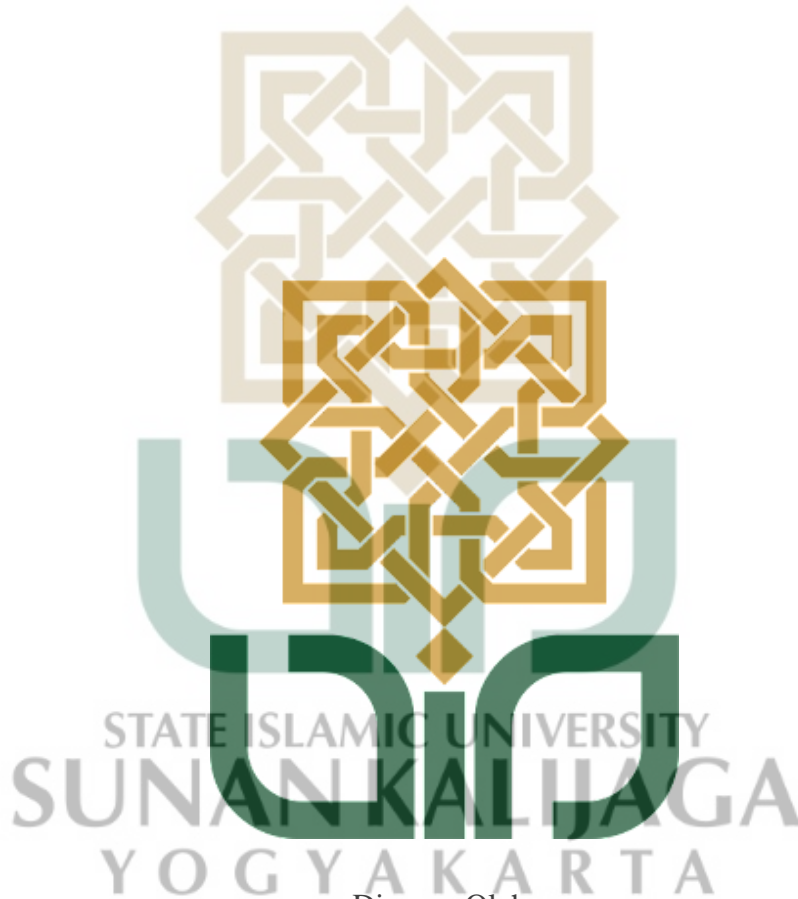


**Rancang Bangun Sistem Otomatisasi Pembuatan README.md  
pada Repository GitHub Menggunakan Teknologi  
*Large Language Model***

**TUGAS AKHIR**

Untuk Memenuhi Sebagian Persyaratan Mencapai Derajat Sarjana S-1  
Program Studi Informatika



Disusun Oleh:

Muhammad Zhafran Hana Alfarossi

NIM. 22106050061

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

**2026**

# LEMBAR PENGESAHAN TUGAS AKHIR



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-1270/Un.02/DST/PP.00.9/06/2026

Tugas Akhir dengan judul : Rancang Bangun Sistem Otomatisasi Pembuatan README.md pada Repository GitHub Menggunakan Teknologi Large Language Model

yang dipersiapkan dan disusun oleh:

Nama : MUHAMMAD ZHAFRAN HANA ALFAROSI  
Nomor Induk Mahasiswa : 22106050061  
Telah diujikan pada : Selasa, 02 Juni 2026  
Nilai ujian Tugas Akhir : A-

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

### TIM UJIAN TUGAS AKHIR



Ketua Sidang  
Muhammad Galih Wonoseto, M.T.  
SIGNED

Valid ID: 6a278ef11840



Penguji I  
Dr. Ir. Aulia Faqih Rifai, M.Kom.  
SIGNED

Valid ID: 6a266f820227d



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

Valid ID: 6a27875b82542



Yogyakarta, 02 Juni 2026  
UIN Sunan Kalijaga  
Dekan Fakultas Sains dan Teknologi  
Prof. Dr. Dra. Hj. Khurul Wardati, M.St.  
SIGNED

Valid ID: 6a27a5d860df

## LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Muhammad Zhafran Hana Alfarossi

NIM : 22106050061

Program Studi : Informatika

Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya, bahwa tugas akhir saya yang berjudul **“Rancang Bangun Sistem Otomatisasi Pembuatan README.md pada Repository GitHub Menggunakan Teknologi *Large Language Model*”** merupakan hasil tugas akhir saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan bukan plagiasi karya orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Sebagai salah satu syarat untuk memperoleh gelar Sarjana Program Studi Informatika pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga, Yogyakarta.

Yogyakarta, 19 Mei 2026

Penulis,



Muhammad Zhafran Hana Alfarossi

NIM. 22106050061

## LEMBAR PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi/Tugas Akhir

Lamp : -

Kepada

Yth.

Dekan Fakultas Sains dan Teknologi  
Universitas Islam Negeri Sunan Kalijaga  
DI Yogyakarta

Assalamu'alaikum wr. wb

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

Nama : Muhammad Zhafran Hana Alfarossi

NIM : 22106050061

Judul Skripsi : Rancang Bangun Sistem Otomatisasi Pembuatan README.md pada Repository GitHub Menggunakan Teknologi *Large Language Model*

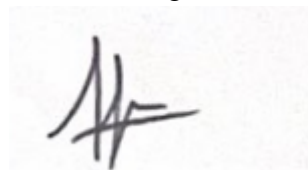
Sudah dapat diajukan kepada Program Studi Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini kami berharap agar skripsi/tugas akhir Saudari dapat segera dimunaqosyahkan. Atas perhatiannya saya ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 19 Mei 2026

Pembimbing,



Muhammad Galih Wonoseto, M.T.

