

**TUGAS AKHIR**

**PENGEMBANGAN GENERATOR DOKUMENTASI API BERBASIS  
OPENAPI SPECIFICATION DENGAN DUKUNGAN *ARTIFICIAL  
INTELLIGENCE***



Disusun Oleh:

Alfian Maulana

22106050058

STATE ISLAMIC UNIVERSITY  
**SUNAN KALIJAGA**  
YOGYAKARTA

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

**2026**

## LEMBAR PERNYATAAN KEASLIAN

### LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Alfian Maulana  
NIM : 22106050058  
Program Studi : Informatika  
Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya, bahwa skripsi saya yang berjudul **“Pengembangan Generator Dokumentasi API Berbasis OpenAPI Specification dengan Dukungan Artificial Intelligence”** merupakan penelitian 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, 25 Februari 2026

Yang membuat pernyataan,

STATE ISLAMIC UNIVERSITY  
SUNAN KALIJAGA  
YOGYAKARTA



Alfian Maulana  
22106050058

## LEMBAR PERSETUJUAN SKRIPSI/TUGAS AKHIR

### LEMBAR PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir

Lamp :-

Kepada  
Yth.  
Dekan Fakultas Sains dan Teknologi  
UIN 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 Saudara:

Nama : Alfian Maulana  
NIM : 22106050058  
Judul : Pengembangan Generator Dokumentasi API Berbasis  
Skripsi : OpenAPI Specification dengan Dukungan Artificial  
Intelligence

Sudah dapat diajukan 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 skripsi / tugas akhir Saudari dapat segera dimunaqosyahkan. Atas perhatiannya saya ucapkan terima kasih.

*Wassalamu'alaikum wr. wb.*

STATE ISLAMIC UNIVERSITY  
SUNAN KALIJAGA  
YOGYAKARTA

Yogyakarta, 27 Februari 2026  
Pembimbing,



Prof. Dr. Ir. Shofwatul 'Uyun, M.Kom.  
NIP. 198205112006042002

# 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-523/Un.02/DST/PP.00.9/03/2026

Tugas Akhir dengan judul : Pengembangan Generator Dokumentasi API Berbasis OpenAPI Specification dengan Dukungan Artificial Intelligence

yang dipersiapkan dan disusun oleh:

Nama : ALFIAN MAULANA  
Nomor Induk Mahasiswa : 22106050058  
Telah diujikan pada : Rabu, 04 Maret 2026  
Nilai ujian Tugas Akhir : A

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

### TIM UJIAN TUGAS AKHIR



Ketua Sidang

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

Valid ID: 69aa73b3a02bb



Penguji I

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

Valid ID: 69aa72da8cc06



Penguji II

Eko Hadi Gunawan, M.Eng.  
SIGNED

Valid ID: 69aa715c00145



Yogyakarta, 04 Maret 2026

UIN Sunan Kalijaga  
Dekan Fakultas Sains dan Teknologi

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

Valid ID: 69aa7b1ba3b3e

## **LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR**

Tugas Akhir ini tidak dipublikasikan, tetapi tersedia di perpustakaan dalam lingkungan UIN 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 UIN Sunan Kalijaga Yogyakarta.



## INTISARI

Dokumentasi *Application Programming Interface* (API) merupakan komponen penting dalam pengembangan perangkat lunak modern, khususnya pada arsitektur berbasis layanan. Namun, proses dokumentasi yang dilakukan secara manual sering menimbulkan permasalahan seperti deskripsi yang tidak lengkap, ketidakkonsistenan struktur, dan minimnya contoh penggunaan (*API documentation smells*). Penelitian ini bertujuan untuk mengembangkan Spec2Docs, sebuah sistem otomatisasi dokumentasi API berbasis analisis statis dan kecerdasan buatan pada framework Express.js. Metode pengembangan yang digunakan adalah *Extreme Programming* (XP) dengan empat iterasi yang mencakup perancangan autentikasi, pengembangan mesin parser berbasis *Abstract Syntax Tree* (AST), pemetaan ke standar OpenAPI Specification (OAS) v3.0.3, serta integrasi layanan kecerdasan buatan untuk pengayaan deskripsi dan pembuatan *mock data* otomatis. Evaluasi *usability* dilakukan menggunakan metode *System Usability Scale* (SUS) terhadap 31 responden yang terdiri dari mahasiswa dan praktisi IT. Hasil pengujian menunjukkan rata-rata skor SUS sebesar 81,935 yang berada pada kategori *Acceptable* dengan *Grade A (Excellent)*. Hasil tersebut menunjukkan bahwa sistem memiliki tingkat kegunaan yang tinggi serta mampu menghasilkan dokumentasi API yang lebih terstruktur, informatif, dan mudah dipahami dibandingkan pendekatan manual.

**Kata Kunci:** Dokumentasi API, Express.js, OpenAPI Specification, *Artificial Intelligence*, *Extreme Programming*.

## ABSTRACT

*Application Programming Interface (API) documentation plays a crucial role in modern software development, particularly in service-based architectures. However, manual documentation processes often lead to issues such as incomplete descriptions, inconsistent structures, and lack of usage examples, commonly referred to as API documentation smells. This research aims to develop Spec2Docs, an automated API documentation system based on static code analysis and artificial intelligence for Express.js applications. The development process adopted the Extreme Programming (XP) methodology through four iterations, including authentication design, development of an Abstract Syntax Tree (AST)-based parser, mapping to OpenAPI Specification (OAS) v3.0.3, and integration of AI services for automatic description enhancement and mock data generation. Usability evaluation was conducted using the System Usability Scale (SUS) involving 31 respondents from academic and professional IT backgrounds. The system achieved an average SUS score of 81,935, classified as Acceptable with Grade A (Excellent). These results indicate that the system provides high usability and produces structured, informative, and comprehensible API documentation compared to manual approaches.*

**Keywords:** *API Documentation, Express.js, OpenAPI Specification, Artificial Intelligence, Extreme Programming.*

## MOTTO

*“Sesungguhnya beserta kesulitan ada kemudahan.”*

Surah Al-Insyirah ayat 6

*"Keberhasilan bukanlah milik orang yang pintar.*

*Keberhasilan adalah kepunyaan mereka yang senantiasa berusaha"*

B.J. Habibie

*“A million dreams are keeping me awake”*

A Million Dreams (The Greatest Showman)



STATE ISLAMIC UNIVERSITY  
**SUNAN KALIJAGA**  
YOGYAKARTA

## HALAMAN PERSEMBAHAN

Dengan memanjatkan rasa syukur ke hadirat Allah SWT Yang Maha Pengasih lagi Maha Penyayang, karya Tugas Akhir ini saya persembahkan sebagai bentuk penghargaan dan ungkapan cinta yang tulus. Semoga karya ini membawa manfaat dan menjadi amal kebaikan. Dengan penuh hormat dan rasa terima kasih, Tugas Akhir ini saya persembahkan kepada:

1. Almarhumah Ibu tercinta, Suis Bandiyah, atas kasih sayang, doa, perhatian, serta pengorbanan yang tidak pernah putus dalam setiap perjalanan hidup saya. Segala ketulusan dan kesabaranmu menjadi kekuatan terbesar bagi saya untuk terus melangkah.
2. Bapak tercinta, Suparno, yang telah menjadi penopang kehidupan saya. Nilai, nasihat, serta perjuangan yang telah diberikan selama ini untuk membantu langkah hidupku.
3. Kedua kakak saya, Ariska Putri Diah S. dan Arnaz Ardi Surya Putra P. , yang senantiasa memberikan dukungan, dorongan semangat dan motivasi hingga saya dapat menyelesaikan tugas akhir ini.
4. Sahabat dan rekan seperjuangan yang telah memberikan bantuan, dukungan, dan semangat dalam berbagai bentuk.

