

**RANCANG BANGUN VISUAL STUDIO CODE
EXTENSION UNTUK MANAJEMEN TEMPLATE
KODE DINAMIS DENGAN INTEGRASI LLM**

TUGAS AKHIR



DISUSUN OLEH:

GIVENDRA DIPANEGERA

NIM. 21106050045

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SUNAN
KALIJAGA**

YOGYAKARTA

2025

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-1246/Un.02/DST/PP.00.9/06/2025

Tugas Akhir dengan judul : "Rancang Bangun Visual Studio Code Extension untuk Manajemen Template Kode Dinamis dengan Integrasi LLM"

yang dipersiapkan dan disusun oleh:

Nama : GIVENDRA DIPANEGARA
Nomor Induk Mahasiswa : 21106050045
Telah diujikan pada : Rabu, 18 Juni 2025
Nilai ujian Tugas Akhir : A-

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

TIM UJIAN TUGAS AKHIR



Ketua Sidang

Ir. Muhammad Didik Rohmad Wahyudi, S.T., MT.
SIGNED

Valid ID: 686223cee75b9



Penguji I

Prof. Dr. Ir. Shofwatul 'Uyun, S.T., M.Kom.,
IPM., ASEAN Eng.
SIGNED

Valid ID: 685f1a927d608



Penguji II

Nurochman, S.Kom., M.Kom
SIGNED

Valid ID: 685dcbc9c7082



Yogyakarta, 18 Juni 2025

UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi

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

Valid ID: 686247f75eb51

LEMBAR PERNYATAAN

SURAT PERNYATAAN KEASLIAN/BEBAS PLAGIASI

Yang bertanda tangan di bawah ini:

Nama : Givendra Dipanegara
NIM : 21106050045
Program Studi : Informatika
Fakultas : Sains dan Teknologi

Dengan ini saya menyatakan bahwa tugas akhir saya yang berjudul "Rancang Bangun *Visual Studio Code Extension* untuk Manajemen Template Kode Dinamis dengan Integrasi *LLM*" benar-benar merupakan hasil karya asli saya sendiri. Saya mengaskan bahwa karya ini belum pernah diajukan dalam bentuk apa pun untuk memperoleh gelar akademik di perguruan tinggi mana pun, serta tidak mengandung unsur plagiasi baik sebagian maupun seluruhnya dari karya orang lain. Segala kutipan atau referensi yang digunakan dalam penulisan tugas akhir ini telah dicantumkan dengan jelas dan sesuai dengan aturan penulisan ilmiah yang berlaku. Apabila di kemudian hari ditemukan pelanggaran atas pernyataan ini, saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 4 Mei 2025


METERAI
TEMPEL
EPAG66AMX239979058

Givendra Dipanegara
NIM. 21106050045

LEMBAR PERSETUJUAN TUGAS AKHIR



Universitas Islam Negeri Sunan Kalijaga FM-



UINSK-BM-05-03/R0

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir
Lamp : -

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

Assalammu'alaikum wr. wb.

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

Nama : Givendra Dipanegara

NIM : 21106050045

Judul Tugas Akhir : Rancang Bangun Visual Studio Code Extension untuk Manajemen Template Kode Dinamis dengan Integrasi LLM

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 mengharap agar skripsi/tugas akhir Saudara dapat segera di-munaqasyah-kan. Atas perhatiannya saya ucapkan terima kasih.

Wassalammu'alaikum wr. wb.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 26 Mei 2025

Pembimbing

Muhammad Didik Rohmad Wahyudi, S.T., MT.

NIP. 19760812 200901 1 015

INTISARI

Kemajuan teknologi dalam lingkungan pengembangan perangkat lunak modern telah mendorong penggunaan template dan *snippet* kode untuk meningkatkan efisiensi pengembangan. Visual Studio Code (VSCode) sebagai salah satu IDE yang paling populer menyediakan fitur *snippet* bawaan, tetapi memiliki keterbatasan dalam hal manajemen template dan berbagi antar proyek. Selain itu, dengan kemajuan teknologi AI dan Large Language Models (LLM), terdapat peluang untuk mengintegrasikan kemampuan pembuatan kode secara otomatis ke dalam proses manajemen template. Tugas akhir ini bertujuan untuk merancang dan membangun ekstensi VSCode yang dapat mengelola template kode secara dinamis dengan integrasi AI. Ekstensi ini diimplementasikan dengan fitur pengelolaan template yang fleksibel, integrasi dengan penyedia AI seperti Groq, Google AI Studio, Openai, dan Ollama, serta kemampuan untuk berbagi template antar proyek. Hasil implementasi menunjukkan bahwa ekstensi ini berhasil menyederhanakan proses manajemen template, meningkatkan produktivitas pengembang, dan memfasilitasi kolaborasi tim dalam pengembangan perangkat lunak.