NIP.19901113 201903 1 012

## **LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR**

Tugas Akhir ini tidak dipublikasikan, tetapi tersedia di perpustakaan dalam lingkungan Universitas Islam Negeri Sunan Kalijaga Yogyakarta, diperkenankan dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun, dan harus menyebutkan sumbernya sesuai dengan kebiasaan ilmiah. Dokumen Tugas Akhir ini merupakan hak milik Universitas Islam Negeri Sunan Kalijaga Yogyakarta



## ABSTRAK

README.md merupakan komponen dokumentasi penting dalam repositori GitHub yang berperan dalam meningkatkan keterbacaan dan popularitas proyek perangkat lunak. Namun, hasil survei terhadap 50 responden menunjukkan bahwa hanya 12% responden yang selalu membuat README.md, dengan hambatan utama berupa keterbatasan waktu, fokus pada pengembangan fitur, dan kurangnya pemahaman mengenai format dokumentasi yang baik. Penelitian ini bertujuan merancang dan membangun sistem berbasis web untuk mengotomatisasi pembuatan README.md pada repositori GitHub publik menggunakan teknologi *Large Language Model* (LLM), serta mengevaluasi tingkat penerimaan pengguna menggunakan kerangka *AI Acceptance Model* (AIM). Sistem dikembangkan menggunakan metode *Prototyping* secara iteratif dalam dua siklus. Pendekatan *Structural Contextualization* diterapkan melalui pemetaan struktur direktori dan ekstraksi metadata konfigurasi penting menggunakan GitHub REST API untuk memitigasi risiko halusinasi AI. Proses generasi dokumentasi dilakukan menggunakan Google Gemini 2.5 Flash dengan teknik prompt engineering. Evaluasi dilakukan melalui dua tahap *User Acceptance Testing* (UAT) dengan total 77 responden. Hasil penelitian menunjukkan bahwa sistem berhasil dibangun dan mampu menghasilkan README.md secara otomatis dengan lebih efisien. UAT iterasi kedua menunjukkan peningkatan pada seluruh dimensi AIM dengan nilai Cronbach's Alpha sebesar 0,8205. Penelitian ini menawarkan arsitektur single-agent berbasis *Context-Augmented Generation* (CAG) yang lebih ringan tanpa memerlukan basis data vektor maupun infrastruktur agen kompleks.

**Kata kunci:** README.md, GitHub, Large Language Model, otomatisasi dokumentasi, AI Acceptance Model, Context-Augmented Generation.

## ABSTRACT

README.md is an important documentation component in GitHub repositories that contributes to improving the readability and popularity of software projects. However, a survey involving 50 respondents revealed that only 12% consistently create README.md files, with the main obstacles including time limitations, focus on feature development, and lack of understanding regarding proper documentation formats. This study aims to design and develop a web-based system to automate README.md generation for public GitHub repositories using *Large Language Model* (LLM) technology, as well as evaluate user acceptance using the AI Acceptance Model (AIM) framework. The system was developed using an iterative *Prototyping* method in two cycles. A *Structural Contextualization* approach was implemented through directory structure mapping and extraction of essential configuration metadata using the GitHub REST API to mitigate AI hallucination risks. The documentation generation process utilized Google Gemini 2.5 Flash with prompt engineering techniques. Evaluation was conducted through two stages of User Acceptance Testing (UAT) involving a total of 77 respondents. The results show that the system was successfully developed and capable of generating README.md files automatically and more efficiently. The second UAT iteration demonstrated improvements across all AIM dimensions, with a Cronbach's Alpha value of 0.8205. This study proposes a lightweight single-agent architecture based on *Context-Augmented Generation* (CAG) without requiring vector databases or complex agent infrastructures.

**Keywords:** README.md, GitHub, Large Language Model, AI Acceptance Model, Context-Augmented Generation.

## MOTTO

*“Repetisi, repetisi, konsistensi, habit”*

- Dr. Davin Silalahi –

“Tidaklah seorang hamba meninggalkan sesuatu karena Allah, melainkan Allah akan menggantinya dengan yang lebih baik”

- (HR. Ahmad) -

*“Exhale depression as the wind blows.”*

- Jarad Anthony Higgins -

*“Salahmu, Lanang”*

- Dr. Davin Silalahi -



STATE ISLAMIC UNIVERSITY  
SUNAN KALIJAGA  
YOGYAKARTA

**HALAMAN PERSEMBAHAN**

Dengan rasa syukur yang mendalam kepada Allah Yang Maha Pengasih lagi Maha Penyayang, saya mempersembahkan Tugas Akhir ini sebagai wujud penghormatan dan ungkapan kasih yang tulus. Semoga karya ini memberikan manfaat serta bernilai sebagai amal kebaikan. Tugas Akhir ini saya dedikasikan dengan penuh cinta dan rasa hormat kepada:

1. Ibu tercinta, Ernanda Nur Hawa, yang senantiasa memberikan kasih sayang, doa, dukungan, serta pengorbanan yang tiada henti dalam setiap langkah hidup saya.
2. Bapak tercinta, Abdi Rifan Husna, yang selalu menjadi teladan, memberikan semangat, serta dukungan moral dan materiil demi keberhasilan saya.
3. Kakak saya, Maitsa Zharfa Hana, yang selalu memberikan dukungan, kebersamaan, dan motivasi dalam setiap proses yang saya jalani.
4. Sahabat dan teman seperjuangan, yang telah menemani, membantu, serta memberikan semangat dan kebersamaan selama masa perkuliahan hingga penyusunan Tugas Akhir ini.

Sahabat dan teman seperjuangan, yang telah menemani, membantu, serta memberikan semangat dan kebersamaan selama masa perkuliahan hingga penyusunan Tugas Akhir ini.

STATE ISLAMIC UNIVERSITY  
SUNAN KALIJAGA  
YOGYAKARTA

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas segala rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Rancang Bangun Sistem Otomatisasi Pembuatan README.md pada Repository GitHub Menggunakan Teknologi *Large Language Model*” 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. Muhammad Didik Rohmad Wahyudi, S.T., MT. selaku dosen penasihat akademik yang telah memberikan arahan dan bimbingan selama masa perkuliahan penulis.
5. Muhammad Galih Wonoseto, M.T. selaku dosen pembimbing yang telah memberikan bimbingan, masukan yang berharga, serta dukungan penuh dengan penuh kesabaran dalam setiap tahapan 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, Abdi Rifan Husna dan Ernanda Nur Hawa, yang senantiasa memberikan doa, kasih sayang, dukungan, serta pengorbanan yang tiada henti demi keberhasilan penulis.
8. Kakak penulis, Maitsa Zharfa Hana, 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.

