

**PENGEMBANGAN WEB ALAT BANTU PEMROSESAN REFERENSI
OTOMATIS PADA NASKAH ARTIKEL ILMIAH BERBASIS NLP**

SKRIPSI

**Disusun untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana**



Disusun Oleh:

Arvin Sultan Satria
2200018418

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN
YOGYAKARTA
2025**

**DEVELOPMENT OF A WEB-BASED TOOL FOR AUTOMATIC REFERENCE
PROCESSING IN SCIENTIFIC MANUSCRIPTS USING NLP**

THESIS

**Prepared to fulfill some of the requirements to
achieve the degree of Bachelor**



Prepared by:

Arvin Sultan Satria
2200018418

**INFORMATICS UNDERGRADUATE PROGRAM
FACULTY OF INDUSTRIAL TECHNOLOGY
UNIVERSITAS AHMAD DAHLAN
YOGYAKARTA
2025**

LEMBAR PERSETUJUAN PEMBIMBING

SKRIPSI

**PENGEMBANGAN WEB ALAT BANTU PEMROSESAN REFERENSI
OTOMATIS PADA NASKAH ARTIKEL ILMIAH BERBASIS NLP**

Dipersiapkan dan disusun oleh:

**ARVIN SULTAN SATRIA
2200018418**

**Program Studi S1 Informatika
Fakultas Teknologi Industri
Universitas Ahmad Dahlan**

Telah disetujui oleh:

Pembimbing

Dr. Ardiansyah S.T., M.Cs.

NIP/NIPM. 19790723 200309 111 0932301

LEMBAR PENGESAHAN

SKRIPSI

PENGEMBANGAN WEB ALAT BANTU PEMROSESAN REFERENSI OTOMATIS PADA NASKAH ARTIKEL ILMIAH BERBASIS NLP

Dipersiapkan dan disusun oleh:

ARVIN SULTAN SATRIA
2200018418

Telah dipertahankan di depan Dewan Penguji
pada tanggal Hari Bulan Tahun
dan dinyatakan telah memenuhi syarat

Susunan Dewan Penguji

Ketua : Nama Ketua Penguji dengan Gelar

Penguji 1 : Nama Penguji 1 dengan Gelar

Penguji 2 : Nama Penguji 2 dengan Gelar

Yogyakarta, Hari Bulan Tahun
Dekan Fakultas Teknologi Industri
Universitas Ahmad Dahlan

Prof. Dr. Ir. Siti Jamilatun, M.T.
NIP/NIPM. 196608121996010110784324

LEMBAR PERNYATAAN KEASLIAN

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Arvin Sultan Satria

NIM : 2200018418

Prodi : Informatika

Judul TA/Skripsi : Pengembangan Web Alat Bantu Pemrosesan Referensi Otomatis Pada

Naskah Artikel Ilmiah Berbasis NLP

Dengan ini saya menyatakan bahwa Laporan Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya/Kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, Hari Bulan Tahun

Mengetahui,
Dosen Pembimbing

Yang menyatakan,

Dr. Ardiansyah S.T., M.Cs.
NIP/NIPM. 19790723 200309 111 0932301

Arvin Sultan Satria
2200018418

KATA PENGANTAR

Tuliskan kata pengantar disini.

DAFTAR ISI

LEMBAR PERSETUJUAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN.....	iii
LEMBAR PERNYATAAN KEASLIAN	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	vii
DAFTAR TABEL.....	viii
DAFTAR KODE PROGRAM.....	ix
DAFTAR LAMPIRAN	x
DAFTAR SINGKATAN DAN ARTI LAMBANG	xi
ABSTRAK.....	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA.....	5
2.1. Kajian Penelitian Terdahulu	5
2.2. Landasan Teori	7
2.2.1. Website	7
2.2.2. Artikel Ilmiah	9
2.2.3. <i>Natural Language Processing (NLP)</i>	10
2.2.4. <i>Large Language Model (LLM)</i>	12
2.2.5. Metode Pengembangan	14
2.2.6. Metode Pengujian	16
BAB III METODOLOGI PENELITIAN	18
3.1. Pengumpulan Data	18
3.2. <i>Software dan Hardware</i>	18
3.3. Analisis.....	19
3.3.1. Identifikasi Masalah	20
3.3.2. Rumusan Masalah	20
3.3.3. Pengumpulan Data.....	20
3.3.4. Perancangan Sistem	20
3.3.5. Metode <i>Prototype</i>	22
3.3.5.1 <i>Quick Plant</i>	22
3.3.5.2 <i>Modeling Quick Design</i>	23
3.3.5.3 <i>Construction of Prototype</i>	23
3.3.5.4 <i>Deployment, Delivery & Feedback</i>	23
3.3.6. Pengujian Sistem	24
3.3.7. Hasil dan Kesimpulan	25
DAFTAR PUSTAKA.....	26
LAMPIRAN	31
Lampiran 1. Bukti Pernyataan Diterima (ACCEPTED).....	31
Lampiran 2. Bukti Terakreditasi atau Terindeks SCOPUS.....	31
Lampiran 3. Bukti Bayar (untuk jurnal yang berbayar)	31

DAFTAR GAMBAR

Gambar 2. 1 Alur <i>Natural Language Processing Pipeline</i>	11
Gambar 2. 2 Perbandingan Arsitektur LLM <i>Encoder-only</i> (BERT) dengan <i>Decoder-only</i> (GPT) ..	14
Gambar 2. 3 Metode <i>Prototype</i>	16
Gambar 3. 1 Tahapan Penelitian	19
Gambar 3. 2 Alur Kerja Sistem	21

DAFTAR TABEL

Tabel 2. 1 Perbandingan beberapa penelitian terdahulu	6
Tabel 3. 1 Skenario Pengujian <i>Black-Box Testing</i>	24

DAFTAR KODE PROGRAM

DAFTAR LAMPIRAN

LAMPIRAN	31
Lampiran 1. Bukti Pernyataan Diterima (ACCEPTED)	31
Lampiran 2. Bukti Terakreditasi atau Terindeks SCOPUS.....	31
Lampiran 3. Bukti Bayar (untuk jurnal yang berbayar)	31

DAFTAR SINGKATAN DAN ARTI LAMBANG

AI	: Artificial Intelligence
API	: Application Programming Interface
CFG	: Context-Free Grammar
FTI	: Fakultas Teknologi Industri
HTML	: HyperText Markup Language
IEEE	: Institute of Electrical and Electronics Engineers
IMRaD	: Introduction, Method, Result, and Discussion
IJAIN	: Indonesian Journal of Artificial Intelligence and Neural Network
NLP	: Natural Language Processing
LLM	: Large Language Model
PDF	: Portable Document Format
SVM	: Support Vector Machine
TTD	: Test-Driven Development
UAD	: Universitas Ahmad Dahlan
VS Code	: Visual Studio Code
WWW	: World Wide Web

ABSTRAK

Proses verifikasi referensi dalam penulisan artikel ilmiah kerap menjadi tantangan karena masih dilakukan secara manual, sehingga memerlukan waktu yang tidak sedikit dan rentan terhadap kesalahan. Setiap jurnal memiliki pedoman yang berbeda terkait jumlah, jenis, dan tahun terbit referensi, yang menambah kompleksitas bagi penulis maupun pengelola jurnal. Penelitian ini bertujuan untuk mengembangkan sebuah alat bantu berbasis *website* yang dapat memproses dan memverifikasi daftar pustaka secara otomatis dengan memanfaatkan teknologi *Natural Language Processing* (NLP). Manfaat dari sistem ini diharapkan mampu membantu penulis dalam memastikan kelayakan referensi sebelum naskah dikirimkan, serta meringankan beban administratif pengelola jurnal pada tahap pratinjau. Sistem ini secara khusus ditujukan bagi penulis (*author*) agar dapat melakukan validasi referensi secara mandiri sejak tahap awal penulisan, sebelum proses *review* dilakukan oleh jurnal.