Kata kunci: Visual Studio Code, Template Kode, Ekstensi, AI, Model Bahasa Besar, Pengembangan Perangkat Lunak



ABSTRACT

Technological advancements in modern software development environments have driven the use of code templates and snippets to improve development efficiency. Visual Studio Code (VSCode) as one of the most popular IDEs provides built-in snippet features, but has limitations in terms of template management and sharing across projects. Additionally, with the advancement of AI technology and Large Language Models (LLM), there is an opportunity to integrate automatic code generation capabilities into the template management process. This final project aims to design and build a VSCode extension that can manage code templates dynamically with AI integration. This extension is implemented with flexible template management features, integration with AI providers such as Groq, Google AI Studio, Openai, and Ollama, as well as the ability to share templates across projects. The implementation results show that the extension successfully simplifies the template management process, increases developer productivity, and facilitates team collaboration in software development.

Keywords: Visual Studio Code, Code Template, Extension, AI, Large Language Models, Software Development



HALAMAN PERSEMBAHAN

“Tugas akhir ini saya persembahkan kepada kedua orang tua saya yang telah memberikan dukungan penuh, baik secara emosional maupun finansial, selama proses pendidikan saya. Terima kasih atas doa, kerja keras, dan pengorbanan yang tidak pernah berhenti demi masa depan saya. Segala pencapaian ini tidak akan mungkin terwujud tanpa ketulusan dan kasih sayang yang terus mengalir dari kalian sejak awal hingga akhir perjalanan pendidikan ini.”



KATA PENGANTAR

Alhamdulillah Rabbil 'Alamin, puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "Rancang Bangun Visual Studio Code Extension untuk Manajemen Template Kode Dinamis dengan Integrasi LLM". Shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW beserta keluarga dan para sahabatnya.

Dalam penyelesaian tugas akhir ini, penulis telah banyak mendapat bantuan, bimbingan, dan dukungan dari berbagai pihak. Untuk itu dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih kepada:

1. Bapak Prof. Noorhaidi Hasan, S.Ag., M.Phil., Ph.D selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Ibu 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. Bapak Nurochman, S.KOM., M.KOM. yang telah memberikan arahan dan bimbingan akademik selama masa perkuliahan.
5. Bapak Muhammad Didik Rohmad Wahyudi, S.T., MT. yang telah membimbing dengan penuh kesabaran dan memberikan arahan berharga dalam penyelesaian tugas akhir ini.
6. Segenap dosen Program Studi Informatika yang telah memberikan ilmu yang bermanfaat serta seluruh staf dan karyawan Fakultas Sains dan Teknologi UIN Sunan Kalijaga.
7. Kedua orang tua tercinta dan keluarga yang senantiasa memberikan dukungan, doa, dan motivasi tiada henti.
8. Rekan-rekan mahasiswa Informatika angkatan 2021 yang telah berjuang bersama dan saling mendukung selama masa perkuliahan.
9. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam penyelesaian tugas akhir ini.