STATE ISLAMIC UNIVERSITY  
SUNAN KALIJAGA  
YOGYAKARTA

## DAFTAR ISI

LEMBAR PENGESAHAN TUGAS AKHIR .....	i
LEMBAR PERNYATAAN KEASLIAN .....	ii
LEMBAR PERSETUJUAN SKRIPSI/TUGAS AKHIR.....	iii
LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR.....	iv
ABSTRAK .....	v
ABSTRACT .....	vi
MOTTO.....	vii
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xv
DAFTAR GAMBAR .....	xvi
BAB I.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	5
1.5.1 Manfaat Praktis .....	5
1.5.2 Manfaat Teoritis Akademis.....	5
BAB II.....	6
2.1 Tinjauan Pustaka .....	6
2.2 Landasan Teori.....	9
2.2.1 Dokumentasi Perangkat Lunak .....	9
2.2.1.1 README.md.....	10
2.2.2.2 Markdown .....	11
2.2.3 GitHub dan GitHub API.....	11

2.2.4	<i>Large Language Model (LLM)</i> .....	12
2.2.5	Arsitektur Generative AI.....	13
2.2.5.1	<i>Retrieval-Augmented Generation (RAG)</i> .....	14
2.2.5.2	<i>Context-Augmented Generation (CAG)</i> .....	15
2.2.5.3	<i>Structural Contextualization</i> .....	16
2.2.6	Google Gemini.....	16
2.2.7	Prompt Engineering .....	17
2.2.8	AI Acceptance Model (AIM).....	18
2.2.9	Rumus Slovin.....	19
BAB III	.....	21
3.1	Metode Pengembangan Perangkat Lunak ( <i>Prototyping</i> ) .....	21
3.2	Alat dan Bahan.....	23
3.2.1	Perangkat Keras ( <i>Hardware</i> ) .....	23
3.2.2	Perangkat Lunak ( <i>Software</i> ).....	23
3.3	Tahapan Pengembangan Sistem.....	24
3.3.1	Analisis Kebutuhan ( <i>Requirements Gathering</i> ).....	24
3.3.1.1	Pemodelan Fungsional Sistem ( <i>Use Case Diagram</i> ).....	25
3.3.1.2	Pemodelan Proses ( <i>Activity Diagram</i> ).....	27
3.3.2	Pembangunan Prototipe ( <i>Development of Prototype</i> ).....	28
3.3.3	Pengujian Umpan Balik Pengguna ( <i>User Testing and Feedback</i> )	29
3.3.4	Revisi atau Penyempurnaan ( <i>Prototype Revision or Refinement</i> )	31
3.3.5	Rilis Produk Akhir ( <i>Final Product Release</i> ).....	31
BAB IV	.....	33
4.1	Hasil Analisis Kebutuhan ( <i>Requirements Analysis</i> ).....	33
4.1.1	Temuan Survei Pra-Pengembangan sebagai Dasar Analisis.....	33
4.1.2	Spesifikasi Kebutuhan Fungsional.....	34
4.1.3	Spesifikasi Kebutuhan Non-Fungsional.....	36
4.1.4	Batasan Teknis API.....	38
4.2	Hasil Pembangunan Prototipe ( <i>Development of Prototype</i> ) .....	39
4.2.1	Lingkungan dan Struktur Direktori Proyek.....	39
4.2.2	Realisasi Arsitektur <i>Client-Server</i> .....	40
4.2.2.1	Lapisan <i>Frontend</i> (React 19 + Vite) .....	41

4.2.2.2	Lapisan <i>Backend</i> (Python FastAPI) .....	42
4.2.3	Implementasi Logika Traversal dan Mekanisme <i>Retry</i> .....	44
4.2.3.1	Fungsi <code>get_directory_contents_async()</code> - Traversal Asinkron .....	44
4.2.3.2	Penyediaan Lapis Ganda dengan Google AI Studio dan OpenRouter .....	46
4.2.4	Antarmuka Pengguna Prototipe ( <i>UI Showcase</i> ).....	48
4.2.5	Logika Penyusunan Prompt Statis .....	49
4.3	Hasil Pengujian dan Umpan Balik Iterasi 1 (Testing dan Feedback) .....	52
4.3.1	Hasil Pengujian Fungsional ( <i>Black Box Testing</i> ).....	52
4.3.2	Pelaksanaan <i>User Acceptance Testing</i> (UAT) & Profil Responden 53	
4.3.3	Hasil Uji Validitas dan Realibilitas.....	55
4.3.3.1	Uji Validitas (Korelasi Pearson) .....	56
4.3.3.2	Uji Realibilitas (Cronbach' Alpha).....	59
4.3.4	Hasil User Acceptance Testing (UAT) Berbasis Kerangka AIM .	60
4.3.5	Analisis Umpan Balik Kualitatif.....	62
4.3.6	Kesimpulan Iterasi 1 .....	63
4.4	Hasil Revisi dan Penyempurnaan Prototipe ( <i>Prototype Refinement</i> ) .....	64
4.4.1	Konsolidasi Penyedia Layanan AI ( <i>Single Provider</i> ) .....	64
4.4.2	Implementasi Fitur Penyuntingan Langsung ( <i>Live Edit</i> ) .....	65
4.4.3	Implementasi Panel Preferensi Kustomisasi ( <i>Preferences Panel</i> )	68
4.4.3.1	Parameter Kustomisasi.....	69
4.4.4	Optimasi Prompt Engineering Anti-Halusinasi .....	72
4.4.5	Implementasi Aliran Umpan Balik Real-time (SSE Streaming)...	77
4.4.6	Implementasi Fitur Konfigurasi AI Kustom ( <i>AI Config Model</i> )... 78	
4.4.6.1	<i>Hybrid Client-Server Architecture</i> .....	78
4.5	Hasil Pengujian dan Evaluasi Penerimaan Pengguna Tahap 2 (Iterasi 2) .....	80
4.5.1	Pelaksanaan UAT Iterasi 2 & Profil Responden.....	81
4.5.2	Hasil Uji Validitas (Korelasi Pearson) Tahap 2.....	81
4.5.2.1	Analisis Justifikasi Akademik atas Item Tidak Valid .....	83
4.5.3	Hasil Uji Reliabilitas Tahap 2 .....	83
4.5.4	Hasil Skor Rata-Rata AIM Tahap 2 .....	86
4.5.5	Analisis Umpan Balik Kualitatif Iterasi 2.....	89

