi: 10.33022/ijcs.v14i3.4756.
 - [20] A. Akhtar, B. Bakhtawar, and S. Akhtar, "EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS," *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 2, no. 2, p. 2022, 2022, doi: 10.54489/ijtim.v2i1.77.
 - [21] D. Prasetyo, A. Utami, and T. G. Laksana, "Website Based Academic Information System Design Using Extreme Programming Method," *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, vol. 6, no. 2, pp. 134–143, Jul. 2024, doi: 10.20895/inista.v6i2.1214.
 - [22] N. Rahmah, N. Hidayat, and D. K. Wibowo, "Development of a Web-Based Management Information System for Student Creativity Program (PKM) Using Extreme Programming and Laravel Framework," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 4, pp. 2550–2565, Sep. 2025, doi: 10.52436/1.jutif.2025.6.4.5267.
 - [23] F. Said, D. Kristomo, and W. Andriyani, "Application of Extreme Programming Methods in the Design and Building of the Nusantara

- Capital Sentiment Analysis System,” *Sinkron : Jurnal dan Penelitian Teknik Informatika*, vol. 9, no. 2, pp. 578–589, Apr. 2025, doi: 10.33395/sinkron.v9i2.14617.
- [24] Anggara S, “Penerapan Metode Extreme Programming dalam Pengembangan Aplikasi Point of Sale pada Usaha Phi Phi Thai,” *Bridge : Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, vol. 3, no. 3, pp. 197–206, Aug. 2025, doi: 10.62951/bridge.v3i3.613.
- [25] E. Supriyati and M. Azrino Gustalika, “Extreme Programming Method for Integrated Service System Website Development in Rejosari Village,” *Jurnal Riset Informatika*, vol. 5, no. 3, pp. 533–538, Jun. 2023, doi: 10.34288/jri.v5i3.527.
- [26] D. A. Prananda, R. Imanda, T. Anhari, M. K. Sulaeman, and M. A. Rizkiawan, “DEVELOPMENT OF RESPONSIVE AND INTERACTIVE MOBILE-BASED REALISTIC MATHEMATICS LEARNING APPLICATION WITH EXTREME PROGRAMMING SYSTEM METHOD,” *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 1, pp. 576–588, Jan. 2025, doi: 10.29100/jupi.v10i1.5939.
- [27] M. I. Shiddiq, “IMPLEMENTASI WHITE BOX TESTING BERBASIS PATH PADA FORM LOGIN APLIKASI BERBASIS WEB,” *Jurnal Siliwangi*, vol. 8, no. 1, 2022.
- [28] McCabe, “A Complexity Measure,” *IEEE Transactions on Software Engineering*, vol. SE-2 No.4, 1976.
- [29] M. Shandy, R. Langit, A. Voutama, A. A. Ridha, S. Informasi, and U. S. Karawang, “BLACK BOX TESTING PADA WEBSITE SISTEM PERPUSTAKAAN MENGGUNAKAN METODE EQUIVALENCE PARTITIONING DAN ANALISIS BOUNDARY VALUE,” *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 8, 2024.
- [30] A. A. Winarto and R. Herawati, “TESTING ON THE PACKAGE PROGRAM USING THE BLACK BOX TESTING METHOD

WITH THE EQUIVALENCE PARTITIONING TECHNIQUE,”
Proxies, vol. 3, no. 2, p. 63, 2020.

- [31] Dini Nurul Azizah *et al.*, “Analysis and Testing of the Combox Web Application System Using Black Box Testing with the Equivalence Partitioning Method,” *International Journal of Electrical Engineering, Mathematics and Computer Science*, vol. 1, no. 4, pp. 37–43, Nov. 2024, doi: 10.62951/ijeemcs.v1i4.118.