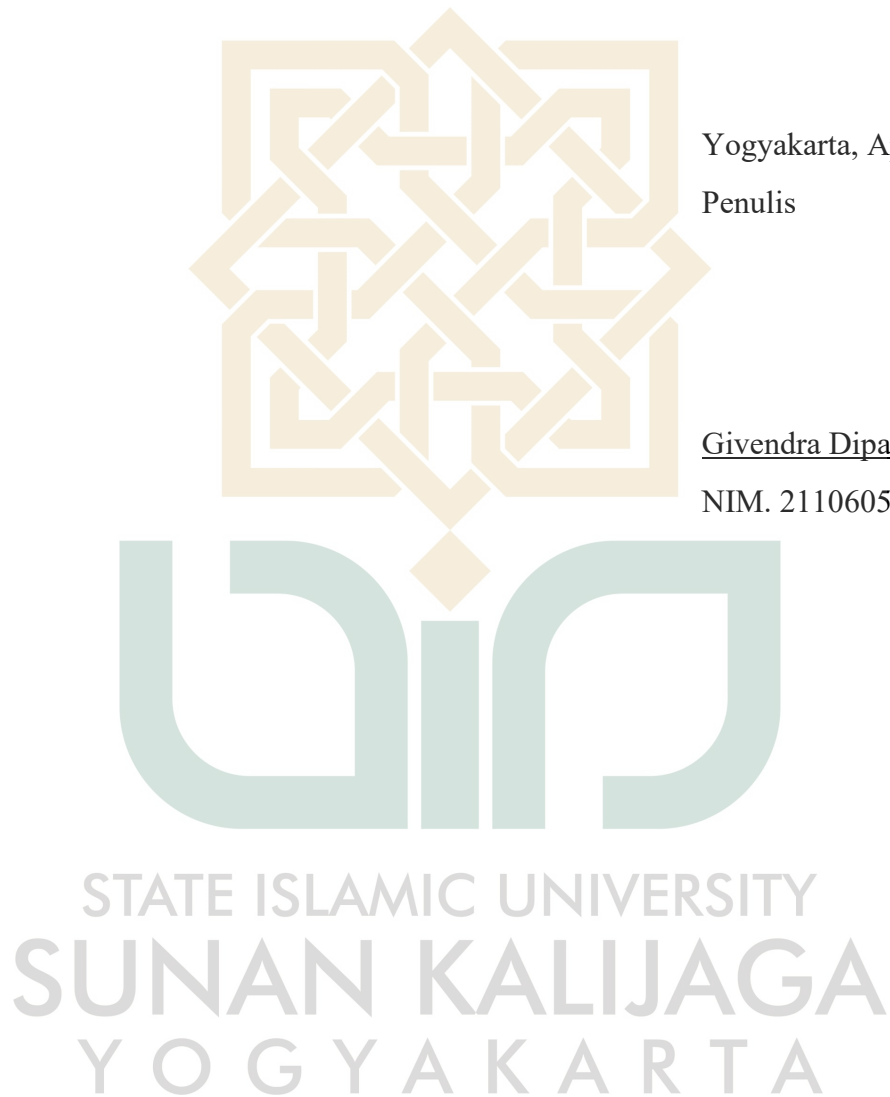
Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Akhir kata, semoga tugas akhir ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Yogyakarta, April 2025

Penulis

Givendra Dipanegara

NIM. 21106050045



DAFTAR ISI

LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
LEMBAR PERSETUJUAN TUGAS AKHIR.....	iv
INTISARI.....	v
ABSTRACT.....	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah	5
1.4 Tujuan.....	5
1.5 Manfaat.....	6
BAB II KAJIAN PUSTAKA	7
2.1 Tinjauan Pustaka.....	7
2.1.1 Visual Studio Code, Sistem Ekstensi, dan Manajemen <i>Template</i>	7
2.1.2 Penggunaan AI dalam Alat Pengembangan Perangkat Lunak	8
2.1.3 <i>Large Language Models</i> (LLM) untuk Generasi Kode.....	9
2.1.4 Keamanan dan Privasi dalam Alat Pengembangan Berbasis AI	10
2.2 Landasan Teori	11
2.2.1 Visual Studio Code.....	11
2.2.2 VScode Extension Development.....	12
2.2.3 Github Gist	20
2.2.4 Large Language Models.....	21
2.2.5 Mocha.....	27
BAB III METODE PENGEMBANGAN SISTEM	29
3.1 Alat dan Bahan	29