4.5.6	Kesimpulan Iterasi 2 .....	90
4.5.7	Analisis Komparatif Hasil UAT Iterasi 1 dengan Iterasi 2 .....	91
4.6	Rilis Produk Final ( <i>Final Product Release</i> ).....	93
4.6.1	Status Deployment .....	93
4.7	Rekapitulasi Jadwal Penelitian ( <i>Project Timeline</i> ) .....	95
4.7.1	<i>Timeline</i> Pelaksanaan Penelitian (Jira Board).....	95
4.8	Pembahasan ( <i>Discussion</i> ).....	97
4.8.1	Paradoks Validitas-Kepuasan ( <i>Ceiling Effect</i> ).....	98
4.8.2	Keterbatasan Penelitian.....	98
BAB V	.....	100
5.1	Kesimpulan .....	100
5.2	Saran.....	101
DAFTAR PUSTAKA	.....	103
LAMPIRAN	.....	108
CURRICULUM VITAE (CV)	.....	146



## DAFTAR TABEL

Tabel 2.1 Ringkasan Tinjauan Pustaka .....	8
Tabel 4.2 Spesifikasi Kebutuhan Non-Fungsional.....	37
Tabel 4.3 Struktur Penyusunan Prompt Statis pada Prototipe Tahap Awal.....	50
Tabel 4.4 Hasil Pengujian Fungsional Prototipe.....	53
Tabel 4.5 Profil Demografi Responden (n=50) .....	54
Tabel 4.6 Hasil Uji Validitas Instrumen Kuesioner AIM (n=48, r-tabel=0,284)..	56
Tabel 4.7 Hasil Uji Realibilitas per Indikator AIM .....	60
Tabel 4.8 Skor Rata-rata per Indikator AIM (n=48) .....	61
Tabel 4.9 Perbandingan Infrastruktur Integrasi AI Tahap 1 dan Tahap 2 .....	65
Table 4.10 Dimensi Parameterisasi Dinamis Prompt Engineering (Iterasi 2) .....	71
Tabel 4.11 Hasil Uji Validitas UAT Iterasi 2 (n=29, r-tabel=0,3673).....	82
Tabel 4.12 Hasil Uji Reliabilitas per Indikator AIM (Tahap 2).....	84
Tabel 4.13 Hasil Analisis Skor Rata-rata per Indikator AIM (Tahap 2, n=29) ....	86
Tabel 4.14 Perbandingan Rata-rata Skor AIM: Iterasi 1 dengan Iterasi 2 .....	91
Tabel 4.15 Timeline Pelaksanaan Penelitian Berdasarkan Jira Board.....	96

STATE ISLAMIC UNIVERSITY  
SUNAN KALIJAGA  
YOGYAKARTA

## DAFTAR GAMBAR

Gambar 2.1 Kerangka Kerja Evaluasi Large Language Model (LLM) .....	13
Gambar 2.2 Perbandingan Mekanisme LLM Tanpa RAG dengan RAG .....	14
Gambar 3.3 Diagram Model <i>Prototyping</i> .....	22
Gambar 4.1 Struktur Direktori Proyek.....	40
Gambar 4.2 Diagram Arsitektur Deployment Sistem .....	42
Gambar 4.3 UML Sequence Diagram Alur Kerja Sistem.....	44
Gambar 4.4 Flowchart Algoritma Traversal Direktori Rekursif.....	45
Gambar 4.5 Implementasi Kode Traversal Direktori Rekursif.....	46
Gambar 4.6 Flowchart Integrasi Penyedia AI Ganda pada Prototipe Awal .....	47
Gambar 4.7 Tampilan Antarmuka Formulir README Generator Prototipe Awal .....	48
Gambar 4.8 Tampilan Hasil README Generator Prototipe Awal.....	49
Gambar 4.9 Anatomi Penyusunan Prompt Statis pada Prototipe Awal .....	50
Gambar 4.10 Implementasi Fungsi Penyusunan Prompt Statis .....	51
Gambar 4.11 Arsitektur aliran sinkronisasi dua arah (Two-Way Data Binding) .	66
Gambar 4.12 Implementasi React State dan Penanganan Sinkronisasi Dua Arah	67
Gambar 4.13 Antarmuka Panel Penyunting Interaktif Live Edit pada Prototipe Refined .....	68
Gambar 4.14 Tampilan Antarmuka Formulir README Generator dengan Panel Preferensi Kustomisasi (Preferences Panel) pada Prototipe Refined.....	69
Gambar 4.15 Logika Pemetaan Parameter Preferensi dan Konfigurasi Presets ...	74
Gambar 4.16 Konstruksi System Instruction dengan Batasan Validasi Aturan Wajib .....	75
Gambar 4.17 Konstruksi User Prompt, Penyertaan Konteks File Konfigurasi, dan Aturan Recency Bias.....	76
Gambar 4.18 Implementasi Logika Pengiriman Event Progresif (Server-Sent Events).....	78
Gambar 4.19 Alur Hybrid Client-Server Architecture.....	80
Gambar 4.20 Panel Status Deployment Vercel Proyek readme-backend.....	94

Gambar 4.21 Panel Status Deployment Vercel Proyek readme-frontend..... 95  
Gambar 4.22 Tampilan Jira Agile Board (Kanban) Proyek Penelitian..... 97



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

README.md merupakan salah satu komponen vital dalam repositori GitHub yang berfungsi sebagai dokumentasi awal suatu proyek perangkat lunak. Berkas ini umumnya berisi deskripsi proyek, tujuan pengembangan, panduan instalasi, hingga struktur sistem yang membantu pengembang lain memahami proyek dengan cepat. Dokumentasi yang baik tidak hanya meningkatkan keterbacaan repositori, tetapi juga mempermudah proses kolaborasi dan mendukung pemeliharaan perangkat lunak dalam jangka panjang. Pentingnya keberadaan dokumen ini didukung oleh studi empiris terhadap 5.000 repositori GitHub, yang membuktikan adanya korelasi positif yang signifikan antara kualitas dan kelengkapan berkas README.md dengan popularitas sebuah repositori [1].