Pengembangan sistem dilakukan menggunakan metode *prototyping* yang bersifat iteratif, dengan tahapan mulai dari identifikasi kebutuhan hingga pengujian sistem. Aplikasi dirancang berbasis web menggunakan *framework* Flask sebagai *backend* dan antarmuka pengguna yang responsif. Proses dimulai dengan mengunggah file naskah oleh pengguna, lalu sistem secara otomatis mengekstraksi daftar pustaka. Referensi yang diperoleh akan dikirim ke layanan Gemini API, dengan *prompt* terstruktur untuk mengevaluasi kesesuaian struktur, kelengkapan elemen seperti penulis, tahun, judul, serta format penulisan berdasarkan gaya selingkung seperti APA atau IEEE. Hasil analisis ditampilkan sebagai laporan umpan balik yang dapat digunakan oleh pengguna untuk melakukan perbaikan.

Evaluasi sistem dilakukan menggunakan metode *black-box testing* dengan merancang berbagai skenario pengujian, mulai dari referensi yang valid, tidak lengkap, berumur lama, hingga yang berasal dari sumber non-ilmiah. Sistem juga diuji terhadap file dengan format yang tidak didukung. Fokus evaluasi terletak pada tingkat keberhasilan (*pass rate*) dari tiap skenario untuk menilai akurasi sistem dalam memberikan umpan balik. Hasil analisis ini akan menjadi dasar untuk menilai efektivitas dan kelayakan alat bantu dalam mendukung efisiensi penulisan referensi pada naskah ilmiah.

Kata kunci: artikel ilmiah; daftar pustaka; *Natural Language Processing*; verifikasi referensi; *website*.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Jurnal ilmiah merupakan bentuk publikasi berkala yang memuat hasil-hasil penelitian ilmiah. Pengelolaan jurnal biasanya dilakukan oleh kalangan profesional, seperti asosiasi keilmuan, institusi pendidikan tinggi, atau perusahaan. Setiap jurnal umumnya menetapkan sejumlah ketentuan bagi naskah yang akan diterbitkan, mencakup aspek-aspek seperti format penulisan, jenis dan ukuran huruf, penyajian tabel dan gambar, penulisan kode program, notasi matematis, hingga penyusunan daftar pustaka [1]. Tujuan penerapan ketentuan ini adalah untuk menyesuaikan dengan gaya penulisan tertentu serta menjaga kualitas dan kredibilitas jurnal secara keseluruhan.

Salah satu hal yang sangat penting dalam naskah ilmiah yang sering menjadi perhatian utama adalah persyaratan mengenai daftar pustaka atau referensi. Daftar pustaka sendiri merupakan himpunan rujukan berupa buku, artikel, atau sumber terbitan lain yang berkaitan erat dengan isi tulisan, sehingga pembaca dapat menelusuri kembali sumber asli yang digunakan oleh penulis [2]. Dalam praktiknya, daftar pustaka tidak disusun secara sembarangan, melainkan harus memenuhi ketentuan khusus yang ditetapkan oleh pengelola jurnal. Beberapa pengelola jurnal ilmiah biasanya menetapkan ketentuan tertentu, seperti jumlah minimum referensi yang harus terdiri sebanyak 30 hingga 150 [3], [4]. Proporsi rujukan sebanyak 80% harus berupa artikel jurnal [5], serta rentang tahun penerbitan sumber yang dirujuk harus terbit dalam kurun waktu tertentu, misalnya lima tahun terakhir [6]. Aturan ini bertujuan untuk memastikan bahwa setiap naskah memiliki landasan ilmiah yang kuat dan bersumber dari referensi yang kredibel serta terkini sebelum dilakukannya peninjauan oleh penelaah (*reviewer*).

Namun, proses verifikasi referensi dengan ketentuan yang berbeda dari masing-masing jurnal seringkali menjadi tantangan bagi penulis. Setiap jurnal memiliki pedoman penulisan referensi yang berbeda, yang disusun berdasarkan standar internasional atau gaya selingkung masing-masing [7]. Perbedaan ini bukan tanpa alasan, melainkan bertujuan untuk mempermudah pencantuman, menghemat ruang tulisan, serta meningkatkan efisiensi dan kemudahan dalam penelusuran referensi yang dilakukan oleh pengelola jurnal [7]. Akibatnya, ketika naskah dialihkan ke jurnal lain, penulis perlu menyesuaikan kembali format referensinya. Penyesuaian ini umumnya dilakukan secara manual, yang tidak hanya memerlukan waktu dan usaha, tetapi juga rentan terhadap kesalahan. Tak jarang, penulis baru menyadari ketidaksesuaian referensi tersebut setelah proses pengiriman naskah atau saat menerima umpan balik dari editor.

Untuk mengatasi hal tersebut, dibutuhkan solusi yang dapat mengotomatisasi proses pengecekan referensi guna meningkatkan efisiensi dan akurasi dalam verifikasi daftar pustaka. Salah satu teknologi yang berpotensi digunakan untuk tujuan tersebut adalah pemrosesan bahasa alami (*Natural Language Processing/NLP*), yaitu teknologi yang memungkinkan komputer untuk memahami, menganalisis, menghasilkan bahasa alami manusia, serta mengolah informasi yang terkandung di dalamnya [8]. Teknologi ini juga banyak digunakan dalam aplikasi seperti analisis sentimen, sistem rekomendasi, serta ekstraksi informasi dari teks tidak terstruktur.

Seiring dengan perkembangan teknologi berbasis *web*, solusi ini dapat diimplementasikan dalam bentuk alat bantu yang memungkinkan penulis melakukan verifikasi referensi secara cepat dan akurat dengan mengintegrasikan teknologi NLP ke dalam sistem berbasis *web*. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan alat bantu berbasis *website* untuk pemrosesan dan verifikasi daftar pustaka secara otomatis pada naskah ilmiah menggunakan

metode NLP. Diharapkan, keberadaan alat ini dapat mempercepat proses pengecekan referensi dan meningkatkan kualitas naskah yang akan diterbitkan dalam jurnal ilmiah.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, diketahui bahwa proses verifikasi referensi dalam naskah ilmiah masih dilakukan secara manual yang memakan waktu, menguras tenaga, serta rentan terhadap kesalahan. Oleh karena itu, rumusan masalah penelitian ini adalah bagaimana mengembangkan *website* alat bantu pemrosesan referensi otomatis pada naskah artikel ilmiah berbasis NLP.

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan yang dicapai dalam penelitian ini sebagai berikut:

1. Mengembangkan sebuah alat bantu berbasis *website* yang dapat memproses dan memverifikasi referensi secara otomatis pada naskah artikel ilmiah. bagi penulis (*author*) sebagai pengguna utama, agar dapat memastikan kelayakan referensi sebelum naskah dikirimkan ke jurnal.
2. Menerapkan teknologi NLP untuk mengidentifikasi dan memvalidasi referensi dari naskah artikel ilmiah.

1.4. Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Memberikan alat bantu berbasis *website* bagi penulis naskah artikel ilmiah yang dapat digunakan untuk memverifikasi kelayakan referensi sebelum naskah

dikirimkan, sehingga dapat mengurangi kesalahan format dan menghemat waktu dalam proses penyesuaian referensi dengan pedoman jurnal.

2. Membantu meringankan kinerja bagi pengelola jurnal dalam pemeriksaan administratif yang berkaitan dengan kelengkapan serta mutu daftar pustaka, terutama saat tahap pratinjau sebelum proses *review*.