Terima kasih yang sebesar-besarnya atas setiap doa dan kebaikan yang diberikan. Semoga Allah SWT membalas segala kebaikan dengan rahmat dan keberkahan yang berlipat ganda.

## KATA PENGANTAR

*Alhamdulillah Rabbil 'Alamin*, penulis mengucapkan syukur kepada Allah Subhanahu wa ta'ala atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Pengembangan Generator Dokumentasi API Berbasis OpenAPI Specification dengan Dukungan *Artificial Intelligence*” dengan baik. Shalawat serta salam senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang penuh cahaya, dan semoga kita semua diberi syafaatnya di akhir zaman.

Penyelesaian tugas akhir ini tidak terlepas dari bantuan, dukungan, serta doa dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan 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. Dr. Muhammad Mustakim, S.T., M.T. selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Ir. Muhammad Didik Rohmad Wahyudi, S.T., MT. selaku dosen penasihat akademik penulis selama menempuh perkuliahan di UIN Sunan Kalijaga.
5. Prof. Dr. Ir. Shofwatul 'Uyun, S.T., M.Kom., IPM., ASEAN Eng. selaku dosen pembimbing tugas akhir yang telah memberikan arahan, bimbingan, serta masukan berharga selama proses penyusunan tugas akhir ini.
6. Seluruh dosen Program Studi Informatika UIN Sunan Kalijaga yang telah memberikan ilmu dan pengalaman berharga selama masa perkuliahan.
7. Kedua orang tua penulis, Bapak Suparno dan Almh. Suis Bandiyah, atas segala dukungan, doa, kasih sayang dan pengorbanan yang tak pernah berhenti hingga penulis dapat menempuh pendidikan tinggi ini.
8. Kedua kakak penulis, Ariska Putri Diah S. dan Arnaz Ardi Surya Putra P. yang telah mendukung dan memotivasi penulis dalam menyelesaikan tugas akhir.

9. Seluruh Mahasiswa Informatika terutama Angkatan 2022 yang telah bersama-sama berjuang dan mendukung pada saat perkuliahan.
10. Sahabat-sahabat penulis, grup PPT, messi, sebestiee dan teman dekat penulis lainnya yang selalu menemani dan memberikan semangat selama masa studi.
11. Seluruh Teman Penulis di UIN Sunan Kalijaga Yogyakarta yang kebersamai dalam setiap langkah penulis dalam menempuh Pendidikan.

Penulis menyadari bahwa tugas akhir ini masih ada kekurangan secara penulisan maupun materi. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi perbaikan di masa mendatang. Semoga tugas akhir ini dapat memberikan manfaat serta menjadi kontribusi dalam pengembangan ilmu pengetahuan, khususnya pada bidang rekayasa perangkat lunak.

Yogyakarta, 25 Februari 2026

Penulis

Alfian Maulana

NIM. 22106050058



STATE ISLAMIC UNIVERSITY  
SUNAN KALIJAGA  
YOGYAKARTA

## DAFTAR ISI

LEMBAR PERNYATAAN KEASLIAN .....	i
LEMBAR PERSETUJUAN SKRIPSI/TUGAS AKHIR.....	ii
LEMBAR PENGESAHAN TUGAS AKHIR .....	iii
LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR .....	iv
INTISARI.....	v
ABSTRACT.....	vi
MOTTO .....	vii
HALAMAN PERSEMBAHAN .....	viii
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI .....	7
2.1 Tinjauan Pustaka .....	7
2.2 Landasan Teori .....	9
2.2.1 Rekayasa Perangkat Lunak.....	9
2.2.2 Dokumentasi Perangkat Lunak.....	14

2.2.3 Otomasi dalam Rekayasa Perangkat Lunak.....	15
2.2.4 <i>Application Programming Interface</i> (API).....	17
2.2.5 RESTful API.....	19
2.2.6 Dokumentasi API.....	20
2.2.7 <i>API Documentation Smells</i> .....	22
2.2.8 OpenAPI Specification .....	24
2.2.9 <i>Artificial Intelligence</i> dalam Dokumentasi Perangkat Lunak.....	25
2.2.10 Express.js <i>API Framework</i> .....	29
2.2.11 Penilaian Kualitas Dokumentasi API .....	30
2.2.12 <i>Unified Modeling Language</i> (UML).....	31
2.2.13 Perancangan Struktur Data .....	35
2.2.14 Metode Pengujian .....	36
2.2.15 Perbandingan <i>Tools</i> Dokumentasi API.....	40
BAB III METODE PENGEMBANGAN .....	43
3.1 Lokasi dan Waktu.....	43
3.2 Alat dan Bahan .....	43
3.2.1 Perangkat Keras .....	43
3.2.2 Perangkat Lunak .....	43
3.3 Metode Pengembangan .....	45
3.4 Tahapan Metode <i>Extreme Programming</i> .....	46
3.5 Tahapan Iterasi Pengembangan.....	49
3.5.1 Rencana Iterasi Pengembangan .....	49
3.5.2 Iterasi 1 : <i>Project Setup</i> dan Autentikasi .....	50
3.5.3 Iterasi 2 : <i>Core Documentation Engine</i> (Non AI).....	51
3.5.4 Iterasi 3 : Integrasi <i>Artificial Intelligence</i> dan Penilaian Dokumentasi .....	52

3.5.5 Iterasi 4 : <i>Persistence</i> , Manajemen Riwayat dan Unduh Dokumentasi	52
3.6 Analisis Kebutuhan .....	53
3.7 Metode Pengujian.....	54
3.7.1 <i>Black Box Testing</i> .....	55
3.7.2 <i>Unit Testing</i> .....	55
3.7.2 <i>System Usability Scale (SUS)</i> .....	56
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM .....	57
4.1 Analisis Kebutuhan Sistem .....	57
4.1.1 Gambaran Umum Sistem.....	57
4.1.2 <i>User Stories</i> .....	59
4.2 Perancangan Sistem.....	60
4.2.1 Arsitektur Sistem .....	61
4.2.2 <i>Use Case Diagram</i> .....	62
4.3 Iterasi Pengembangan Pertama .....	64
4.3.1 Perancangan ( <i>Planning</i> ) Iterasi Pertama.....	65
4.3.2 Perancangan ( <i>Design</i> ) Iterasi Pertama.....	67
4.3.3 Implementasi Kode ( <i>Coding</i> ) Iterasi Pertama .....	73
4.3.4 Pengujian ( <i>Testing</i> ) Iterasi Pertama.....	76
4.3.5 Evaluasi Iterasi Pertama.....	78
4.4 Iterasi Pengembangan Kedua .....	78
4.4.1 Perencanaan ( <i>Planning</i> ) Iterasi Kedua .....	79
4.4.2 Perancangan ( <i>Design</i> ) Iterasi Kedua.....	81
4.4.3 Implementasi Kode ( <i>Coding</i> ) Iterasi Kedua .....	87
4.4.4 Pengujian ( <i>Testing</i> ) Iterasi Kedua.....	90
4.4.5 Evaluasi Iterasi Kedua .....	93