Namun, pada praktiknya, masih terdapat kesenjangan besar (documentation gap) antara urgensi dokumentasi dengan implementasi di lapangan. Secara teoritis, dokumentasi yang baik merupakan aset tak ternilai bagi proyek perangkat lunak karena membantu pemangku kepentingan untuk menggunakan, memahami, memelihara, dan mengembangkan sistem, namun pada kenyataannya dokumentasi sering kali dipengaruhi oleh berbagai kekurangan seperti konten yang tidak memadai, usang, atau ambigu [2]. Berdasarkan survei yang dilakukan peneliti terhadap 50 responden (mahasiswa informatika dan pengembang perangkat lunak), ditemukan bahwa meskipun hanya 12,9% responden yang merasa README tidak penting, faktanya hanya 12% responden yang selalu melengkapi repositori mereka dengan dokumentasi tersebut. Masalah utama yang ditemukan adalah fokus pengembang yang lebih diarahkan pada implementasi fitur aplikasi (57,1%), faktor lupa atau malas (51%), serta anggapan bahwa pembuatan dokumentasi memakan waktu (42,9%). Selain itu, 32,7% responden mengaku tidak mengetahui format penulisan README yang baik. Kondisi ini menunjukkan adanya kebutuhan mendesak akan solusi yang dapat mengotomatisasi pembuatan dokumentasi secara praktis. Seiring dengan perkembangan teknologi kecerdasan buatan, khususnya *Large Language Model* (LLM), proses pembuatan teks kini dapat dilakukan secara

otomatis dengan mempertimbangkan konteks informasi yang tersedia. LLM memiliki kemampuan unik dalam memahami konteks bahasa dan menghasilkan teks yang koheren. Beberapa penelitian terdahulu telah mengeksplorasi potensi ini, seperti Dvivedi et al. [3] yang melakukan analisis perbandingan model LLM dan membuktikan bahwa model LLM secara konsisten mampu mengungguli kualitas dokumentasi buatan manusia. Penelitian lain terkait otomatisasi README terus berkembang, di antaranya Koreeda et al. [4] yang mengotomatisasi pembuatan README.md menggunakan LLM dengan pendekatan identifikasi kode representatif berbasis heuristik untuk menghasilkan dokumentasi yang koheren dan benar secara faktual. Selain itu, Cui et al. [5] mengusulkan kerangka kerja *multi-agent* (RMGenie) menggunakan model *Tree of Actions* untuk meningkatkan akurasi faktual dokumentasi, serta Emerson et al. [6] yang mengembangkan sistem dokumentasi berbasis *Retrieval-Augmented Generation* (RAG) untuk menghasilkan dokumentasi terstruktur.

Meskipun penelitian-penelitian tersebut menunjukkan potensi besar, terdapat celah penelitian (*research gap*) yang dapat dikembangkan. Pendekatan berbasis RAG seperti pada penelitian Emerson et al. [6] cenderung membutuhkan sumber daya komputasi yang lebih besar karena proses analisis dilakukan terhadap keseluruhan isi kode sumber. Padahal, untuk menghasilkan README.md yang efektif, pengembang lebih membutuhkan gambaran makro daripada detail logika baris-per-baris. Selain itu, banyak sistem yang sudah ada masih memiliki ambang batas penggunaan (*barrier to entry*) yang tinggi, seperti perlunya konfigurasi CLI atau integrasi *pipeline* yang rumit.

Oleh karena itu, penelitian ini mengusulkan pengembangan sistem otomatisasi README.md yang berfokus pada pendekatan *Structural Contextualization*. Sistem ini tidak membaca seluruh isi kode sumber [4], melainkan memetakan struktur proyek melalui API GitHub [7] dan mengekstraksi metadata dari file konfigurasi kunci (seperti package.json atau requirements.txt) sebagai *ground truth* untuk memitigasi halusinasi AI [8]. Dengan platform *web-*

*based*, sistem ini dirancang untuk memberikan kemudahan bagi pengembang dalam menghasilkan draf dokumentasi yang terstruktur dalam hitungan detik.

Berdasarkan permasalahan dan celah penelitian tersebut, penelitian ini bertujuan untuk "Rancang Bangun Sistem Otomatisasi README.md pada *Repository* GitHub dengan Teknologi *Large Language Model*". Sistem ini diharapkan dapat meningkatkan efisiensi waktu pengembang dan memperbaiki kualitas dokumentasi pada ekosistem pengembangan perangkat lunak secara keseluruhan.

Untuk mencapai tujuan tersebut, penelitian ini dilaksanakan melalui beberapa tahapan sistematis dengan menggunakan metode *Prototyping* secara iteratif. Tahap pertama adalah analisis kebutuhan (*requirements gathering*) yang didasarkan pada hasil survei pra-pengembangan terhadap 50 responden untuk mengidentifikasi kebutuhan fungsional dan non-fungsional sistem. Tahap kedua adalah pembangunan prototipe awal (*development of prototype*), di mana sistem diimplementasikan berbasis arsitektur *client-server* dengan pendekatan *Structural Contextualization*. Tahap ketiga adalah pengujian dan pengumpulan umpan balik pengguna (*user testing and feedback*) melalui *User Acceptance Testing* (UAT) iterasi pertama terhadap 48 responden menggunakan instrumen berbasis kerangka *AI Acceptance Model* (AIM). Tahap keempat adalah revisi dan penyempurnaan prototipe (*prototype refinement*) berdasarkan temuan dari iterasi pertama, mencakup konsolidasi penyedia AI, penambahan fitur *Live Edit*, Panel Preferensi, dan optimasi *prompt engineering* anti-halusinasi. Tahap kelima adalah rilis produk akhir (*final product release*) yang mencakup UAT iterasi kedua terhadap 29 responden, diikuti penerapan sistem secara publik dan evaluasi akhir terhadap seluruh perbaikan yang telah dilakukan.

## 1.2 Rumusan Masalah

1. Bagaimana merancang dan membangun sistem untuk mengotomatisasi pembuatan README.md pada *repository* GitHub?
2. Bagaimana kualitas dan kesesuaian README.md yang dihasilkan oleh sistem dengan kebutuhan dokumentasi *repository* GitHub?

### 1.3 Batasan Masalah

1. Sistem yang dikembangkan hanya berfokus pada otomatisasi pembuatan file README.md dan tidak mencakup pembuatan dokumentasi lain seperti wiki, issue, atau dokumentasi teknis lanjutan.
2. Sistem hanya memproses *repository* GitHub yang bersifat *public*, sehingga *repository private* tidak menjadi bagian dari penelitian ini.
3. Model *Large Language Model* yang digunakan dalam penelitian ini adalah Google Gemini 2.5 Flash yang diakses melalui Google AI Studio, tanpa melakukan perbandingan dengan model LLM lainnya.
4. Evaluasi sistem dilakukan berdasarkan kualitas README.md yang dihasilkan, meliputi aspek struktur, kelengkapan, dan kesesuaian dengan kebutuhan dokumentasi *repository* GitHub dengan melibatkan pengujian langsung kepada pengguna (*user testing*).
5. Sistem dikembangkan dalam bentuk aplikasi *web-based* berupa *open source* yang memungkinkan pengguna mengirimkan *link repository* GitHub dan memperoleh hasil README.md dalam bentuk tampilan teks, salinan kode, atau unduhan file.
6. Sistem difokuskan untuk menangani *repository* dengan skala kecil hingga menengah. *Repository* dengan tingkat kompleksitas tinggi atau ukuran besar berpotensi membutuhkan waktu proses yang lebih lama atau menghasilkan kegagalan proses, sehingga tidak menjadi fokus utama dalam penelitian ini.