BAB II

TINJAUAN PUSTAKA

2.1. Kajian Penelitian Terdahulu

Dalam lima tahun terakhir, berbagai penelitian telah mengembangkan sistem otomatisasi berbasis *Natural Language Processing* (NLP) dan *Large Language Model* (LLM) untuk mempermudah proses evaluasi dan verifikasi naskah dalam publikasi ilmiah. Salah satu penelitian dilakukan oleh Farid dan Kusuma [9] yang menggunakan pendekatan berbasis struktur kalimat (sintaktik) di mana setiap kalimat diurai untuk mengenali strukturnya seperti subjek, predikat, dan objek menggunakan metode *parse tree* berbasis *Context-Free Grammar* (CFG) dengan bantuan pustaka NLTK. Dengan menggunakan model tersebut, sistem berhasil mencapai akurasi sebesar 81% dalam mengidentifikasi subjek asli dari sebuah pronomina. Hasil ini relevan karena membantu sistem dalam memahami kalimat dengan lebih baik, sehingga informasi seperti nama penulis atau tahun terbit bisa diekstrak dengan lebih akurat.

Sementara itu, Rahayu [10] dan Giglio et al. [11] menekankan pendekatan praktis melalui penggunaan alat bantu seperti Mendeley, asisten pencarian literatur seperti Elicit dan ResearchRabbit, serta model generatif seperti ChatGPT. Rahayu berfokus pada efisiensi dalam proses penulisan artikel ilmiah, sementara Giglio membahas tantangan yang dihadapi penulis *non-native*, seperti struktur dan bahasa tulisan. Keduanya menunjukkan bahwa kecerdasan buatan tidak hanya membantu secara teknis pemrosesan, tetapi juga dapat meningkatkan kualitas akhir tulisan, termasuk dalam pengelolaan referensi. Dengan pendekatan yang berbeda, Khan et al. [12] mengkaji pemanfaatan NLP untuk mendukung proses pencarian literatur ilmiah berbasis pemahaman makna teks. Mereka merancang alur kerja (*workflow*) menggunakan metode seperti *word2vec* dan *Hierarchical Dirichlet Process* (HDP) untuk menyaring literatur dan mengungkap keterkaitan tersembunyi antar dokumen. Penelitian ini

berhasil meningkatkan kualitas hasil pencarian dibandingkan pencarian berbasis kata kunci tradisional. Pendekatan ini membuka peluang pemahaman referensi yang lebih dalam, meskipun tidak secara langsung memproses daftar pustaka. Jika dibandingkan dengan penelitian Rahayu, fokus utama penelitian Khan adalah meningkatkan kesesuaian literatur yang ditemukan dengan topik penelitian, sehingga referensi yang digunakan menjadi lebih relevan dan bermakna.

Berbeda dari pendekatan sebelumnya, Pendyala et al. [13] mengembangkan sistem otomatisasi peninjauan artikel ilmiah yang menggunakan kombinasi *Machine Learning* (ML) dan model NLP. Untuk mengklasifikasikan penerimaan naskah, mereka menggunakan model-model seperti *Support Vector Machine* (SVM), *Long Short-Term Memory* (LSTM), dan jaringan saraf tiruan sekuensial (Sequential ANN). Berbagai metode diuji pada model-model ini untuk mengubah teks menjadi data numerik, mulai dari teknik klasik seperti TF-IDF dan Word2Vec, hingga metode yang lebih modern seperti BERT, FastText, dan *Large Language Model* (LLM) PaLM 2. Kombinasi yang paling unggul menggunakan DistilBERT dan berhasil mencapai akurasi hingga 86% dalam memprediksi penerimaan naskah. Selain untuk penulis, penelitian ini bertujuan mempercepat proses editorial dengan mengotomatiskan penugasan *reviewer* dan penilaian kualitas naskah. Dengan sistem yang dirancang secara lengkap, pendekatan ini menunjukkan bahwa NLP dapat menjadi alat yang efektif untuk memeriksa dan memvalidasi isi naskah ilmiah secara otomatis dan efisien.

Tabel 2. 1 Perbandingan beberapa penelitian terdahulu

Peneliti (Sitasi) *	Teknologi	Database	Hasil*
Farid dan Kusuma [9]	Pyhton (local)	NLP berbasis sintaktik, Context-Free Grammar (CFG), NLTK.	Mengenali pronomina Bahasa Indonesia dengan akurasi 81%. Fokus pada pra-pemrosesan teks, bukan verifikasi referensi.
Rahayu [10]	Web + Cloud	Analisis kualitatif terhadap alat bantu seperti Mendeley, ChatGPT.	AI terbukti meningkatkan efisiensi dan produktivitas dalam penulisan ilmiah. Fokus pada bantuan penulisan umum, bukan validasi teknis referensi.
Giglio et al. [11]	Web-based system	Elicit, ChatGPT, Grammarly	AI menjadi solusi bagi penulis non-native English untuk meningkatkan kejelasan dan gaya bahasa

Peneliti (Sitasi) *	Teknologi	Database	Hasil*
			tulisan. Fokus pada perbaikan bahasa, bukan validasi teknis
Khan et al. [12]	Python (local)	word2vec, HDP, LDA, Naive Bayes	Merancang workflow yang mengurangi overload literatur dan meningkatkan relevansi hasil pencarian. Fokus pada penemuan (discovery) literatur, bukan verifikasi (verification) kelayakan referensi.
Pendyala et al. [13]	Web + Cloud	DistilBERT, PaLM 2, SVM, Machine Learning	Peer review untuk prediksi penerimaan naskah (akurasi 86% dengan DistilBERT), skor aspek, dan rekomendasi reviewer. Pengecekan referensi tidak termasuk dalam fitur otomatisasi.
Penelitian yang dilakukan	Web + LLM	MySQL	Website untuk mengekstrak, memeriksa, dan memvalidasi kelayakan referensi (jumlah, usia, jenis) berdasarkan pedoman jurnal secara otomatis.

Berdasarkan penelitian sebelumnya yang telah dipaparkan pada Tabel 2.1 menunjukkan bahwa penelitian tentang otomatisasi penulisan ilmiah telah berkembang pesat. Beberapa penelitian sebelumnya telah berhasil mengembangkan sistem untuk aspek linguistik spesifik (Farid dan Kusuma), menyediakan bantuan penulisan umum (Rahayu; Giglio et al.), meningkatkan proses penemuan literatur (Khan et al.), hingga membangun sistem peninjauan naskah yang komprehensif (Pendyala et al.). Namun, belum ada penelitian yang secara khusus mengembangkan sistem yang mampu memproses dan memverifikasi secara otomatis daftar referensi berdasarkan pedoman kelayakan jurnal, seperti jumlah, usia, dan jenis sumber. Selain itu, proses pengecekan referensi pada penelitian-penelitian sebelumnya umumnya masih dilakukan secara manual oleh penulis atau *reviewer*. Oleh karena itu, penelitian ini menawarkan solusi berbasis *website* yang tidak hanya dapat mengekstrak dan memeriksa struktur referensi, tetapi juga dapat memvalidasinya sesuai standar teknis dan kelayakan jurnal.