4.5 Iterasi Pengembangan Ketiga .....	94
4.5.1 Perencanaan ( <i>Planning</i> ) Iterasi Ketiga .....	94
4.5.2 Perancangan ( <i>Design</i> ) Iterasi Ketiga .....	96
4.5.3 Implementasi Kode ( <i>Coding</i> ) Iterasi Ketiga.....	101
4.5.4 Pegujian ( <i>Testing</i> ) Iterasi Ketiga .....	103
4.5.5 Evaluasi Iterasi Ketiga .....	104
4.6 Iterasi Pengembangan Keempat .....	105
4.6.1 Perencanaan ( <i>Planning</i> ) Iterasi Keempat .....	105
4.6.2 Perancangan ( <i>Design</i> ) Iterasi Keempat.....	108
4.6.3 Implementasi Kode ( <i>Coding</i> ) Iterasi Keempat .....	112
4.6.4 Pegujian ( <i>Testing</i> ) Iterasi Keempat.....	116
4.6.5 Evaluasi Iterasi Keempat .....	118
4.7 Evaluasi <i>Usability</i> Sistem.....	118
4.7.1 Instrumen Pengujian .....	119
4.7.3 Data Awal Hasil Pengujian.....	120
4.7.4 Hasil Perhitungan Skor SUS.....	122
4.8 Pembahasan Akhir.....	125
BAB V KESIMPULAN DAN SARAN.....	127
5.1 Kesimpulan.....	127
5.2 Saran .....	128
DAFTAR PUSTAKA .....	130
LAMPIRAN.....	1

## DAFTAR GAMBAR

Gambar 2.1 Tahapan Metode <i>Extreme Programming</i> .....	11
Gambar 3.1 Alur pengembangan Spec2Docs Tiap Iterasi .....	47
Gambar 4.1 Arsitektur Sistem Spec2Docs .....	61
Gambar 4.2 <i>Use Case Diagram</i> Sistem Spec2Docs .....	63
Gambar 4.3 <i>Activity Diagram</i> Register ke Spec2Docs .....	68
Gambar 4.4 <i>Activity Diagram</i> Login ke Spec2Docs .....	69
Gambar 4.5 <i>Wireframe</i> Halaman Beranda .....	71
Gambar 4.6 <i>Wireframe</i> Halaman Login dan Registrasi .....	72
Gambar 4.7 <i>Wireframe</i> Halaman Dashboard Generate .....	73
Gambar 4.8 Hasil Implementasi Halaman Landing Page .....	74
Gambar 4.9 Hasil Implementasi Halaman Registrasi .....	75
Gambar 4.10 Hasil Implementasi Halaman Login .....	75
Gambar 4.11 Hasil Implementasi Halaman Dashboard Utama .....	76
Gambar 4.12 <i>Activity Diagram</i> Membuat Dokumentasi API .....	83
Gambar 4.13 Struktur Data Spec2Docs .....	85
Gambar 4.14 <i>Wireframe</i> Halaman Penampil Dokumentasi .....	87
Gambar 4.15 Implementasi Cuplikan Logika Parser pada Backend .....	88
Gambar 4.16 Hasil Implementasi Halaman Penampil Dokumentasi .....	89
Gambar 4.17 Hasil Implementasi Parsing Dialog .....	90
Gambar 4.18 Hasil <i>Unit Testing</i> Modul Parser .....	93
Gambar 4.19 Sequence Diagram Generasi Dokumentasi Berbasis AI .....	97
Gambar 4.20 Cuplikan Integrasi Gemini API pada modul aiService .....	101
Gambar 4.21 Hasil Implementasi Antarmuka Dokumentasi dengan <i>AI Enrichment</i> .....	103
Gambar 4.22 <i>Activity Diagram</i> Mengekspor Dokumentasi API .....	109
Gambar 4.23 <i>Wireframe</i> Halaman Riwayat Dokumentasi .....	110
Gambar 4.24 <i>Wireframe</i> Halaman Panduan Penggunaan Sistem .....	111
Gambar 4.25 <i>Wireframe</i> Halaman Pengaturan .....	112
Gambar 4.26 Hasil Implementasi Halaman Riwayat Proyek Dokumentasi .....	114
Gambar 4.27 Hasil Implementasi Halaman Pengaturan .....	114

Gambar 4.28 Hasil Implementasi Halaman Panduan ..... 115  
Gambar 4.29 Hasil Implementasi Dark Mode ..... 116



## DAFTAR TABEL

Tabel 2.1 Kriteria Penilaian Skor SUS .....	39
Tabel 2.2 Hasil Analisis Perbandingan Tools Dokumentasi API .....	40
Tabel 3.1 Daftar Perangkat Lunak Pengembangan.....	44
Tabel 3.2 Rencana Iterasi Pengembangan Spec2Docs .....	49
Tabel 4.1 Daftar <i>User Stories</i> Sistem Spec2Docs.....	59
Tabel 4.2 <i>Acceptance Test Criteria</i> Iterasi Pertama.....	66
Tabel 4.3 Hasil Pengujian <i>Black Box</i> Iterasi Pertama .....	76
Tabel 4.4 <i>Acceptance Test Criteria</i> Iterasi Kedua .....	81
Tabel 4.5 Hasil Pengujian <i>Black Box</i> Iterasi Kedua .....	91
Tabel 4.6 <i>Acceptance Test Criteria</i> Iterasi Ketiga .....	96
Tabel 4.7 Parameter Penilaian Kualitas Dokumentasi.....	99
Tabel 4.8 Hasil Pengujian <i>Black Box</i> Iterasi Ketiga .....	103
Tabel 4.9 <i>Acceptance Test Criteria</i> Iterasi Keempat .....	107
Tabel 4.10 Hasil Pengujian <i>Black Box</i> Iterasi Keempat .....	116
Tabel 4.11 Kuesioner <i>System Usability Scale</i> (SUS).....	119
Tabel 4.12 Rekapitulasi Data Awal Hasil Pengujian SUS.....	120
Tabel 4.13 Hasil Perhitungan Skor SUS.....	122

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi informasi mendorong penggunaan *Application Programming Interface* (API) sebagai komponen utama dalam pengembangan perangkat lunak modern, khususnya pada pengembangan sistem *backend* berbasis web. API berperan sebagai penghubung antar komponen sistem maupun antar aplikasi yang berbeda, sehingga memungkinkan pertukaran data dan integrasi layanan secara efisien [1], [2]. Seiring meningkatnya adopsi pendekatan API sebagai produk dan berkembangnya ekosistem perangkat lunak yang saling terhubung, dokumentasi API menjadi komponen krusial yang menentukan keberhasilan pemanfaatan API oleh pengembang [3].

Dokumentasi API berfungsi sebagai sumber informasi utama bagi pengembang untuk memahami cara kerja, struktur, serta perilaku API. Dokumentasi yang baik harus bersifat konsisten, mudah dipahami, dan selalu selaras dengan implementasi API yang sesungguhnya. Namun, dalam praktiknya, proses pendokumentasian API masih menjadi tantangan serius di banyak organisasi. Dokumentasi sering kali dibuat secara manual, terpisah dari kode sumber, dan tidak diperbarui secara berkelanjutan seiring perubahan API, sehingga berpotensi menimbulkan ketidaksesuaian antara dokumentasi dan implementasi [4].