### 1.4 Tujuan Penelitian

1. Merancang dan mengimplementasikan sistem untuk mengotomatisasi pembuatan README.md pada *repository* GitHub
2. Menganalisis kualitas dan kesesuaian README.md yang dihasilkan oleh sistem dengan kebutuhan dokumentasi *repository* GitHub

## 1.5 Manfaat Penelitian

### 1.5.1 Manfaat Praktis

1. Bagi Pengembang Perangkat Lunak: Membantu mengatasi hambatan waktu dan kesulitan dalam memulai penyusunan dokumentasi (*cold start problem*), sehingga pengembang dapat tetap fokus pada implementasi fitur tanpa mengabaikan kualitas dokumentasi repositori.
2. Bagi Komunitas *Open Source*: Meningkatkan standarisasi dan keterbacaan proyek pada GitHub, sehingga memudahkan kolaborasi antar kontributor dan mempercepat pemahaman pengguna baru terhadap suatu proyek perangkat lunak.
3. Bagi Mahasiswa/Pemula: Menyediakan referensi struktur README.md yang baik secara otomatis, sekaligus menjadi sarana edukasi mengenai komponen apa saja yang harus ada dalam sebuah dokumentasi proyek yang profesional.

### 1.5.2 Manfaat Teoritis Akademis

1. Pengembangan Implementasi LLM: Memberikan referensi mengenai implementasi *Large Language Model* (LLM) dengan pendekatan *Context-Augmented Generation* (CAG) khusus untuk domain otomasi dokumentasi kode sumber.
2. Studi Efektivitas AI: Menjadi bahan kajian atau referensi bagi peneliti selanjutnya dalam mengevaluasi kualitas luaran teks teknis yang dihasilkan oleh model generatif AI berbasis struktur meta-data repositori.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan meliputi tahap analisis kebutuhan, perancangan dan implementasi sistem, serta evaluasi penerimaan pengguna (*User Acceptance Testing / UAT*) dalam dua siklus iteratif diperoleh simpulan sebagai berikut:

1. Sistem otomatisasi pembuatan README.md berbasis kecerdasan buatan berhasil dirancang dan dibangun menggunakan metode *Prototyping*, yaitu pendekatan pengembangan bertahap di mana sistem dibangun, diuji oleh pengguna, dan terus disempurnakan secara berulang berdasarkan umpan balik yang diterima. Secara fungsional, sistem bekerja melalui serangkaian tahapan: dimulai dari penerimaan URL repositori GitHub dari pengguna, dilanjutkan dengan pengambilan metadata repositori dan pemetaan seluruh struktur foldernya secara otomatis melalui GitHub REST API, kemudian membaca isi berkas konfigurasi kunci seperti package.json atau requirements.txt untuk memahami dependensi dan teknologi yang digunakan proyek. Seluruh informasi ini disusun menjadi sebuah konteks terstruktur yang dikirimkan ke model AI (Google Gemini) melalui teknik *prompt engineering* berlapis yang dirancang khusus untuk mencegah halusinasi dan memastikan keluaran selalu dalam format Markdown yang bersih. Proses ini berjalan secara asinkron dan hasilnya ditampilkan kepada pengguna secara bertahap melalui teknologi *streaming* real-time (SSE), sehingga pengguna tidak perlu menunggu keseluruhan proses selesai. Pengguna juga dapat menyesuaikan gaya dan tujuan README melalui Panel Preferensi, serta menyunting hasil langsung di halaman yang sama melalui fitur *Live Edit*. Sistem telah dirilis secara publik dan dapat diakses bebas di <https://readmegeneratorai.vercel.app>.

2. Kualitas dan tingkat penerimaan sistem diukur menggunakan kerangka *AI Acceptance Model* (AIM) melalui dua tahap pengujian bersama pengguna nyata. Pada prototipe awal, sistem menggunakan dua penyedia layanan AI sekaligus OpenRouter dan Google AI Studio sebagai mekanisme *dual-provider* untuk menjamin ketersediaan layanan. Namun pada pengujian tahap pertama (48 responden), pendekatan ini justru menimbulkan tiga masalah utama: koneksi yang sering gagal akibat *rate limit* dari OpenRouter, tidak adanya cara untuk memperbaiki hasil yang kurang tepat, dan ketidakjelasan kebijakan penggunaan kunci API yang menimbulkan kekhawatiran bagi pengguna. Setelah sistem diperbaiki dengan mengkonsolidasikan penyedia AI ke Google AI Studio saja yang lebih stabil dan andal, menambahkan fitur *Live Edit* untuk koreksi hasil, menghadirkan Panel Preferensi (termasuk pengaturan nada penulisan) yang meningkatkan kesesuaian dan kualitas akhir README yang dihasilkan, serta menyisipkan edukasi kebijakan penggunaan pengujian tahap kedua (29 responden) menunjukkan peningkatan pada seluruh 8 dimensi penilaian, dengan penurunan kekhawatiran risiko yang paling mencolok ( $\Delta = -1,46$ ) dan peningkatan niat untuk menggunakan kembali sistem yang paling tinggi ( $\Delta = +0,45$ ). Secara keseluruhan, instrumen kuesioner dinyatakan andal dengan nilai *Cronbach's Alpha* sebesar 0,8205, dan seluruh 29 responden menyatakan sistem ini sangat layak dijadikan alat bantu utama dalam membuat dokumentasi repositori.

## 5.2 Saran

Berdasarkan hasil penelitian, keterbatasan yang ditemukan, dan masukan dari para responden, berikut saran yang dapat menjadi pertimbangan untuk pengembangan dan penelitian selanjutnya:

1. Dukungan Repositori Privat.

Saat ini sistem hanya dapat memproses repositori GitHub yang bersifat publik. Ke depan, sistem dapat dikembangkan agar juga bisa menangani

repositori privat melalui mekanisme login akun GitHub, sehingga lebih banyak pengguna yang dapat memanfaatkannya.