2.2. Landasan Teori

2.2.1. Website

Website, atau yang juga disebut sebagai situs *web*, merupakan sekumpulan informasi yang saling terhubung dan dapat diakses melalui peramban. Informasi tersebut dapat berupa teks, gambar, animasi, video, serta berbagai jenis file digital lainnya. Umumnya, situs *web* disimpan di dalam server *web* dan dapat diakses melalui jaringan seperti internet atau jaringan lokal (LAN) dengan menggunakan alamat khusus yang disebut *Uniform Resource Locator* (URL). Kumpulan situs-situs ini yang dapat diakses secara publik dikenal sebagai *World Wide Web* (WWW), yang mentransfer data ke komputer dan menampilkannya secara visual melalui peramban agar dapat dipahami oleh pengguna. Untuk mengakses halaman-halaman dalam situs *web* tersebut, pengguna biasanya memerlukan perangkat lunak khusus yang disebut *web browser* atau peramban. [14]

Website pada umumnya dapat dikategorikan ke dalam beberapa jenis seperti yang dijelaskan oleh Ani Oktarini Sari, S.Kom, MMSI., et al. [15] yaitu sebagai berikut:

a. *Web Statis*

Merupakan jenis *website* dengan halaman yang isinya tetap dan jarang berubah. Jika ingin mengubah konten pada halaman tersebut, pengelola harus melakukannya secara manual dengan mengedit langsung kode yang membentuk struktur *website*.

b. *Web Dinamis*

Merupakan jenis *website* yang dirancang agar kontennya dapat diperbarui secara berkala. Biasanya dilengkapi dengan halaman *backend* yang memungkinkan pengelola mengubah konten dengan mudah. Contohnya: *web portal*, *web berita*, dan sebagainya.

c. *Web Interaktif*

Merupakan jenis *website* yang memungkinkan interaksi antar pengguna. Biasanya berupa forum diskusi atau blog, dengan adanya moderator yang berperan untuk mengatur jalannya diskusi.

Seiring perkembangannya, *website* tidak lagi sekadar menampilkan informasi secara statis, tetapi juga telah berkembang menjadi *platform* aplikasi interaktif yang dikenal sebagai *Web Applications*. *Platform* ini sangat cocok digunakan dalam pengembangan alat bantu untuk memproses dan memvalidasi referensi ilmiah secara otomatis, karena menawarkan berbagai keunggulan, terutama dalam hal aksesibilitas. Dengan adanya *platform* ini, pengguna tidak perlu mengunduh atau meng-*instal* aplikasi tambahan, cukup menggunakan peramban dan koneksi internet untuk dapat mengakses sistem.

Sistem berbasis *web* dirancang untuk meningkatkan efisiensi dalam pengolahan data dan dapat membantu mengurangi beban kerja serta kebutuhan sumber daya manusia[16]. Ini menunjukkan bahwa sistem berbasis *web* memiliki kemampuan untuk menyederhanakan proses akademik yang sebelumnya dilakukan secara manual. Penggunaan sistem informasi berbasis *website* juga mampu meningkatkan efisiensi dan efektivitas penyebaran informasi serta pengelolaan data akademik, yang mendukung kebutuhan aksesibilitas pengguna dalam lingkungan pendidikan [17].

Dengan berbagai kelebihan tersebut, *platform web* menjadi pilihan yang ideal untuk membangun sistem bantu akademik seperti yang dikembangkan dalam penelitian ini karena mudah diakses, ringan dijalankan, dan praktis untuk digunakan oleh penulis ilmiah dari berbagai latar belakang.

2.2.2. Artikel Ilmiah

Artikel ilmiah adalah tulisan yang berisi hasil penelitian atau kontribusi keilmuan dalam suatu bidang ilmu tertentu. Artikel ilmiah ditulis untuk memperkenalkan atau memperbaharui pengetahuan di bidang tertentu dan dipublikasikan di jurnal ilmiah atau dalam bentuk buku. Sebuah artikel ilmiah harus memenuhi standar akademik seperti metode penelitian yang valid,

hasil yang signifikan dan orisinal, serta kesimpulan yang didukung oleh bukti-bukti kuat. Selain itu, artikel ilmiah juga harus dikutip dan diperiksa secara teliti oleh para ahli sebidang sebelum dipublikasikan, dengan melalui proses *peer-review*. Artikel ilmiah umumnya memiliki struktur IMRaD (*Introduction, Methodology, Results, Discussion*) atau yang lebih mudah dipahami sebagai abstrak, pendahuluan, tinjauan pustaka, metode, hasil, pembahasan, dan kesimpulan [18].

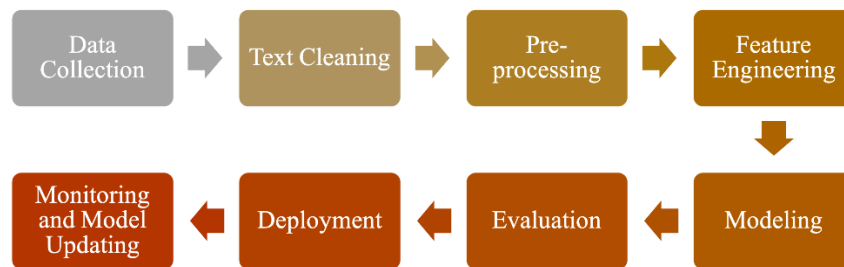
Salah satu unsur penting dalam artikel ilmiah adalah referensi, yang berfungsi untuk menunjukkan landasan teori, mendukung argumen penulis, serta menunjukkan kontribusi terhadap pengetahuan yang telah ada. Referensi ilmiah merupakan entitas bibliografis yang wajib memuat metadata penting seperti nama penulis, judul artikel, nama jurnal atau konferensi, tahun terbit, volume, dan nomor halaman. Elemen-elemen ini sangat penting agar sumber dapat ditelusuri dan diverifikasi, baik oleh pembaca maupun oleh sistem otomatis. Studi oleh Santos et al. (2022) [19] yang menganalisis lebih dari 34.000 referensi lintas-disiplin menegaskan pentingnya metadata tersebut sebagai elemen universal dalam definisi referensi ilmiah. Namun, studi yang sama juga menemukan bahwa masih banyak referensi yang tidak lengkap atau salah format, sehingga menyulitkan proses pengindeksian dan pemrosesan otomatis oleh sistem digital. Oleh karena itu, keteraturan dan konsistensi penulisan referensi menjadi salah satu faktor utama dalam menjamin kualitas sebuah artikel ilmiah.

2.2.3. *Natural Language Processing (NLP)*

Natural Language Processing (NLP) adalah cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada interaksi antara komputer dan bahasa manusia, baik dalam bentuk teks maupun suara [20]. Tujuan utama dari NLP adalah memungkinkan mesin untuk memahami, menafsirkan, dan menghasilkan bahasa manusia secara bermakna dan kontekstual, sehingga dapat diaplikasikan dalam berbagai sistem cerdas seperti penjawab pertanyaan

(*question answering*), ekstraksi informasi (*information ekstraction*), analisis sentimen (*sentiment analysis*), penerjemahan mesin (*machine translation*), pemerolehan informasi (*information retrieval*), perangkuman otomatis (*automatic summarization*), dan pengenalan wicara (*speech recognition*) [20].

Untuk mewujudkan kemampuan tersebut, NLP dijalankan melalui serangkaian proses sistematis yang dikenal dengan istilah NLP Pipeline. Pipeline ini merupakan alur kerja berurutan dalam pemrosesan bahasa alami yang bertujuan untuk mengubah input teks mentah menjadi informasi yang bisa dipahami dan diproses oleh komputer.



Gambar 2. 1 Alur *Natural Language Processing Pipeline* [21]

Penjelasan dari gambar 2.1 di atas adalah sebagai berikut:

1. *Data Collection*, merupakan tahap awal berupa pengumpulan data dalam bentuk korpus teks, dokumen, atau pertanyaan relevan yang akan digunakan untuk melatih dan menguji model NLP.
2. *Text Cleaning*, Proses pembersihan data teks dari karakter atau elemen yang tidak relevan seperti simbol khusus, tanda baca, angka yang tidak dibutuhkan, dan spasi ganda.
3. *Pre Processing*, yaitu tahapan normalisasi teks meliputi konversi huruf menjadi huruf kecil (*case folding*), penghapusan *stopword*, *stemming* atau *lemmatization*, *tokenizing*, dan *padding* jika diperlukan.