3.1.1	Perangkat Keras (<i>Hardware</i>).....	29
3.1.2	Perangkat Lunak (<i>Software</i>).....	29
3.2	Metode dan Tahapan Pengembangan	31
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM.....		33
4.1	Perencanaan Sistem	33
4.1.1	Analisis Kebutuhan Fungsional.....	33
4.1.2	Analisis Kebutuhan Non-Fungsional	38
4.2	Perancangan Sistem.....	42
4.2.1	Arsitektur Sistem.....	43
4.2.2	Desain Penyimpanan Data.....	46
4.2.3	Desain Antarmuka (UI Components).....	49
4.2.4	Desain Sistem Perintah (Command System).....	54
4.3	Implementasi Sistem.....	57
4.3.1	Implementasi Extension Host.....	58
4.3.2	Implementasi Template Manager.....	59
4.3.3	Implementasi AI Integration	60
4.3.4	Implementasi Storage Service	61
4.3.5	Implementasi UI Components.....	62
4.3.6	Implementasi Command System.....	63
4.3.7	Implementasi Github Gist	64
4.3.8	Implementasi Utility.....	65
4.4	Pengujian Sistem	65
4.4.1	Unit Testing.....	65
4.4.2	Black Box Testing.....	82
4.4.3	Panduan Pengguna	96
BAB V PENUTUP.....		98
5.1	Kesimpulan.....	98
5.2	Saran	98
DAFTAR PUSTAKA		100
LAMPIRAN.....		103

DAFTAR GAMBAR

Gambar 2.1	Logo Visual Studio Code.....	11
Gambar 2.2	Tampilan UI Github Gist	20
Gambar 2.3	Logo Groq.....	22
Gambar 2.4	Logo Google AI Studio.....	23
Gambar 2.5	Logo Openai.....	23
Gambar 2.6	Logo Ollama	24
Gambar 2.7	Logo Mocha.....	27
Gambar 3.1	Diagram Tahap Pengembangan VSCode Extension.....	32
Gambar 4.1	Diagram Arsitektur Sistem VSCode Extension	43
Gambar 4.2	Struktur Data Template.....	46
Gambar 4.3	Isi Konfigurasi File .repoignore	48
Gambar 4.4	UI untuk Membuat Template Baru.....	49
Gambar 4.5	UI untuk Mengedit Template.....	50
Gambar 4.6	UI untuk Pratinjau Isi Template.....	51
Gambar 4.7	UI untuk Generative AI	52
Gambar 4.8	Contoh Prompt dengan Respon LLM	52
Gambar 4.9	UI untuk Pratinjau Gist	53
Gambar 4.10	Perintah di Command Palette.....	54
Gambar 4.11	Kode Implementasi Extension Host.....	58
Gambar 4.12	Kode Implementasi readTemplates di Template Manager	59
Gambar 4.13	Kode Implementasi Provider AI Groq.....	60
Gambar 4.14	Penerapan Secret Storage untuk Menyimpan Token Github	61
Gambar 4.15	Kode Implementasi Editor Template	62
Gambar 4.16	Kode Implementasi Sistem Perintah	63
Gambar 4.17	Implementasi Github Gist Menggunakan Octokit.js.....	64
Gambar 4.18	Implementasi Regex untuk Konfigurasi File .repoignore	65
Gambar 4.19	Pengujian Inti Ekstensi	66
Gambar 4.20	Pengujian Utilitas Ekstensi	67
Gambar 4.21	Pengujian Sistem Perintah	68
Gambar 4.22	Pengujian Fitur Pelengkapan Otomatis pada Template	69
Gambar 4.23	Pengujian Fitur Penambahan Template Baru.....	70
Gambar 4.24	Pengujian Fitur Penyisipan Template	70

Gambar 4.25 Pengujian Fitur Edit Template	71
Gambar 4.26 Pengujian Fitur Penghapusan Template.....	72
Gambar 4.27 Pengujian Fitur Template Favorit	72
Gambar 4.28 Pengujian Fitur Generative AI	73
Gambar 4.29 Pengujian Fitur Berbagi Template	74
Gambar 4.30 Pengujian Integrasi dengan Github Gist	74
Gambar 4.31 Pengujian Fitur Penyalinan Direktori Proyek	75
Gambar 4.32 Pengujian Fitur Pratinjau Gist.....	76
Gambar 4.33 Pengujian Fitur Editor Template.....	76
Gambar 4.34 Pengujian Fitur Pratinjau Template	77



DAFTAR TABEL

Tabel 2.1	Perbandingan Model antar Provider.....	27
Tabel 3.1	Software yang Digunakan Selama Pengembangan Sistem	31
Tabel 4.1	Unit Testing Pada Fitur Extension	82
Tabel 4.2	Pengujian Black Box untuk Kebutuhan Fungsional	93
Tabel 4.3	Pengujian Black Box untuk Kebutuhan Non-Fungsional	96