Kondisi tersebut diperkuat oleh Postman State of the API Report 2025 yang menunjukkan bahwa 58% pengembang masih menghabiskan waktu yang signifikan untuk mendokumentasi API, sementara 93% tim pengembang API menghadapi permasalahan kolaborasi akibat dokumentasi yang tidak konsisten, tidak terpusat, atau tidak terbaru [4]. Tingginya angka tersebut mengindikasikan bahwa dokumentasi API bukan sekadar aktivitas pendukung, melainkan faktor utama yang memengaruhi produktivitas pengembang, kualitas kolaborasi tim, dan kecepatan pengembangan perangkat lunak.

Lebih lanjut, berbagai penelitian dan studi praktik industri telah mengidentifikasi fenomena yang dikenal sebagai *API documentation smells*, yaitu pola-pola permasalahan dalam dokumentasi API seperti deskripsi endpoint yang ambigu, parameter yang tidak terdokumentasi dengan baik, contoh *request* atau *response* yang tidak representatif, serta struktur dokumentasi yang tidak konsisten. *API documentation smells* ini telah divalidasi oleh para praktisi dan terbukti berdampak negatif terhadap produktivitas pengembang, di mana lebih dari 95% responden melaporkan penurunan efisiensi kerja akibat kualitas dokumentasi API yang buruk. Dampak tersebut mencakup meningkatnya waktu pembelajaran API, kesalahan integrasi, serta meningkatnya kebutuhan komunikasi informal untuk mengklarifikasi perilaku API yang seharusnya dapat dijelaskan melalui dokumentasi [3].

Di sisi lain, tren adopsi kecerdasan buatan dalam pengembangan perangkat lunak semakin meningkat. Hasil Postman State of the API Report 2025 menunjukkan bahwa 89% pengembang telah menggunakan teknologi AI dalam aktivitas pengembangan sehari-hari, termasuk untuk membantu pembuatan dokumentasi API secara otomatis [4]. Hal ini membuka peluang baru untuk memanfaatkan *Large Language Models* (LLM) dalam menghasilkan deskripsi naratif dokumentasi API secara lebih konsisten dan terstandarisasi [5]. Namun demikian, pemanfaatan AI dalam konteks dokumentasi API masih menghadapi tantangan, khususnya terkait keterandalan, konsistensi dengan spesifikasi teknis, serta integrasinya dengan standar dokumentasi yang dapat dibaca oleh manusia dan mesin.

OpenAPI Specification (OAS) telah menjadi standar *de facto* dalam pendeskripsian API berbasis REST, karena mampu merepresentasikan struktur API secara formal dalam format YAML atau JSON yang mudah diintegrasikan dengan berbagai alat pengujian, dokumentasi, dan manajemen API. Meskipun demikian, proses penyusunan spesifikasi OpenAPI secara manual tetap membutuhkan usaha yang besar dan rentan terhadap kesalahan, terutama pada proyek dengan skala menengah hingga besar [1], [6].

Berdasarkan permasalahan tersebut, diperlukan sebuah pendekatan yang mampu mengotomatisasi proses pembuatan dokumentasi API secara terstandarisasi, konsisten, dan selaras dengan implementasi API. Pendekatan tersebut diharapkan tidak hanya menghasilkan spesifikasi OpenAPI yang valid, tetapi juga mampu menyediakan deskripsi naratif yang mudah dipahami oleh pengembang. Oleh karena itu, penelitian ini berfokus pada pengembangan generator dokumentasi API otomatis berbasis OpenAPI Specification yang didukung oleh integrasi *Artificial Intelligence* untuk membantu pembuatan deskripsi endpoint, penjelasan parameter, serta contoh request dan response, tanpa menggantikan peran utama alat dokumentasi API yang telah ada [5], [6]. Dengan adanya sistem ini, diharapkan permasalahan dokumentasi API, khususnya yang berkaitan dengan *API documentation smells*, dapat diminimalkan, sehingga meningkatkan kualitas dokumentasi, produktivitas pengembang, serta efektivitas kolaborasi tim dalam pengembangan perangkat lunak.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat diambil rumusan masalah dalam pengembangan proyek ini, sebagai berikut :

1. Bagaimana merancang dan membangun sistem generator dokumentasi API berbasis *OpenAPI Specification* sebagai sumber utama dokumentasi yang dihasilkan secara otomatis dan konsisten?
2. Bagaimana mengintegrasikan teknologi *Artificial Intelligence* untuk membantu menghasilkan deskripsi API agar lebih jelas dan mudah dipahami tanpa mengurangi akurasi spesifikasi?

## 1.3 Batasan Masalah

Dalam ruang lingkup proyek ini perlu diberikan batasan-batasan untuk membatasi lingkup pengembangan agar sesuai dengan penyelesaian masalah, fokus dan terarah. Berikut adalah Batasan-batasan yang ditetapkan untuk generator dokumentasi API berbasis *OpenAPI Specification* dengan dukungan AI :

1. Generator ini bukan berfungsi sebagai pengganti utama *tools* dokumentasi diluar sana, namun generator ini memiliki posisi sendiri sebagai pendekatan

dokumentasi yang otomatis dan tersatandarisasi *OpenAPI Specification* yang tetap dapat diintegrasikan ke *tools* yang lain.

2. Perangkat lunak ini berbasis *website* dan harus dijalankan menggunakan internet.
3. Bahasa antarmuka (*copywriting*) dan sistem secara keseluruhan menggunakan Bahasa Inggris. Hal ini disesuaikan dengan praktik umum pada publikasi *tools open source* yang mayoritas menggunakan Bahasa Inggris sebagai standar global untuk kolaborasi.
4. Standar *de facto* yang digunakan dalam pembuatan dokumentasi ini terpaku pada *OpenAPI Specification*, bukan jenis atau gaya dokumentasi yang lain seperti API Bluperint, RAML, dsb.
5. Penggunaan *Artificial Intellegence* disini khususnya pada *Large Language Models* bukan merupakan penelitian utama sehingga tidak mementingkan jenis LLM yang paling baik namun hanya dipilih berdasarkan kecocokan untuk pembuatan deskripsi naratif.
6. Pemilihan *Large Language Models* disini menggunakan Gemini API model karena merupakan API yang stabil, cepat, memiliki *context window* yang besar dan banyak digunakan dalam hal serupa terutama pada pemahaman serta pembuatan teks naratif.
7. Fitur integrasi AI disini terbatas pada pembuatan deskripsi naratif dokumentasi API tidak mmebuat dokumentasi secara penuh.
8. Untuk arsitektur API hanya dibatasi untuk REST API atau RESTful API karena sesuai tujuan dari *OpenAPI Specification* yang dirancang khusus untuk mendeskripsikan arsitektur tersebut dalam format yang bisa dibaca manusia dan mesin yaitu YAML/JSON.
9. Untuk bahasa dan framework yang didukung oleh *tools* ini adalah Javascript dengan framework Express.js. Hal ini dikarenakan Javascript merupakan bahasa yang paling populer dan Express.js sebagai kerangka kerja yang paling banyak digunakan dalam lingkup backend javascript dengan karakteristik *unopiniated* yang memberi kebebasan penuh terhadap

pengembang untuk menggunakannya sehingga menjadikan Express.js cocok untuk menjadi eksperimen yang menantang dan luas dampaknya.