4. *Feature Engineering*, yaitu mengubah data mentah menjadi representasi numerik atau vektor fitur yang sesuai dengan kebutuhan algoritma.
5. *Modeling*, yaitu proses pelatihan model *machine learning* atau *deep learning* seperti *Bidirectional Long Short-Term Memory* (Bi-LSTM), yang memungkinkan pemahaman konteks kata dari dua arah (depan dan belakang).
6. *Evaluation*, yaitu pengujian akurasi dan performa sistem melalui metode seperti *black box testing*.
7. *Deployment*, yaitu integrasi model ke dalam platform seperti website menggunakan framework Flask.
8. *Monitoring and Model Updating*, adalah tahap pemantauan kinerja model setelah diimplementasikan ke dalam sistem. Jika ditemukan penurunan akurasi atau adanya kebutuhan baru dari pengguna, maka model dapat diperbarui menggunakan data terbaru agar hasil yang diberikan tetap akurat dan relevan.

2.2.4. Large Language Model (LLM)

Large Language Model (LLM) merupakan model kecerdasan buatan yang dirancang untuk memahami, menghasilkan, dan memanipulasi bahasa alami dalam skala besar. Model ini dilatih dengan pendekatan *self-supervised learning* pada korpus teks yang sangat besar, sehingga mampu mengembangkan pemahaman konteks secara mendalam tanpa memerlukan pelabelan manual [22]. LLM umumnya dibangun dengan arsitektur *Transformer* dan memiliki miliaran parameter, sehingga memungkinkan model ini menyelesaikan berbagai tugas pemrosesan bahasa alami (*Natural Language Processing/NLP*) secara serentak, seperti ekstraksi informasi, klasifikasi teks, hingga *summarization* dan tanya-jawab. Contoh terkemuka dari model ini adalah seri Gemini dari Google yang dapat digunakan untuk berbagai keperluan seperti meringkas teks, menjawab

pertanyaan, mengklasifikasikan data, melakukan analisis sentimen, mengekstrak data, membuat content, membuat kode program, dan melakukan percakapan [23].

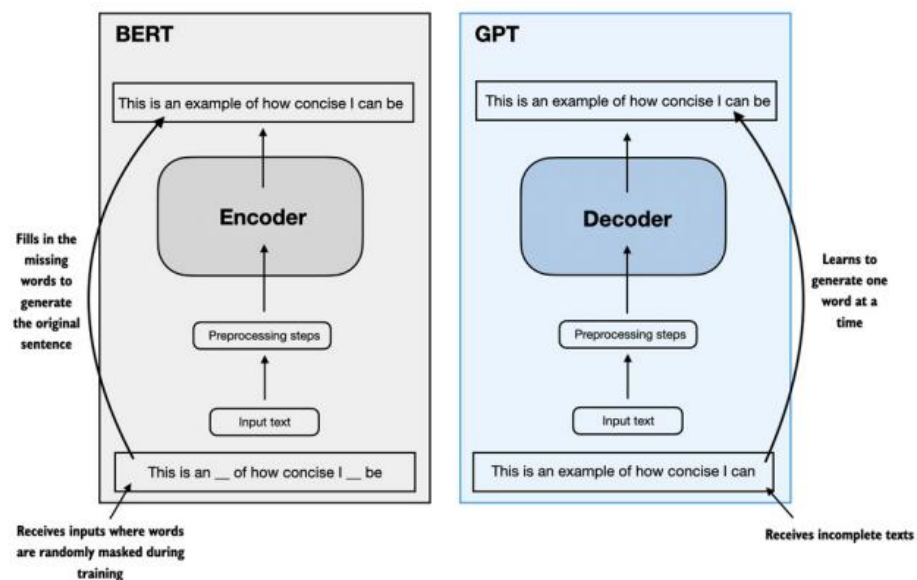
Perkembangan LLM *modern* tidak dapat dipisahkan dari arsitektur Transformer yang menjadi fondasinya. Arsitektur ini pertama kali diperkenalkan oleh Vaswani et al. pada tahun 2017 [24]. Kunci dari arsitektur ini adalah mekanisme *self-attention*, yang memungkinkan model memahami konteks dengan menimbang hubungan antar kata secara relatif dalam keseluruhan urutan *input*. Berbeda dengan RNN atau LSTM yang memproses data secara berurutan, Transformer mampu menangkap dependensi jangka panjang secara lebih efisien, sehingga menjadikannya fondasi utama dalam pengembangan LLM saat ini [25].

Seiring perkembangannya, arsitektur *Transformer* diadaptasi menjadi beberapa varian utama:

- **Arsitektur *Encoder-only*:** Model seperti BERT (*Bidirectional Encoder Representations from Transformers*) dirancang khusus untuk menyelesaikan tugas-tugas seperti klasifikasi sentimen, ekstraksi informasi, dan penjawab pertanyaan. Keunggulan BERT terletak pada kemampuannya membaca teks dari dua arah sekaligus (kiri ke kanan dan sebaliknya). Dalam proses pelatihannya, BERT belajar memprediksi kata-kata yang sengaja disamarkan dalam sebuah kalimat melalui teknik *Masked Language Modeling* [25].
- **Arsitektur *Decoder-Only*:** Model dengan arsitektur decoder-only seperti GPT (*Generative Pre-trained Transformer*) sangat efektif untuk tugas generatif, karena dirancang menghasilkan satu token dalam satu waktu berdasarkan konteks kata-kata sebelumnya. Dengan menerapkan teknik *causal attention*, model ini hanya memproses informasi dari token-token yang muncul sebelum kata yang sedang diprediksi, sehingga alur informasi tetap mengikuti urutan input. Pendekatan ini menjadi dasar dari *causal language*

modeling, yaitu strategi pelatihan untuk memprediksi kata berikutnya (*next word prediction*) yang digunakan oleh model seperti GPT [26].

Perbedaan mendasar antara kedua pendekatan ini dapat dilihat pada Gambar 2.1. Model BERT (di sisi kiri) menerima input berupa kalimat yang sebagian kata-katanya sengaja disamarkan, lalu dilatih untuk menebak kembali kata-kata yang hilang tersebut. Sebaliknya, model GPT (di sisi kanan) memanfaatkan teks sebagai konteks awal dan dilatih untuk membentuk kelanjutan kalimat secara bertahap, dan menghasilkan satu kata pada satu waktu.



Gambar 2. 2 Perbandingan Arsitektur LLM *Encoder-only* (BERT) dengan *Decoder-only* (GPT) [27]

2.2.5. Metode Pengembangan

Metode pengembangan perangkat lunak *prototyping* merupakan pendekatan yang menekankan proses iteratif secara cepat serta keterlibatan aktif antara pengembang dan pengguna. Pendekatan ini bertujuan untuk memastikan bahwa sistem yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna melalui proses perbaikan yang berkelanjutan.

berdasarkan *feedback* yang diterima selama tahapan pengembangan berlangsung [28]. Dengan adanya proses evaluasi ini, pengguna dapat memberikan masukan secara dini, sehingga pengembangan sistem selanjutnya dapat dilakukan dengan lebih terarah dan tepat sasaran.

Proses pengembangan dengan metode *prototyping* terdiri dari beberapa tahapan utama berikut ini:

1. *Communication*

Tahap awal yang melibatkan komunikasi antara pengembang dan pengguna untuk mengidentifikasi kebutuhan sistem secara umum serta memahami permasalahan yang ingin diselesaikan.

2. *Quick Plan*

Perencanaan awal dilakukan secara cepat untuk merumuskan ruang lingkup sistem serta menentukan fitur-fitur utama yang akan dikembangkan dalam bentuk prototipe.

3. *Modeling/Quick Design*

Pembuatan desain awal antarmuka atau alur kerja sistem secara sederhana dan cepat sebagai dasar dalam membangun prototipe, dengan fokus pada elemen yang dapat langsung diuji oleh pengguna.