DAFTAR LAMPIRAN

Lampiran 1 Link Repository Github.....	103
--	-----



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era modern yang ditandai dengan perkembangan teknologi yang pesat, pengembangan perangkat lunak menjadi aspek yang sangat penting dalam mendukung berbagai aktivitas bisnis, pendidikan, dan penelitian. Seiring dengan itu, kompleksitas proyek perangkat lunak juga meningkat, menuntut efisiensi dan produktivitas yang lebih tinggi dari para pengembang. Salah satu alat bantu yang esensial dalam ekosistem pengembangan modern adalah Integrated Development Environment (IDE) atau text editor. Visual Studio Code (VSCode), software yang dikembangkan oleh Microsoft, telah muncul sebagai salah satu text editor yang paling populer dan banyak digunakan oleh komunitas pengembang global. Cara kerjanya didasarkan pada arsitektur yang ringan namun kuat, yang memungkinkannya untuk menjadi sangat responsif dan dapat diperluas. VSCode bekerja dengan membuka folder atau workspace proyek, menganalisis struktur file dan, melalui berbagai parser dan layanan bahasa, menyediakan fitur-fitur seperti syntax highlighting, IntelliSense (code completion), debugging, dan integrasi Git. Popularitasnya didorong oleh fleksibilitas, kustomisasi yang luas melalui ekstensi, dan dukungan lintas platform. Salah satu kekuatan utama VSCode adalah model ekstensibilitasnya, yang memungkinkan pengembang untuk menambahkan fungsionalitas baru melalui API ekstensi.

Sejalan dengan kemajuan IDE, teknologi Kecerdasan Buatan (Artificial Intelligence AI) juga mengalami perkembangan yang sangat pesat dan mulai

merambah berbagai aspek dalam siklus hidup pengembangan perangkat lunak. Saat ini, teknologi AI, khususnya Large Language Model (LLM), telah menunjukkan kemampuan yang luar biasa dalam pemahaman dan generasi kode, prediksi kesalahan, hingga otomatisasi tugas-tugas repetitif. Salah satu contoh nyata adalah alat bantu seperti GitHub Copilot yang mampu memberikan sugesti kode secara cerdas dan kontekstual, secara signifikan mempercepat proses coding.

Dalam proses pengembangan perangkat lunak, pengembang seringkali dihadapkan pada kebutuhan untuk menulis ulang struktur kode yang serupa atau menggunakan pola desain tertentu berulang kali. Untuk mengatasi hal ini dan meningkatkan efisiensi, penggunaan template atau snippet kode menjadi solusi yang umum. Secara umum, template dalam konteks pemrograman adalah kerangka atau cetak biru pra-desain dari sebuah atau potongan kode, dokumen, atau struktur data yang dapat digunakan berulang kali. Template berisi bagian-bagian yang tetap (boilerplate) dan bagian-bagian yang dapat diubah (placeholder) yang diisi sesuai dengan kebutuhan spesifik pada saat penggunaan. Tujuan utama dari template adalah untuk menyediakan struktur standar dan mengurangi pekerjaan repetitif.

VSCode, dalam konteks pengembangan perangkat lunak, membutuhkan template kode karena beberapa alasan krusial yang berkaitan langsung dengan kebutuhan efisiensi dan kualitas. Template memastikan bahwa blok kode yang sering digunakan memiliki struktur dan format yang konsisten di seluruh proyek dan tim, yang sangat penting untuk kolaborasi dan pemeliharaan kode jangka panjang. Penggunaan template secara signifikan mengurangi waktu yang dibutuhkan untuk menulis kode dari awal, memungkinkan pengembang untuk lebih

fokus pada logika bisnis yang lebih kompleks daripada tugas-tugas pengetikan rutin. Dengan menggunakan template yang sudah teruji, risiko kesalahan ketik atau logika minor dalam kode boilerplate dapat diminimalisir, sehingga meningkatkan kualitas kode secara keseluruhan. Lebih lanjut, dalam tim pengembangan, template membantu dalam menerapkan standar coding dan praktik terbaik, serta dapat menjadi alat pembelajaran yang efektif bagi pengembang baru untuk memahami bagaimana struktur kode tertentu seharusnya ditulis. Mengingat VSCode digunakan untuk berbagai bahasa pemrograman dan framework, template yang dapat disesuaikan untuk masing-masing teknologi menyediakan potongan kode yang relevan dan siap pakai, mempercepat proses adaptasi dan pengembangan.