2. Dukungan Platform Lain Selain GitHub.

Sesuai batasan penelitian, sistem saat ini hanya mendukung repositori GitHub yang bersifat publik. Menambahkan dukungan untuk platform repositori lain seperti GitLab atau Bitbucket akan membuat sistem dapat digunakan oleh komunitas pengembang yang jauh lebih luas.

3. Cadangan Penyedia AI

Sebagaimana diakui dalam keterbatasan penelitian, sistem saat ini bergantung sepenuhnya pada satu layanan AI (Google AI Studio), yang berpotensi menjadi titik kegagalan tunggal. Sebagai tindak lanjut, pengembang dapat menambahkan mekanisme beralih secara otomatis ke layanan AI lain saat terjadi gangguan, sehingga sistem lebih tahan terhadap masalah eksternal.

4. Fitur Riwayat Generasi.

Menambahkan fitur penyimpanan riwayat agar pengguna dapat melihat dan mengakses kembali README yang pernah dibuat sebelumnya, terutama bagi yang mengelola banyak repositori sekaligus.

5. Perluasan Skala Pengujian.

Pengujian tahap kedua hanya melibatkan 29 mahasiswa dari satu program studi. Penelitian selanjutnya sebaiknya melibatkan responden yang lebih beragam dari berbagai institusi dan latar belakang agar hasil penilaian lebih dapat digeneralisasikan.

6. Penggunaan Metode Pengukuran yang Lebih Tepat.

Pada pengujian tahap kedua, ditemukan bahwa alat ukur statistik yang biasa digunakan (korelasi Pearson) kurang cocok ketika hampir semua pengguna memberikan nilai tertinggi secara bersamaan. Penelitian selanjutnya

disarankan untuk menggunakan metode pengukuran tambahan yang lebih sesuai untuk kondisi ini.

#### 7. Perbandingan Antar-Model AI.

Penelitian ini hanya menggunakan satu model AI (Google Gemini 2.5 Flash). Membandingkan hasilnya dengan model AI lain seperti GPT-4o atau Claude akan memberikan gambaran yang lebih lengkap tentang model mana yang paling cocok untuk tugas pembuatan dokumentasi teknis.

### DAFTAR PUSTAKA

- [1] T. Wang, S. Wang, and T.-H. (Peter) Chen, “Study the correlation between the readme file of GitHub projects and their popularity,” *Journal of Systems and Software*, vol. 205, p. 111806, 2023, doi: <https://doi.org/10.1016/j.jss.2023.111806>.
- [2] E. Aghajani *et al.*, “Software documentation: the practitioners’ perspective,” in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, in ICSE ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 590–601. doi: 10.1145/3377811.3380405.
- [3] S. S. Dvivedi, V. Vijay, S. L. R. Pujari, S. Lodh, and D. Kumar, “A Comparative Analysis of Large Language Models for Code Documentation Generation,” in *Proceedings of the 1st ACM International Conference on AI-Powered Software, AIware 2024, Porto de Galinhas, Brazil, July 15-16, 2024*, B. Adams, T. Zimmermann, I. Ozkaya, D. Lin, and J. M. Zhang, Eds., ACM, 2024. doi: 10.1145/3664646.3664765.
- [4] Y. Koreeda, T. Morishita, O. Imaichi, and Y. Sogawa, “LARCH: Large Language Model-based Automatic Readme Creation with Heuristics,” in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, in CIKM ’23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 5066–5070. doi: 10.1145/3583780.3614744.
- [5] X. Cui, J. Wu, Z. Li, T. Luo, and X. Ling, “RMGenie: An LLM-Based Agent Framework for *Open Source* Software README Generation,” in *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2025, pp. 505–516. doi: 10.1109/ICSME64153.2025.00052.

- [6] A. Emerson, T. Meehan, M. Rogers, W. Cowen, and C. Darabos, “CodeDocs: GenAI to generate documentation from Git Repositories,” in *Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration*, in PEARC '25. New York, NY, USA: Association for Computing Machinery, 2025. doi: 10.1145/3708035.3736102.
- [7] F. Di Lauro, “GitHub-Sourced Web API Evolution: A Large-Scale OpenAPI Dataset,” in *Web Engineering*, K. and M. M. and H. S. and K. H. and Q. E. Stefanidis Kostas and Systä, Ed., Cham: Springer Nature Switzerland, 2024, pp. 360–368. doi: [https://doi.org/10.1007/978-3-031-62362-2\\_26](https://doi.org/10.1007/978-3-031-62362-2_26).
- [8] F. Liu *et al.*, “Beyond Functional Correctness: Exploring Hallucinations in LLM-Generated Code,” *IEEE Transactions on Software Engineering*, vol. 52, no. 3, pp. 1037–1055, 2026, doi: 10.1109/TSE.2026.3657432.
- [9] D. Yang *et al.*, “DocAgent: A Multi-Agent System for Automated Code Documentation Generation,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, P. Mishra, S. Muresan, and T. Yu, Eds., Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 460–471. doi: 10.18653/v1/2025.acl-demo.44.
- [10] E. Jabrayilzade, A. Yurtoğlu, and E. Tüzün, “Taxonomy of inline code comment smells,” *Empir. Softw. Eng.*, vol. 29, no. 3, p. 58, 2024, doi: 10.1007/s10664-023-10425-5.
- [11] Y. Liu, E. Noei, and K. Lyons, “How ReadMe files are structured in *open source* Java projects,” *Inf. Softw. Technol.*, vol. 148, p. 106924, Aug. 2022, doi: 10.1016/J.INFSOF.2022.106924.
- [12] A. H. C. van Kampen *et al.*, “ENCORE: a practical implementation to improve reproducibility and transparency of computational research,” *Nat. Commun.*, vol. 15, no. 1, p. 8117, 2024, doi: 10.1038/s41467-024-52446-8.
- [13] Z. Chen *et al.*, “MDEval: Evaluating and Enhancing Markdown Awareness in Large Language Models,” in *Proceedings of the ACM on Web Conference 2025*, in WWW '25. New York, NY, USA: Association for Computing Machinery, 2025, pp. 2981–2991. doi: 10.1145/3696410.3714674.
- [14] D. Kang, T. Kang, and J. Jang, “Papers with code or without code? Impact of GitHub repository usability on the diffusion of machine learning research,” *Inf. Process. Manag.*, vol. 60, no. 6, p. 103477, 2023, doi: <https://doi.org/10.1016/j.ipm.2023.103477>.