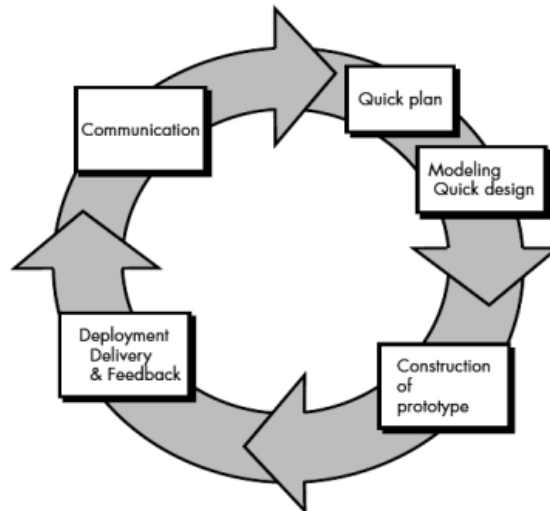
4. *Construction of Prototype*

Berdasarkan desain yang telah dibuat, prototipe awal dibangun untuk menampilkan fungsionalitas inti. Tujuannya adalah agar pengguna dapat langsung memberikan masukan, meskipun sistem belum sepenuhnya lengkap.

5. *Deployment, Delivery & Feedback*

Prototipe diserahkan kepada pengguna untuk diuji, dan pengembang mengumpulkan umpan balik terkait kegunaan, fungsi, dan tampilan sistem. Masukan ini digunakan

sebagai dasar untuk pengembangan lebih lanjut agar sistem semakin sesuai dengan kebutuhan pengguna.



Gambar 2. 3 Metode *Prototype* [29]

2.2.6. Metode Pengujian

Black-box testing atau yang juga dikenal sebagai *behavioural testing*, adalah sebuah metode pengujian perangkat lunak yang berfokus pada validasi fungsionalitas sistem tanpa melihat ke dalam struktur kode atau logika internal program. Pengujian ini dilakukan berdasarkan spesifikasi kebutuhan pengguna, bukan berdasarkan bagaimana sistem dibangun secara teknis. Pengujian *black-box testing* ini juga dilakukan sepenuhnya dari sudut pandang pengguna, sehingga proses ini tidak membutuhkan akses atau pemahaman terhadap kode sumber aplikasi. Artinya, penguji cukup memvalidasi apakah sistem merespons *input* dengan benar tanpa harus mengetahui bagaimana proses dilakukan di dalamnya [30].

Secara umum, skenario pengujian *black-box* mencakup tiga aspek utama yaitu kebutuhan fungsionalitas, kegunaan (*usability*), dan penanganan kesalahan. Pengujian fungsional bertujuan untuk memastikan bahwa setiap fitur dalam sistem berjalan sesuai dengan spesifikasi, dengan

mengevaluasi output berdasarkan input yang diberikan tanpa melihat struktur internal dari kode program [31]. Pengujian *usability* menilai kenyamanan dan kemudahan pengguna dalam menggunakan sistem, misalnya melalui observasi langsung untuk meninjau navigasi, kejelasan antarmuka, serta efektivitas interaksi pengguna [32]. Sementara itu, pengujian penanganan kesalahan dilakukan untuk melihat bagaimana sistem merespons *input* yang tidak valid, seperti kolom yang dikosongkan atau format yang salah, agar sistem tetap stabil dan mampu memberikan pesan kesalahan yang sesuai [33].

Sebagai contoh dalam penelitian ini, pengujian dilakukan dengan memasukkan berbagai jenis referensi ilmiah ke dalam sistem, seperti referensi jurnal atau buku untuk menilai apakah sistem mampu mengenali dan memvalidasi format referensi secara otomatis serta menghasilkan *output* yang sesuai dengan format yang diinginkan. Pengujian juga mencakup uji kasus ketika referensi tidak lengkap, terdapat kesalahan pada format file, dan juga referensi berasal dari sumber yang tidak kredibel, untuk memastikan bahwa sistem dapat memberikan peringatan yang akurat kepada pengguna.

BAB III

METODOLOGI PENELITIAN

3.1. Pengumpulan Data

Penelitian ini menggunakan dua jenis data, yaitu data sekunder dan data primer. Data sekunder diperoleh dari berbagai sumber terpercaya seperti jurnal ilmiah, buku, artikel konferensi, serta dokumentasi teknis NLP (misalnya spaCy dan HuggingFace). Salah satu sumber utama adalah dataset dari Scimago Journal & Country Rank (CSV) yang berisi nama jurnal, bidang keilmuan, dan peringkat quartile jurnal. Sementara itu, data primer diperoleh melalui simulasi dan pengujian langsung terhadap sistem yang dikembangkan. Peneliti menggunakan daftar pustaka artikel ilmiah yang diketik ulang atau diunggah oleh pengguna, serta melakukan observasi terhadap proses penulisan referensi. Temuan dari observasi ini dijadikan dasar untuk merancang kebutuhan dan fitur validasi dalam sistem.

3.2. Software dan Hardware

Berikut ini spesifikasi dari perangkat yang digunakan untuk penelitian:

a. Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah:

- Sistem Operasi: Windows 11
- Text Editor: Visual Studio Code & Google Colab
- Backend: Python (dengan pustaka pendukung seperti requests, transformers, dan spaCy untuk integrasi NLP dan API)
- Web Framework: Flask
- Frontend: HTML, CSS, dan JavaScript untuk membangun antarmuka pengguna yang responsif dan interaktif.

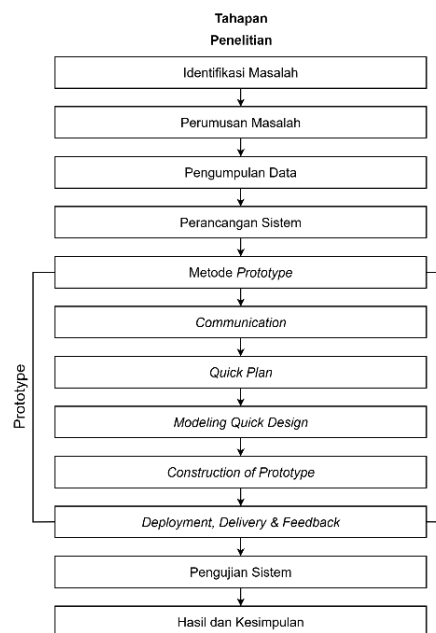
- API: Layanan Gemini API untuk analisis struktur referensi.

b. Perangkat Keras (Hardware)

Perangkat keras yang digunakan pada penelitian ini adalah personal computer dengan spesifikasi sebagai berikut:

- Manufacturer: ASUS TUF Gaming F15 FX506HC_FX506HC
- Processor: 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz (12 CPUs), ~2.7GHz
- RAM: 16 GB
- Memori: 512 GB SSD NVMe

3.3. Analisis



Gambar 3. 1 Tahapan Penelitian

Gambar di atas menunjukkan alur tahapan penelitian yang dilakukan, dimulai dari identifikasi masalah hingga tahap kesimpulan. Adapun penjabaran dari setiap tahapan sebagai berikut :

3.3.1. Identifikasi Masalah

Saat ini validasi referensi dalam artikel ilmiah masih banyak dilakukan secara manual oleh penulis maupun *reviewer*, sehingga sering terjadi ketidaksesuaian seperti jumlah referensi yang tidak mencukupi, usia referensi yang terlalu lama, atau jenis sumber yang tidak ilmiah. Hal ini menunjukkan perlunya alat bantu otomatis yang dapat memverifikasi referensi berdasarkan aturan jurnal.