Visual Studio Code sendiri menyediakan fitur snippet bawaan, namun fungsionalitasnya terkadang memiliki keterbatasan, terutama dalam hal manajemen template yang lebih kompleks, dinamis, dan kemudahan berbagi antar proyek atau tim. Meskipun demikian, alat manajemen template yang ada saat ini umumnya masih belum terintegrasi secara mendalam dengan kapabilitas AI generatif. Hal ini membatasi kemampuan sistem dalam menghasilkan dan menyesuaikan template secara dinamis berdasarkan konteks dan kebutuhan spesifik pengembang secara lebih cerdas.

Kondisi ini menciptakan peluang untuk mengembangkan sebuah sistem manajemen template yang lebih canggih, yang tidak hanya bersifat dinamis dan fleksibel tetapi juga memanfaatkan kemampuan AI untuk memberikan solusi yang lebih adaptif dan cerdas. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sebuah extension untuk Visual Studio Code yang berfokus pada

manajemen code snippet atau template kode secara dinamis dengan integrasi AI. Sistem ini diharapkan dapat diintegrasikan secara langsung dengan VSCode, menyediakan antarmuka yang ramah bagi pengguna untuk membuat, menyimpan, mencari, dan menggunakan template kode. Untuk mendukung kolaborasi dan kemudahan berbagi template antar pengembang maupun antar proyek, sistem ini juga akan dilengkapi dengan integrasi ke layanan populer seperti GitHub Gist, memfasilitasi penyimpanan dan sinkronisasi template secara online. Lebih lanjut, dengan integrasi AI, ekstensi ini diharapkan mampu memberikan saran template yang lebih relevan, atau bahkan membantu dalam generasi template baru berdasarkan deskripsi kebutuhan pengguna. Dengan menyediakan solusi yang beragam, sistem ini bertujuan untuk menjadi alat bantu yang andal bagi pengembang dalam berbagai skenario pengembangan perangkat lunak, meningkatkan produktivitas, dan konsistensi kode.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana cara merancang dan mengimplementasikan sistem manajemen template yang fleksibel dan mudah digunakan dalam lingkungan VSCode?
2. Bagaimana mengintegrasikan kemampuan generasi kode berbasis AI ke dalam sistem manajemen template secara efektif dan aman?
3. Bagaimana memfasilitasi fitur berbagi dan sinkronisasi template antar proyek dan tim dengan menjaga konsistensi?

1.3 Batasan Masalah

Untuk memfokuskan pengembangan sistem, batasan masalah ditetapkan bahwa ekstensi yang dikembangkan hanya akan berjalan pada platform Visual Studio Code sebagai lingkungan pengembangan perangkat lunak.

1.4 Tujuan

Penelitian ini bertujuan untuk mengembangkan sebuah ekstensi Visual Studio Code yang dapat meningkatkan produktivitas pengembang melalui sistem manajemen template kode yang dinamis dan terintegrasi dengan kecerdasan buatan. Dari sisi pengguna VSCode, sistem ini diharapkan dapat memberikan fleksibilitas dalam pembuatan, penyimpanan, pencarian, serta penggunaan template kode secara cepat dan intuitif. Selain itu, pengguna akan dibantu oleh fitur AI yang mampu memberikan saran template yang relevan berdasarkan konteks pekerjaan mereka, sehingga dapat mengurangi beban kognitif dan mempercepat proses pengembangan kode. Fitur berbagi template juga menjadi fokus utama agar standarisasi kode dan praktik terbaik dapat diterapkan secara konsisten dalam tim pengembang.