10. Format ekspor dokumentasi terdiri dari beberapa pilihan yaitu YAML/JSON, Markdown, HTML. Pilihan tersebut karena sering digunakan dalam pendokumentasian API yang dapat diintegrasikan ke *tools* lain seperti Postman *testing*.
11. Unggah berkas (*file upload / dropzone*) pada sistem belum dilengkapi dengan mekanisme pengamanan lanjutan terhadap serangan *seperti Cross Site Scripting (XSS)* atau injeksi skrip berbahaya lainnya. Oleh karena itu, sistem ini tidak ditujukan untuk menerima atau memproses berkas yang mengandung skrip berbahaya, serta belum dirancang untuk skenario pengujian keamanan tingkat lanjut terhadap *file upload*.

#### 1.4 Tujuan Penelitian

Tujuan dari pengembangan generator dokumentasi ini antara lain:

1. Merancang arsitektur dan membangun sistem generator dokumentasi API berbasis *OpenAPI Specification* sebagai *single source of truth* dokumentasi API.
2. Mengimplementasikan mekanisme otomatisasi dokumentasi API yang memastikan konsistensi antara spesifikasi API dan dokumentasi yang dihasilkan.
3. Mengintegrasikan teknologi *Artificial Intelligence* untuk menghasilkan deskripsi naratif sesuai konteks, terstruktur, dan mudah dipahami.

#### 1.5 Manfaat Penelitian

Secara praktis dan teoritis, penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Bagi Pengembang Perangkat Lunak

Sistem generator dokumentasi API yang dikembangkan dapat membantu pengembang dalam menghasilkan dokumentasi API secara otomatis, konsisten, dan terstandarisasi, sehingga mengurangi waktu dan usaha yang diperlukan untuk membuat dokumentasi secara manual. Dengan

dokumentasi yang lebih jelas dan mudah dipahami, pengembang diharapkan dapat mempercepat proses integrasi API serta meminimalkan kesalahan implementasi.

2. Bagi Tim Pengembang dan Organisasi.

Dokumentasi API yang dihasilkan secara otomatis dan selaras dengan spesifikasi OpenAPI dapat meningkatkan kualitas kolaborasi antar anggota tim pengembang. Dokumentasi yang konsisten dan selalu terbaru diharapkan dapat mengurangi miskomunikasi, mempercepat proses onboarding pengembang baru, serta meningkatkan efisiensi kerja tim dalam pengembangan dan pemeliharaan sistem berbasis API.

3. Bagi Praktik Dokumentasi API

Penelitian ini dapat menjadi contoh penerapan pendekatan dokumentasi API yang mengombinasikan standar teknis *OpenAPI Specification* dengan dukungan *Artificial Intelligence* sebagai alat bantu pembuatan deskripsi naratif. Pendekatan ini dapat dijadikan alternatif atau pelengkap bagi alat dokumentasi API yang telah ada, tanpa menggantikan peran utama tools dokumentasi tersebut.

4. Bagi Pengembangan Sistem di Masa Depan

Hasil penelitian ini dapat menjadi dasar pengembangan lebih lanjut untuk mendukung bahasa pemrograman, *framework*, atau arsitektur API lainnya. Selain itu, sistem yang dikembangkan juga berpotensi untuk diintegrasikan dengan berbagai alat pengujian, manajemen API, dan dokumentasi lainnya, sehingga memiliki fleksibilitas dan relevansi yang luas dalam ekosistem pengembangan perangkat lunak.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil perencanaan, perancangan, implementasi, dan pengujian yang telah dilakukan menggunakan metode *Extreme Programming (XP)*, penelitian ini berhasil mengembangkan sistem generator dokumentasi API otomatis yang mengombinasikan standarisasi spesifikasi teknis dengan dukungan *Artificial Intelligence*. Pendekatan ini dirancang untuk menjawab permasalahan dokumentasi API yang sering tidak konsisten, sulit diperbarui, dan rentan menimbulkan *documentation smells*, dengan menghadirkan solusi dokumentasi yang lebih terstruktur, otomatis, dan mudah dipahami. Berdasarkan rumusan masalah dan tujuan penelitian, maka diperoleh kesimpulan berikut

1. Perancangan dan pembangunan sistem generator dokumentasi API berbasis OpenAPI Specification berhasil direalisasikan sebagai sumber utama dokumentasi. Penelitian ini berhasil merancang arsitektur dan mengimplementasikan sistem yang mampu menghasilkan dokumentasi API secara otomatis dengan memanfaatkan OpenAPI Specification sebagai dasar representasi formal. Sistem melakukan proses analisis statis dan ekstraksi struktur endpoint, metode HTTP, parameter, serta skema respons langsung dari kode sumber untuk membentuk spesifikasi OpenAPI yang valid dan terstandarisasi. Dengan pendekatan ini, dokumentasi yang dihasilkan memiliki konsistensi tinggi terhadap implementasi API dan dapat berfungsi sebagai *single source of truth*. Selain itu, mekanisme otomatisasi yang dibangun mampu mengurangi ketergantungan pada dokumentasi manual, meminimalkan risiko ketidaksesuaian dokumentasi, serta mendukung praktik dokumentasi yang lebih berkelanjutan dalam pengembangan sistem berbasis API.
2. Integrasi *Artificial Intelligence* berhasil meningkatkan kualitas dokumentasi naratif tanpa mengurangi akurasi spesifikasi teknis yang terbukti teruji oleh mekanisme penilaian dokumentasi API. Penelitian ini juga berhasil mengintegrasikan teknologi *Artificial Intelligence*

berbasis *Large Language Models* menggunakan Gemini API sebagai komponen pendukung dalam menghasilkan deskripsi naratif dokumentasi API. AI dengan teknik *Prompt Engineering* yang baik dapat dimanfaatkan untuk menyusun penjelasan endpoint, deskripsi parameter, serta contoh request dan response secara lebih komunikatif dan kontekstual, namun tetap dikendalikan oleh struktur spesifikasi OpenAPI yang telah tervalidasi. Pendekatan ini memastikan bahwa AI tidak menjadi sumber kebenaran utama, melainkan berperan sebagai penguat kualitas dokumentasi yang berorientasi pada keterbacaan dan pemahaman pengguna. Hasil implementasi menunjukkan bahwa integrasi ini mampu meningkatkan kejelasan dokumentasi, mengurangi potensi *documentation smells*, serta mendukung efektivitas komunikasi dan kolaborasi dalam pengembangan perangkat lunak berbasis API.

## 5.2 Saran

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem Spec2Docs di masa mendatang guna meningkatkan fungsionalitas dan cakupan penggunaannya:

1. Perluasan Dukungan Bahasa dan Framework: Pengembangan selanjutnya disarankan untuk memperluas cakupan Bahasa dan framework tidak hanya terbatas pada Express.js, tetapi juga mencakup framework populer lainnya seperti NestJS atau Fastify.
2. Optimasi Parser: Peningkatan kemampuan mesin *parser* AST perlu terus dilakukan agar sistem dapat menangani pola logika kode yang lebih kompleks, seperti pemrosesan *middleware* bertingkat atau penggunaan variabel dinamis guna memastikan akurasi data teknis tetap terjaga pada proyek skala besar.
3. Peningkatan Kualitas AI: Eksplorasi lebih lanjut mengenai teknik *fine-tuning* pada model bahasa atau penyempurnaan instruksi prompt sangat dianjurkan agar hasil deskripsi fungsional dapat disesuaikan dengan standar industri tertentu.