3.3.2. Rumusan Masalah

Seperti yang telah dijelaskan dalam bab 1, penelitian ini berfokus pada perancangan dan pembangunan sistem berbasis web yang mampu membantu pengguna dalam melakukan validasi referensi ilmiah secara otomatis. Sistem ini dirancang untuk memudahkan proses identifikasi format referensi, menampilkan hasil validasi, serta memberikan umpan balik terkait kesalahan penulisan. Dengan adanya sistem ini, diharapkan proses penyusunan referensi menjadi lebih cepat, akurat, dan sesuai standar gaya penulisan ilmiah yang berlaku.

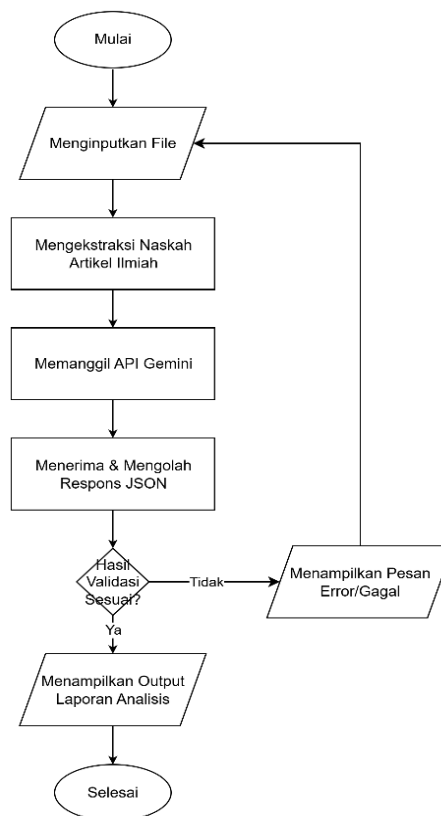
3.3.3. Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh dari sumber primer dan sekunder, sebagaimana telah dijelaskan pada subbab 3.1. Pengumpulan data ini bertujuan untuk merumuskan kebutuhan sistem, memahami karakteristik pengguna, serta menyusun acuan teknis dalam pengembangan aplikasi.

3.3.4. Perancangan Sistem

Sistem ini didesain khusus untuk mempermudah proses verifikasi referensi oleh penulis secara mandiri sebelum naskah diserahkan ke jurnal. Tidak seperti sistem yang ditujukan untuk

editor, yang dirancang untuk membantu proses editorial seperti pengelolaan banyak naskah dari berbagai penulis dan validasi administratif selama proses *review*, sistem dalam penelitian ini lebih difokuskan untuk mendukung penulis secara langsung. Sistem ini menitikberatkan pada kemudahan penulis dalam memverifikasi kualitas referensinya sejak tahap penulisan, sebelum naskah dikirimkan ke jurnal untuk melalui proses *review*.



Gambar 3. 2 Alur Kerja Sistem

Berikut ini merupakan penjelasan dari tahapan yang tergambar pada Gambar 3.2:

1. Input dan Ekstraksi Referensi

Pengguna mengunggah file naskah artikel ilmiah ke sistem. Setelah file diterima, sistem secara otomatis mengekstrak bagian daftar pustaka dari isi dokumen tersebut.

2. Analisis dan Validasi oleh API Gemini

Referensi yang telah diekstrak, kemudian dikirim ke layanan API Gemini untuk dianalisis kesesuaian struktur, elemen, dan format penulisan referensi dengan standar gaya tertentu, seperti APA atau IEEE. Setelah proses analisis dilakukan, sistem akan menerima respons dalam format JSON yang berisi hasil evaluasi tersebut. Data ini kemudian diproses lebih lanjut untuk mengidentifikasi referensi mana yang sesuai dan mana yang perlu diperbaiki.

3. *Output* dan Notifikasi Kesalahan

Jika hasil validasi menunjukkan bahwa referensi sudah sesuai, sistem akan menampilkan laporan analisis secara lengkap. Namun, jika terdapat ketidaksesuaian format, sistem akan menampilkan pesan kesalahan atau peringatan agar referensi dapat segera diperbaiki oleh pengguna.

3.3.5. **Metode *Prototype***

Metode *prototype* merupakan pendekatan pengembangan perangkat lunak yang dilakukan secara iteratif melalui pembuatan versi awal sistem. Prototipe yang dibangun akan dievaluasi langsung oleh pengguna, lalu disempurnakan berdasarkan umpan balik yang diberikan hingga sistem akhir sesuai dengan kebutuhan. Berikut tahapan-tahapan detail dari metode ini:

3.3.5.1 ***Quick Plant***

Menetapkan tujuan utama sistem, yaitu menyediakan layanan berbasis *web* untuk memvalidasi daftar pustaka secara otomatis. Kebutuhan awal yang teridentifikasi mencakup format referensi umum seperti APA dan IEEE, serta fungsionalitas inti seperti input referensi, pemeriksaan kesesuaian format, dan pemberian umpan balik otomatis. Tahap ini menjadi dasar dalam perancangan fitur-fitur sistem yang akan dikembangkan pada iterasi selanjutnya.

3.3.5.2 *Modeling Quick Design*

Pada tahap ini dilakukan perancangan awal antarmuka dan alur interaksi sistem berdasarkan kebutuhan yang telah diidentifikasi. Desain awal mencakup komponen utama seperti form input referensi, tampilan hasil validasi, serta pesan notifikasi kesalahan format. Perancangan dilakukan secara sederhana untuk mempercepat proses evaluasi dan dijadikan acuan dalam pembangunan prototipe pada tahap selanjutnya.

3.3.5.3 *Construction of Prototype*

Pada tahap ini, desain awal sistem mulai direalisasikan ke dalam bentuk prototipe menggunakan bahasa pemrograman Python dengan dukungan *web framework* Flask sebagai fondasi pengembangan aplikasi berbasis *web*. Implementasi dilakukan dengan membangun fitur inti seperti form input referensi, proses validasi format otomatis, dan tampilan hasil evaluasi. Prototipe yang dihasilkan kemudian diuji menggunakan metode *Black-box Testing* untuk menilai apakah fungsi-fungsi yang tersedia berjalan sesuai dengan kebutuhan pengguna, terutama dalam hal keakuratan validasi, tampilan antarmuka, serta kemudahan penggunaan.

3.3.5.4 *Deployment, Delivery & Feedback*

Tahap ini mencakup proses evaluasi prototipe dan pengumpulan umpan balik dari pengguna terkait fungsionalitas dan tampilan sistem. Umpan balik diperoleh melalui simulasi penggunaan sistem pemrosesan referensi, yang bertujuan untuk mengetahui apakah fitur-fitur yang tersedia telah sesuai dengan kebutuhan pengguna. Jika ditemukan kekurangan atau ketidaksesuaian dalam validasi referensi maupun alur penggunaan, maka prototipe akan disesuaikan dan disempurnakan melalui iterasi berikutnya hingga sistem memenuhi kriteria yang diharapkan.

3.3.6. Pengujian Sistem

Pengujian pada penelitian ini dilakukan dengan menggunakan metode *black-box testing*, di mana setiap fitur fungsionalitas pada sistem diuji berdasarkan input dan output yang dihasilkan, tanpa memeriksa logika internal program.