Dari perspektif pengembangan ekosistem VSCode, ekstensi ini diharapkan dapat memberikan kontribusi nyata dengan menghadirkan alat bantu baru yang fungsional dan bermanfaat. Integrasi teknologi AI ke dalam proses manajemen dan generasi template diharapkan menunjukkan potensi adopsi AI dalam pengembangan perangkat lunak sehari-hari secara lebih dalam. Tak hanya itu, ekstensi ini juga dirancang sebagai fondasi atau kerangka awal yang memungkinkan pengembangan fitur-fitur lanjutan di masa depan, khususnya dalam konteks manajemen template kode yang lebih canggih dan adaptif.

Secara keseluruhan, tujuan utama dari penelitian ini adalah menghasilkan sebuah ekstensi Visual Studio Code yang tidak hanya mudah digunakan dan praktis, tetapi juga mampu memberikan dampak nyata dalam meningkatkan efisiensi kerja dan kualitas kode pengembang melalui pemanfaatan manajemen template yang cerdas dan terotomatisasi.

1.5 Manfaat

1. Memberikan solusi praktis bagi pengembang dalam mengelola dan menyimpan template kode.
2. Meningkatkan efisiensi kerja pengembang dengan meminimalkan waktu yang dihabiskan untuk tugas berulang.
3. Berkontribusi pada pengembangan ekosistem Visual Studio Code.

BAB V

PENUTUP

5.1 Kesimpulan

Pengembangan ekstensi Visual Studio Code untuk manajemen template kode dinamis dengan integrasi LLM telah berhasil diselesaikan dengan baik. Ekstensi ini menyediakan sistem manajemen template yang fleksibel dan mudah digunakan, memungkinkan pengguna untuk membuat, mengedit, dan mengorganisir template kode secara efisien. Integrasi dengan berbagai penyedia LLM seperti Groq, Google AI Studio, OpenAI, dan Ollama telah diimplementasikan dengan sukses, memberikan kemampuan generatif yang dapat meningkatkan produktivitas pengembang. Fitur kolaborasi juga menjadi salah satu keunggulan, dengan adanya dukungan ekspor-impor template serta integrasi dengan GitHub Gist yang memudahkan pengguna dalam berbagi dan mengakses template secara lintas proyek. Selain fungsionalitas yang kuat, antarmuka pengguna dirancang secara intuitif dengan memperhatikan aspek pengalaman pengguna (UX), menjadikan ekstensi ini mudah dioperasikan bahkan oleh pengguna yang baru pertama kali menggunakannya.

5.2 Saran

Terdapat beberapa saran untuk pengembangan sistem di masa mendatang, diantaranya:

1. Peningkatan Manajemen Tag: Mengembangkan fungsionalitas manajemen *tag* yang lebih lanjut, misalnya dengan memperkaya sistem *tagging* melalui

fitur seperti *nested tags* (tag bersarang) atau saran *tag* otomatis berdasarkan konten *template*.

2. Perluasan Dukungan Model LLM: Mengintegrasikan model dari *provider* yang lebih beragam seperti DeepSeek, Qwen, Mistral AI, dan lainnya untuk memberikan pilihan yang lebih luas kepada pengguna.
3. Pengembangan Fitur Percakapan LLM yang Lebih Interaktif: Meningkatkan interaksi dengan LLM dengan menambahkan dukungan untuk riwayat percakapan (*conversational history*). Fitur ini akan memungkinkan pengguna untuk mengirimkan serangkaian *prompt* secara berurutan dalam satu sesi, menciptakan alur interaksi yang lebih natural dan kontekstual seperti yang biasa ditemukan pada *website* LLM pada umumnya. Hal ini akan mempermudah pengguna dalam mengembangkan atau menyempurnakan kode secara iteratif dengan bantuan AI.