4. Pengembangan Fitur Kolaborasi dan Kustomisasi Visual: Penambahan fitur kolaborasi tim yang memungkinkan beberapa pengembang untuk melakukan peninjauan (*review*) dan penyuntingan bersama pada draf dokumentasi sebelum diekspor akan memperkuat aspek kolaboratif dalam siklus pengembangan perangkat lunak. Terakhir, optimasi antarmuka pada sisi visualisasi dokumentasi statis dapat ditingkatkan dengan penambahan tema kustom agar hasil ekspor dapat lebih mudah disesuaikan dengan identitas *branding* dari masing-masing proyek pengembang.



## DAFTAR PUSTAKA

- [1] O. Baniş, D. Florea, R. Gyalai, and D. I. Curiac, “Automated specification-based testing of REST APIs,” *Sensors*, vol. 21, no. 16, Aug. 2021, doi: 10.3390/s21165375.
- [2] G. A. Mutiara, Periyadi, M. R. Alfarisi, M. A. Rifqi Zain, M. G. Rijali, and F. N. Rochim, “Design and implementation of a REST API-based client-server architecture for multi-sensor IoT monitoring,” *International Journal of Advanced Technology and Engineering Exploration*, vol. 12, no. 124, pp. 426–449, 2025, doi: 10.19101/IJATEE.2024.111101934.
- [3] J. Y. Khan, Md. T. I. Khondaker, G. Uddin, and A. Iqbal, “Automatic Detection of Five API Documentation Smells: Practitioners’ Perspectives,” Feb. 2021, [Online]. Available: <http://arxiv.org/abs/2102.08486>
- [4] Inc. Postman, “2025 State of the API Report,” 2025. Accessed: Dec. 14, 2025. [Online]. Available: <https://www.postman.com/state-of-api/2025/>
- [5] X. Peng *et al.*, “Automatic generation of API documentations for open-source projects,” in *Proceedings - 3rd International Workshop on Dynamic Software Documentation, DySDoc3 2018*, Institute of Electrical and Electronics Engineers Inc., Nov. 2018, pp. 7–8. doi: 10.1109/DySDoc3.2018.00010.
- [6] R. F. Vilela, S. Alves De Andrade, E. B. F. Santos, W. Silva, and P. H. D. Valle, “An Intelligent Agent for Automated Test Generation from OpenAPI Specifications.” [Online]. Available: <https://www.postman.com/>
- [7] A. A. H. Alzahrani, “Software Systems Documentation: A Systematic Review,” 2024. [Online]. Available: [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- [8] M. Meng, S. M. Steinhardt, and A. Schubert, “Optimizing API documentation: Some guidelines and effects,” in *SIGDOC 2020 - Proceedings of the 38th ACM International Conference on Design of Communication*, Association for Computing Machinery, Inc, Oct. 2020. doi: 10.1145/3380851.3416759.
- [9] A. Lercher, C. Macho, C. Bauer, and M. Pinzger, “Generating Accurate OpenAPI Descriptions from Java Source Code,” Oct. 2024, [Online]. Available: <http://arxiv.org/abs/2410.23873>
- [10] S. Wang, Y. Tian, and D. He, “Reduce API Debugging Overhead via Knowledge Prepositioning,” in *ACM Web Conference 2023 - Companion of the World Wide Web Conference, WWW 2023*, Association for Computing Machinery, Inc, Apr. 2023, pp. 57–60. doi: 10.1145/3543873.3587311.

- [11] Y. Jorelle, "Generation of API Documentation using Large Language Models -- Towards Self-explaining APIs," 2024. [Online]. Available: [www.aalto.fi](http://www.aalto.fi)
- [12] J. Golec and T. Hachaj, "Ten Natural Language Processing Tasks with Generative Artificial Intelligence," *Applied Sciences (Switzerland)*, vol. 15, no. 16, Aug. 2025, doi: 10.3390/app15169057.
- [13] M. Tonggiroh, V. B. A. Pardosi, Basiroh, and F. Nugroho, *Rekayasa Perangkat Lunak*. Kota Solok, 2024.
- [14] E. Ali, *Rekayasa-Perangkat-Lunak*. Yogyakarta, 2019.
- [15] E. M. T. Sanmocte and J. A. Costales, "Exploring Effectiveness in Software Development: A Comparative Review of System Analysis and Design Methodologies," *International Journal of Computer Theory and Engineering*, vol. 17, no. 1, pp. 36–37, 2025, doi: 10.7763/IJCTE.2025.V17.1367.
- [16] M. Gaborov *et al.*, "Journal of Applied Technical and Educational Sciences jATES Comparative analysis of agile and traditional methodologies in IT project management," vol. 11, no. 4, pp. 1–24, 2021, doi: 10.24368/jates.v11i4.279.
- [17] F. Uzomah, J. Agbana, and A. Joseph Alu, "A Comparative Analysis of Agile and Traditional Project Management Methodologies: Impact on Project Success Metrics in the IT Sector Amid Rapid Technological Evolution," *Educational Administration: Theory and Practice*, 2024, doi: 10.53555/kuey.v30i1.9237.
- [18] Y. A. A.-S. Saeed, Z. Zulkifli, S. B. S. Nasir, and W. B. Ahmed, "Factors in Determining Information Systems Development Methodologies: A Systematic Literature Review," *Sohar University Journal of Sustainable Business*, vol. 1, no. 1, pp. 1–15, Jan. 2025, doi: 10.69983/sujsb/0101/03.
- [19] A. Aljehani and M. Rizwan Qureshi, "Agile Method: Challenges and Adaptations for Complex Project Environments," *International Journal of Information Engineering and Electronic Business*, vol. 17, no. 3, pp. 84–98, Jun. 2025, doi: 10.5815/ijieeb.2025.03.06.
- [20] M. Reiter, "Comparative Analysis of Agile Frameworks: Scrum, Kanban, Extreme Programming," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 253, Springer Science and Business Media Deutschland GmbH, 2025, pp. 335–348. doi: 10.1007/978-3-031-89718-4\_27.