- [15] Y. Chang *et al.*, “A Survey on Evaluation of Large Language Models,” *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, Mar. 2024, doi: 10.1145/3641289.
- [16] W. Fan *et al.*, “A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 6491–6501. doi: 10.1145/3637528.3671470.
- [17] Z. Li, C. Li, M. Zhang, Q. Mei, and M. Bendersky, “Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, F. Dernoncourt, D. Preo\c{t}iuc-Pietro, and A. Shimorina, Eds., Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 881–893. doi: 10.18653/v1/2024.emnlp-industry.66.
- [18] M. Reid *et al.*, “Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context,” *ArXiv*, vol. abs/2403.05530, 2024, [Online]. Available: <https://api.semanticscholar.org/CorpusID:268297180>
- [19] Y.-Y. Liu *et al.*, “A comprehensive taxonomy of prompt engineering techniques for large language models,” *Front. Comput. Sci.*, vol. 20, no. 3, p. 2003601, 2025, doi: 10.1007/s11704-025-50058-z.
- [20] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, “Unleashing the potential of prompt engineering for large language models,” *Patterns*, vol. 6, no. 6, p. 101260, Jun. 2025, doi: 10.1016/J.PATTER.2025.101260.
- [21] L. Xue, J. Mahat, and N. Ghazali, “Technology Acceptance Model in Artificial Intelligence in Education: A Meta-Analysis,” *Sage Open*, vol. 16, May 2026, doi: 10.1177/21582440251409441.
- [22] I. Baroni, G. R. Calegari, D. Scandolari, and I. Celino, “AI-TAM: a model to investigate user acceptance and collaborative intention in human-in-the-loop AI applications,” *Hum. Comput.*, vol. 9, pp. 1–21, 2022, [Online]. Available: <https://api.semanticscholar.org/CorpusID:249029221>
- [23] D. Shin, “The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI,” *Int. J. Hum. Comput. Stud.*, vol. 146, p. 102551, 2021, doi: <https://doi.org/10.1016/j.ijhcs.2020.102551>.
- [24] C. S. Sargent, A. Koohang, K. Floyd, and R. Kilburn, “Artificial Intelligence: Ethical concerns, trust, and risk,” *Issues in Information Systems*, vol. 25, no. 2, pp. 71–83, 2024, doi: 10.48009/2\_iis\_2024\_106.

- [25] N. A. Dahri *et al.*, “Extended TAM based acceptance of AI-Powered ChatGPT for supporting metacognitive self-regulated learning in education: A mixed-methods study,” *Heliyon*, vol. 10, no. 8, p. e29317, 2024, doi: <https://doi.org/10.1016/j.heliyon.2024.e29317>.
- [26] N. I. Majdina, B. Pratikno, and A. Tripena, “PENENTUAN UKURAN SAMPEL MENGGUNAKAN RUMUS BERNOULLI DAN SLOVIN: KONSEP DAN APLIKASINYA,” *Jurnal Ilmiah Matematika dan Pendidikan Matematika*, 2024, [Online]. Available: <https://api.semanticscholar.org/CorpusID:2711785568>
- [27] B. Antoro, “ANALISIS PENERAPAN FORMULA SLOVIN DALAM PENELITIAN ILMIAH: KELEBIHAN, KELEMAHAN, DAN KESALAHAN DALAM PERSPEKTIF STATISTIK,” *Jurnal Multidisiplin Sosial dan Humaniora*, vol. 1, pp. 53–63, Nov. 2024, doi: 10.70585/jmsh.v1i2.38.
- [28] Maryani, H. Prabowo, F. L. Gaol, and A. N. Hidayanto, “Comparison of the System Development Life Cycle and Prototype Model for Software Engineering,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 4, pp. 155–162, Apr. 2022, doi: 10.46338/ijetae0422\_19.
- [29] A. Obaid, “Using Prototypes in Agile Software Development,” *International Journal of Computers and Informatics*, vol. 3, pp. 23–38, Feb. 2024, doi: 10.59992/IJCI.2024.v3n2p2.
- [30] G. Zhang, J. H. Schuessler, and C. Y. Shao, “Small-World Phenomenon of Global Open-Source Software Collaboration on Github: A Social Network Analysis,” *Journal of Global Information Management (JGIM)*, vol. 33, no. 1, pp. 1–24, 2025, doi: 10.4018/JGIM.387412.
- [31] X. Hou *et al.*, “Large Language Models for Software Engineering: A Systematic Literature Review,” *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 8, Dec. 2024, doi: 10.1145/3695988.
- [32] E. Tounzal, “Rendering Interactive Generative User Interface Components Based on React Libraries When Working with Large Language Models,” *Universal Library of Engineering Technology*, vol. 3, no. 1, pp. 116–121, Mar. 2026, doi: 10.70315/uloap.ulete.2026.0301020.
- [33] E. Krisnawati, K. D. Artanti, and N. H. Umar, “Validity and Reliability Test of Research Instruments on Husbands’ Support on Barriers to Using Long-Term Contraceptive Methods among Multiparous Active Acceptors in Surabaya,” *Media Gizi Kesmas*, vol. 13, no. 2, pp. 659–664, Dec. 2024, doi: 10.20473/mgk.v13i2.2024.659-664.

- [34] I. Hussey, T. Alsalti, F. Bosco, M. Elson, and R. Arslan, “An Aberrant Abundance of Cronbach’s Alpha Values at .70,” *Adv. Methods Pract. Psychol. Sci.*, vol. 8, p., 2025, doi: 10.1177/25152459241287123.
- [35] G. Blinowski, A. Ojdowska, and A. Przybylek, “Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation,” *IEEE Access*, vol. 10, p. 1, Jan. 2022, doi: 10.1109/ACCESS.2022.3152803.
- [36] I. Karabey Aksakalli, T. Çelik, A. Can, and B. Tekinerdogan, “Deployment and communication patterns in microservice architectures: A systematic literature review,” *Journal of Systems and Software*, vol. 180, p. 111014, Jun. 2021, doi: 10.1016/j.jss.2021.111014.
- [37] X. Tan, Y. Jiang, Y. Yang, and H. Xu, “Towards End-to-End Optimization of LLM-based Applications with Ayo,” in *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2024, p. doi: 10.1145/3676641.3716278.
- [38] Y. Fu *et al.*, “ServerlessLLM: low-latency serverless inference for large language models,” in *Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation*, in OSDI’24. USA: USENIX Association, 2024.