DAFTAR PUSTAKA

- [1] “Technology | 2024 Stack Overflow Developer Survey.” Accessed: May 18, 2025. [Online]. Available: <https://survey.stackoverflow.co/2024/technology#most-popular-technologies-new-collab-tools>
- [2] “Extension API.” Accessed: May 18, 2025. [Online]. Available: <https://code.visualstudio.com/api/index>
- [3] “Snippets in Visual Studio Code.” Accessed: May 18, 2025. [Online]. Available: <https://code.visualstudio.com/docs/editing/userdefinedsnippets>
- [4] “Project Templates - Visual Studio Marketplace.” Accessed: May 18, 2025. [Online]. Available: <https://marketplace.visualstudio.com/items?itemName=cantonios.project-templates>
- [5] “JavaScript (ES6) code snippets - Visual Studio Marketplace.” Accessed: May 18, 2025. [Online]. Available: <https://marketplace.visualstudio.com/items?itemName=xabikos.JavaScript Snippets>
- [6] U. K. Durrani, M. Akpınar, M. Fatih Adak, A. Talha Kabakus, M. Maruf Öztürk, and M. Saleh, “A Decade of Progress: A Systematic Literature Review on the Integration of AI in Software Engineering Phases and Activities (2013-2023),” *IEEE Access*, vol. 12, pp. 171185–171204, 2024, doi: 10.1109/ACCESS.2024.3488904.
- [7] E. Kalliamvakou, “Research: quantifying GitHub Copilot’s impact on developer productivity and happiness,” The GitHub Blog. Accessed: May 18, 2025. [Online]. Available: <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>
- [8] Z. Zhang *et al.*, “Unifying the Perspectives of NLP and Software Engineering: A Survey on Language Models for Code,” Jun. 26, 2024, *arXiv*: arXiv:2311.07989. doi: 10.48550/arXiv.2311.07989.

- [9] J. Huang *et al.*, “Large Language Models Can Self-Improve,” Oct. 25, 2022, *arXiv*: arXiv:2210.11610. doi: 10.48550/arXiv.2210.11610.
- [10] “Gemini 2.5: Our most intelligent AI model,” Google. Accessed: May 18, 2025. [Online]. Available: <https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/>
- [11] M. Chen *et al.*, “Evaluating Large Language Models Trained on Code,” Jul. 14, 2021, *arXiv*: arXiv:2107.03374. doi: 10.48550/arXiv.2107.03374.
- [12] J. Austin *et al.*, “Program Synthesis with Large Language Models,” Aug. 16, 2021, *arXiv*: arXiv:2108.07732. doi: 10.48550/arXiv.2108.07732.
- [13] E. Lin, I. Koishybayev, T. Dunlap, W. Enck, and A. Kapravelos, “UntrustIDE: Exploiting Weaknesses in VS Code Extensions,” in *Proceedings 2024 Network and Distributed System Security Symposium*, San Diego, CA, USA: Internet Society, 2024. doi: 10.14722/ndss.2024.24073.
- [14] Y. Fu *et al.*, “Security Weaknesses of Copilot-Generated Code in GitHub Projects: An Empirical Study,” Feb. 06, 2025, *arXiv*: arXiv:2310.02059. doi: 10.48550/arXiv.2310.02059.
- [15] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, “Not what you’ve signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection,” May 05, 2023, *arXiv*: arXiv:2302.12173. doi: 10.48550/arXiv.2302.12173.
- [16] K. Madampe, J. Grundy, and N. Arachchilage, “AI-based Programming Assistants for Privacy-related Code Generation: The Developers’ Experience,” Mar. 06, 2025, *arXiv*: arXiv:2503.03988. doi: 10.48550/arXiv.2503.03988.
- [17] “Webview API.” Accessed: May 21, 2025. [Online]. Available: <https://code.visualstudio.com/api/extension-guides/webview>
- [18] “What is a VS Code workspace?” Accessed: May 21, 2025. [Online]. Available: <https://code.visualstudio.com/docs/editing/workspaces/workspaces>

- [19] “User interface.” Accessed: May 21, 2025. [Online]. Available: https://code.visualstudio.com/docs/getstarted/userinterface#_command-palette
- [20] “About Groq - Fast AI Inference.” Accessed: May 21, 2025. [Online]. Available: <https://groq.com/about-us/>
- [21] “Google AI Studio | Gemini API,” Google AI for Developers. Accessed: May 21, 2025. [Online]. Available: <https://ai.google.dev/aistudio>
- [22] “Run Code Llama locally · Ollama Blog.” Accessed: May 21, 2025. [Online]. Available: <https://ollama.com/public/Run Code Llama locally>
- [23] “Mocha - the fun, simple, flexible JavaScript test framework.” Accessed: May 21, 2025. [Online]. Available: <https://mochajs.org/>
- [24] “VS Code API.” Accessed: May 21, 2025. [Online]. Available: <https://code.visualstudio.com/api/references/vscode-api#SecretStorage>