- [21] M. M. Muller and W. E. Tichy, "Case Study: Extreme Programming in a University Environment."
- [22] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User Stories and Natural Language Processing: A Systematic Literature Review," *IEEE Access*, vol. 9, pp. 53811–53826, 2021, doi: 10.1109/ACCESS.2021.3070606.
- [23] C. A. Binangkit, A. Voutama, and N. Heryana, "Pemanfaatan UML (Unified Modeling Language) dalam Perencanaan Sistem Pengelolaan Sewa Alat Musik Berbasis Website," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 2, pp. 1429–1436, Apr. 2023.
- [24] C. Augusto, A. Bertolino, I. DE Guglielmo Angelis, I. Francesca Lonetti, and I. Jesús Morán, "Large Language Models for Software Testing: A Research Roadmap," 2025. [Online]. Available: <https://code.visualstudio.com/>
- [25] W. Hadinata and L. Stianingsih, "Analisis Perbandingan Performa RESTful API antara Express.js dengan Laravel Framework," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 1, Jan. 2024, doi: 10.23960/jitet.v12i1.3845.
- [26] S. S. Sarker and T. T. Ifty, "Automated and Context-Aware Code Documentation Leveraging Advanced LLMs," Sep. 2025, [Online]. Available: <http://arxiv.org/abs/2509.14273>
- [27] A. A. H. Alzahrani, "Pre-During-After Software Development Documentation (PDA-SDD): A Phase-Based Approach for Comprehensive Software Documentation in Modern Development Paradigms," *Computers*, vol. 14, no. 9, Sep. 2025, doi: 10.3390/computers14090378.
- [28] K. Ajeigbe and O. Emma, "Dynamic Documentation Generation with AI," 2024.
- [29] F. Lemos, F. F. Correia, A. Aguiar, and P. G. G. Queiroz, "Live software documentation of design pattern instances," *PeerJ Comput. Sci.*, vol. 10, 2024, doi: 10.7717/PEERJ-CS.2090.
- [30] R. Kumar, "Standardizing API Contracts: Enabling Interoperability in Distributed Systems," *International Journal of Multidisciplinary Research and Growth Evaluation.*, vol. 3, no. 6, pp. 718–727, 2022, doi: 10.54660/ijmrge.2022.3.6.718-727.
- [31] Quang Hai Khuat, "API documentation: A guide to making your data APIs developer-friendly," *World Journal of Advanced Research and Reviews*, vol. 26, no. 1, pp. 1541–1556, Apr. 2025, doi: 10.30574/wjarr.2025.26.1.1179.

- [32] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, and D. Mishra, "RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions," May 01, 2022, *MDPI*. doi: 10.3390/app12094369.
- [33] S. Endrikat, S. Hanenberg, R. Robbes, and A. Stefik, "How do API documentation and static typing affect API usability?," in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, May 2014, pp. 632–642. doi: 10.1145/2568225.2568299.
- [34] S. V. Sheta, "The Importance of Software Documentation in the Development and Maintenance Phases," 2023. [Online]. Available: <http://www.veterinaria.org>
- [35] M. Coblenz, W. Guo, K. Voozhian, and J. S. Foster, "A Qualitative Study of REST API Design and Specification Practices."
- [36] M. Alenezi and M. Akour, "AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions," *Applied Sciences (Switzerland)*, vol. 15, no. 3, Feb. 2025, doi: 10.3390/app15031344.
- [37] N. Nazia *et al.*, "Intelligent Automation in Software Engineering: Transforming Developer Productivity and Code Quality Through Machine Learning," *Spectrum of Engineering Sciences*, vol. 3, no. 12, pp. 492–514, Dec. 2025, doi: 10.5281/zenodo.17985963.
- [38] R. Lisgiani and S. Nurmajid, "Implementasi Autentikasi dari Sisi Backend pada Arsitektur Microservices Menggunakan Express Js," *Infotronik: Jurnal Teknologi Informasi dan Elektronika*, vol. 7, no. 1, p. 27, Jun. 2022, doi: 10.32897/infotronik.2022.7.1.1330.
- [39] R. Huang, M. Motwani, I. Martinez, and A. Orso, "Generating REST API Specifications through Static Analysis," in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, May 2024, pp. 1311–1323. doi: 10.1145/3597503.3639137.
- [40] J. Curtis, "Student Research Abstract: On Language-Agnostic Abstract-Syntax Trees," in *Proceedings of the ACM Symposium on Applied Computing*, Association for Computing Machinery, Apr. 2022, pp. 1619–1622. doi: 10.1145/3477314.3506962.
- [41] W. Sun *et al.*, "Abstract Syntax Tree for Programming Language Understanding and Representation: How Far Are We?," New York, NY, USA: Association for Computing Machinery (ACM), Jan. 2023, pp. 1–47. doi: 10.1145/nnnnnnn.nnnnnn.

- [42] V. Terragni, A. Vella, P. Roop, and K. Blincoe, “The Future of AI-Driven Software Engineering,” *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 5, May 2025, doi: 10.1145/3715003.
- [43] E. Dos Santos, R. Gallegos, and S. Cruz -Argentina, “Insights into API Management Quality Characterization, Measurement and Evaluation.” [Online]. Available: <https://apilandscape.apiscene.io/>
- [44] R. Koçi, X. Franch, P. Jovanovic, and A. Abelló, “Web API evolution patterns: A usage-driven approach,” *Journal of Systems and Software*, vol. 198, Apr. 2023, doi: 10.1016/j.jss.2023.111609.
- [45] P. Desai and S. Fruitwala, “Swagger/OpenAPI Specification as a Governance Tool for Internal Data Products: Enabling Standardization, Transparency, and Control,” *The American Journal of Applied Sciences*, vol. 7, no. 07, pp. 39–47, Jul. 2025, doi: 10.37547/tajas/Volume07Issue07-05.
- [46] S. Kanthed, “Rest vs. GraphQL: Comparative Analysis of API Design Approaches,” *International Journal of Multidisciplinary Research and Growth Evaluation.*, vol. 4, no. 1, pp. 984–991, 2023, doi: 10.54660/ijmrge.2023.4.1.984-991.
- [47] C. Yang *et al.*, “APIDocBooster: An Extract-Then-Abstract Framework Leveraging Large Language Models for Augmenting API Documentation,” Jan. 2024, [Online]. Available: <http://arxiv.org/abs/2312.10934>
- [48] H. Chen, Y. Li, C. Chen, F. Lin, and W. Li, “OpenAI for OpenAPI: Automated generation of REST API specification via LLMs,” Jan. 2026, [Online]. Available: <http://arxiv.org/abs/2601.12735>
- [49] S. Serbout, A. Romanelli, and C. Pautasso, “ExpressO: From Express.js Implementation Code to OpenAPI Interface Descriptions,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2023, pp. 29–44. doi: 10.1007/978-3-031-36889-9\_4.
- [50] S. Wang, Y. Tian, and D. He, “gDoc: Automatic Generation of Structured API Documentation,” in *ACM Web Conference 2023 - Companion of the World Wide Web Conference, WWW 2023*, Association for Computing Machinery, Inc, Apr. 2023, pp. 57–60. doi: 10.1145/3543873.3587311.
- [51] E. Sundberg, T. Ekmark, and W. Y. Ayele, “Validating API Design Requirements for Interoperability: A Static Analysis Approach Using OpenAPI,” Nov. 2025, [Online]. Available: <http://arxiv.org/abs/2511.17836>

- [52] S. Chauhan *et al.*, “From Specification to Service: Accelerating API-First Development Using Multi-Agent Systems,” Oct. 2025, [Online]. Available: <http://arxiv.org/abs/2510.19274>
- [53] J. Kridatama and S. Dan Teknologi Pemanfaatan, “Pemanfaatan Artificial Intelligence pada Pembelajaran dan Asesmen di Era Digitalisasi.”
- [54] A. Satav, “APIs as Business Drivers: The Rise of Ecosystem-Led Growth,” *Journal of Global Research in Computer Sciences*, 2025, doi: 10.4172/2229.
- [55] “Applying Artificial Intelligence to Analyze Emotions in Social Media Comments using Large Language Models \_ Astiti \_ Sistemasi\_ Jurnal Sistem Informasi”.
- [56] K. Lazar *et al.*, “Generating OpenAPI Specifications from Online API Documentation with Large Language Models.” [Online]. Available: <https://swagger.io/specification/>
- [57] A. Alansari and H. Luqman, “Large Language Models Hallucination: A Comprehensive Survey,” Oct. 2025, [Online]. Available: <http://arxiv.org/abs/2510.06265>
- [58] G. Amasanti and J. Jahic, “The Impact of AI-Generated Solutions on Software Architecture and Productivity: Results from a Survey Study,” Jun. 2025, [Online]. Available: <http://arxiv.org/abs/2506.17833>
- [59] T. Fukuda, T. Nakagawa, K. Miyazaki, and S. Tokumoto, “Development of Automated Software Design Document Review Methods Using Large Language Models,” Sep. 2025, [Online]. Available: <http://arxiv.org/abs/2509.09975>
- [60] Gemini Team *et al.*, “Gemini: A Family of Highly Capable Multimodal Models,” May 2025, [Online]. Available: <http://arxiv.org/abs/2312.11805>
- [61] J. White *et al.*, “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT,” Feb. 2023, [Online]. Available: <http://arxiv.org/abs/2302.11382>
- [62] S. Schulhoff *et al.*, “The Prompt Report: A Systematic Survey of Prompt Engineering Techniques,” Feb. 2025, [Online]. Available: <http://arxiv.org/abs/2406.06608>
- [63] I. W. Syahputri, E. K. Budiardjo, and P. O. H. Putra, “Unlocking the Potential of the Prompt Engineering Paradigm in Software Engineering: A Systematic Literature Review,” Sep. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/ai6090206.

- [64] S. Ruangtanusak, P. Taveekitworachai, and K. Pipatanakul, "Talk Less, Call Right: Enhancing Role-Play LLM Agents with Automatic Prompt Optimization and Role Prompting," Oct. 2025, [Online]. Available: <http://arxiv.org/abs/2509.00482>
- [65] M. F. Nugroho *et al.*, "Rancang Bangun REST API Aplikasi Manajemen Toko Menggunakan Nodejs pada Cantika Paint," 2023.
- [66] M. Szewczyk and M. Skublewska-Paszowska, "Performance comparison of development frameworks in selected environments in REST API architecture Porównanie wydajności szkieletów programistycznych w wybranych środowiskach w architekturze REST API," 2025.
- [67] U. Effiong, E. E. Ogheneovo, and B. O. Eke, "Architecture of an Advanced Software Evaluation DSS using Multicriteria Weighted Sum Algorithm," *International Journal of Innovation and Developmental Technology of the Global Trends Alliance (IJIDTGA)*, vol. 2, pp. 2456–4184, 2025, doi: 10.5281/zenodo.15363708.
- [68] Y. H. Syahputra, I. Santoso, and Z. Lubis, "Sistem Pendukung Keputusan Penerimaan Karyawan Terbaik Menggunakan Metode Weighted Sum Model (WSM)," 2022.
- [69] M. O. Esangbedo, S. Bai, S. Mirjalili, and Z. Wang, "Evaluation of human resource information systems using grey ordinal pairwise comparison MCDM methods," *Expert Syst. Appl.*, vol. 182, Nov. 2021, doi: 10.1016/j.eswa.2021.115151.
- [70] B. Ezell, C. J. Lynch, and P. T. Hester, "Methods for weighting decisions to assist modelers and decision analysts: A review of ratio assignment and approximate techniques," Nov. 2021, *MDPI*. doi: 10.3390/app112110397.
- [71] B. Bhatt and M. Nandu, "An Overview of Structural UML Diagrams," *International Research Journal of Engineering and Technology*, 2021, [Online]. Available: [www.irjet.net](http://www.irjet.net)
- [72] Y. Aryani, I. Aqil, and B. Paramita, "Penerapan Unified Modeling Language (UML) pada Digitalisasi Sistem Informasi Perpustakaan," *Digital Transformation Technology*, vol. 4, no. 2, pp. 1032–1040, Jan. 2025, doi: 10.47709/digitech.v4i2.5153.
- [73] P. Rizky Pangestu and A. Voutama, "Pemanfaatan UML (Unified Modelling Language) pada Sistem Pengelolaan Aspirasi Mahasiswa Berbasis Website," 2024.

- [74] F. Siewe and G. M. Ngounou, "On the Execution and Runtime Verification of UML Activity Diagrams," *Software*, vol. 4, no. 1, p. 4, Feb. 2025, doi: 10.3390/software4010004.
- [75] M. G. Alkhairi, S. P. A. Alkadri, and P. Y. Utami, "Implementasi Unit Testing Dan End-To-End Testing Pada Sistem Informasi Akademik Teknik Informatika," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 4, pp. 2208–2219, Nov. 2024, doi: 10.29100/jipi.v9i4.5626.
- [76] A. Valerian, D. Bernady, F. Wahyudi, J. P. Dinatha, and D. Barlelyano, "Implementasi Unit Testing, Integration Testing, System Testing, dan Validation Testing pada Aplikasi Berbasis Website (Studi Kasus: Kafe Saturdays)," 2025.
- [77] M. A. R. Hidayat, N. N. Izzati, A. R. P. Prihatama, Y. P. Wijaya, and S. Kurmilasari, "Pengujian Perangkat Lunak pada Website Ka'cake: Implementasi Unit Testing, Integration Testing, System Testing, dan Validation Testing untuk Menjamin Kualitas dan Keandalan Sistem," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 4, pp. 6805–6811, Aug. 2025.
- [78] J. R. Lewis and J. Sauro, "The Factor Structure of the System Usability Scale."
- [79] S. R. Arifin, "System Usability Scale (SUS) implementation in Ruang Baca Virtual – UT Library," *MATRIX: Jurnal Manajemen Teknologi dan Informatika*, vol. 14, no. 1, pp. 1–8, Mar. 2024, doi: 10.31940/matrix.v14i1.1-8.
- [80] K. T. Nugroho, B. Julianto, and D. F. Nur MS, "Usability Testing pada Sistem Informasi Manajemen AKN Pacitan Menggunakan Metode System Usability Scale," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 1, p. 74, Apr. 2022, doi: 10.23887/janapati.v11i1.43209.
- [81] N. Huda, F. Habrizons, A. Satriawan, M. Iranda, and T. Pramuda, "Analisis Usability Testing Menggunakan Metode SUS (System Usability Scale) Terhadap Kepuasan Pengguna Aplikasi Shopee," *SIMKOM*, vol. 8, no. 2, pp. 208–220, Aug. 2023, doi: 10.51717/simkom.v8i2.158.
- [82] V. Yoga Pudya Ardhana, "Evaluasi Usability E-Learning Universitas Qamarul Huda Menggunakan System Usability Scale (SUS)," 2022. [Online]. Available: <https://djournals.com/jieec>

- [83] F. Galuh Sembodo, G. Fadila Fitriana, and N. A. Prasetyo, "Evaluasi Usability Website Shopee Menggunakan System Usability Scale (SUS)," 2021. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [84] E. Kurniawan, A. Nata, and S. Royal, "Penerapan System Usability Scale (SUS) dalam Pengukuran Kebergunaan Website Program Studi di STMIK Royal," 2022. [Online]. Available: <http://jurnal.goretanpena.com/index.php/JSSR>
- [85] Y. Sriyeni, "Analisis Usability Aplikasi Investasi Digital Menggunakan Metode Heuristic Evaluation dan System Usability Scale," *Jurnal Ilmiah Informatika Global*, vol. 13, no. 2, Aug. 2022, doi: 10.36982/jiig.v13i2.2294.