Berikut skenario pengujian *black box* yang dilakukan:

Tabel 3. 1 Skenario Pengujian *Black-Box Testing*

No.	Skenario Pengujian	Input	Ekspektasi Output	Status Uji
1.	Pengujian form input daftar pustaka	Teks referensi valid (APA Style, IEEE, dll.)	Sistem berhasil membaca, menampilkan, dan menyimpan input tanpa error	Akan diuji
2.	Pengujian validasi format referensi	Referensi tidak lengkap / tidak sesuai gaya penulisan	Sistem menandai kesalahan format dan memberikan peringatan kepada pengguna	Akan diuji
3.	Pengujian tahun terbit referensi	Referensi lebih dari 5 tahun (misalnya: tahun 2010)	Sistem memberikan peringatan bahwa referensi terlalu lama dan tidak sesuai kebijakan jurnal tertentu	Akan diuji
4.	Pengujian validasi jenis referensi (jurnal vs buku, dll.)	Referensi berasal dari blog atau situs non-ilmiah	Sistem mendeteksi dan memberikan peringatan bahwa referensi tidak valid secara akademik	Akan diuji
5.	Pengujian kuantitas referensi minimal	Referensi hanya berjumlah 3	Sistem menandai bahwa jumlah referensi belum memenuhi standar minimum jurnal (misalnya 10 atau 30)	Akan diuji
6.	Pengujian pemrosesan file referensi format .docx	Upload file dokumen berisi daftar pustaka (.docx)	Sistem berhasil mengekstrak teks referensi dan menampilkannya pada antarmuka	Akan diuji
7.	Menekan tombol "Validasi Referensi"	Klik tombol validasi setelah input	Sistem memproses referensi dan memberikan penilaian kelayakan tiap item	Akan diuji
8.	Pengujian handling kesalahan format input	Referensi ditulis acak atau tidak terbaca	Sistem menampilkan pesan error "Referensi tidak dikenali" dan tidak crash	Akan diuji
9.	Pengujian tampilan hasil validasi	Input 10 referensi campuran	Sistem menampilkan status setiap referensi (valid/tidak) dan summary hasil secara jelas	Akan diuji

10.	Pengujian ketidaksesuaian file (bukan .docx/.pdf)	Upload file .xls atau gambar	Sistem menolak file dan menampilkan peringatan "Format file tidak didukung"	Akan diuji
-----	---	------------------------------	---	------------

Beberapa metrik yang digunakan meliputi:

- Jumlah Kasus Uji: Total skenario pengujian yang telah dirancang dan dilaksanakan.
- Jumlah Kasus Uji Lulus: Banyaknya kasus uji yang memberikan hasil sesuai dengan ekspektasi.
- Jumlah Kasus Uji Gagal: Kasus uji yang hasilnya tidak sesuai dengan ekspektasi.
- Tingkat Keberhasilan (Pass Rate): Persentase keberhasilan pengujian

Untuk memberikan ukuran kuantitatif terhadap hasil pengujian, tingkat keberhasilan sering kali dihitung menggunakan rumus berikut:

$$\text{Tingkat Keberhasilan (\%)} = \left(\frac{\text{Jumlah Uji Lulus}}{\text{Jumlah Kasus Uji}} \right) \times 100\%$$

Penggunaan metode perhitungan ini membuat pengembang dan penguji dapat memperoleh gambaran kuantitatif mengenai kelayakan dan kesiapan suatu sistem sebelum digunakan oleh pengguna akhir.

3.3.7. Hasil dan Kesimpulan

Penyimpulan hasil dalam penelitian ini bertujuan untuk merangkum temuan utama dari proses pengembangan sistem pemrosesan referensi ilmiah berbasis web. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi sejauh mana sistem yang dibangun mampu membantu pengguna dalam menyusun dan memvalidasi daftar pustaka secara otomatis, serta meningkatkan efisiensi dan ketepatan penulisan referensi sesuai standar gaya ilmiah yang berlaku.

DAFTAR PUSTAKA

- [1] P. Kurniati, H. M. Putra, W. Pratomo, and A. Prakasa, "Cara Mudah Belajar Menulis Jurnal Menggunakan Referensi Otomatis Microsoft Word Dan Zotero," *P2M STKIP Siliwangi*, vol. 9, no. 1, pp. 43–53, 2022, doi: 10.22460/p2m.v9i1.3131.
- [2] K. Arisetywan, L. F. Prastiwi, E. Lika, E. K. Kiha, and B. G. Rado, "Pelatihan Pembuatan Daftar Pustaka Pada Karya Ilmiah Mahasiswa Menggunakan Ms. Word Dan Mendeley," *Community Development Journal : Jurnal Pengabdian Masyarakat*, vol. 3, no. 2, pp. 849–855, 2022, doi: 10.31004/cdj.v3i2.4917.
- [3] AJARR, "Author Guidelines," 2010. Accessed: May 06, 2025. [Online]. Available: <https://journalajarr.com/index.php/AJARR/about/submissions>
- [4] IJSR, "Structure of the Manuscript," 2023. Accessed: May 06, 2025. [Online]. Available: <https://iojs.unida.ac.id/index.php/IJSR/Author-Guidelines>
- [5] IJAIN, "Authors Guideline," 2023. Accessed: May 06, 2025. [Online]. Available: <https://ijain.org/index.php/IJAIN/about/submissions#authorGuidelines>
- [6] JTIK, "Penyerahan Naskah," 2024. Accessed: May 06, 2025. [Online]. Available: <https://jtiik.ub.ac.id/index.php/jtiik/about/submissions#authorGuidelines>
- [7] Lembaga LIPI, "Buku Pedoman Karya Tulis Ilmiah Lembaga Ilmu Pengetahuan Indonesia 2022," *Jurnal Pendidikan*, pp. 9–21, 2022.
- [8] O. N. Julianti, N. Suarna, and W. Prihartono, "Penerapan Natural Language Processing Pada Analisis Sentimen Judi Online Di Media Sosial Twitter," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, pp. 2936–2941, 2024, doi: 10.36040/jati.v8i3.9613.

- [9] M. F. Naufal and S. Ferdiana Kusuma, "NATURAL LANGUAGE PROCESSING UNTUK OTOMATISASI PENGENALAN PRONOMINA DALAM KALIMAT BAHASA INDONESIA," *JTIK*, Oct. 2022.
- [10] S. Rahayu Balai Diklat Keagamaan Ambon and J. Laksdya Leo Wattimena, "Pemanfaatan Artificial Intelligence (AI) dalam Penulisan Artikel Ilmiah," Sep. 2024.
- [11] A. Del Giglio and M. U. P. da Costa, "The use of artificial intelligence to improve the scientific writing of non-native english speakers," *Rev Assoc Med Bras*, vol. 69, no. 9, 2023, doi: 10.1590/1806-9282.20230560.
- [12] N. Khan, D. Elizondo, L. Deka, and M. A. Molina-Cabello, "Natural Language Processing Tools and Workflows for Improving Research Processes," *Applied Sciences (Switzerland)*, vol. 14, no. 24, Dec. 2024, doi: 10.3390/app142411731.
- [13] V. S. Pendyala, K. Kamdar, and K. Mulchandani, "Automated Research Review Support Using Machine Learning, Large Language Models, and Natural Language Processing," *Electronics (Switzerland)*, vol. 14, no. 2, Jan. 2025, doi: 10.3390/electronics14020256.
- [14] Imilda, O. Suryadi, and L. Ahmad, "Perancangan Sistem Informasi Penerimaan Peserta Didik Baru Dan Ujian Online Dengan Php Dan Codeigniter Di Smk Negeri 1 Sigli," *Jurnal Sistem Komputer (SISKOM)*, vol. 4, no. 2, pp. 104–114, Aug. 2024, doi: 10.35870/siskom.v4i2.816.
- [15] A. Oktarini, S. ; Ari, and A. ; Sunarti, *WEB PROGRAMMING*. 2019.
- [16] R. F. Syafariani and A. Devi, "Web-Based Academic Information System," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1757-899X/662/2/022042.

- [17] D. Prasetyo, A. Utami, and T. G. Laksana, "Website Based Academic Information System Design Using Extreme Programming Method," *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, vol. 6, no. 2, pp. 134–143, Jul. 2024, doi: 10.20895/inista.v6i2.1214.
- [18] LPPM Universitas Muhammadiyah Surakarta, "Jurnal Ilmiah: Definisi, Jenis-Jenis dan Manfaat," 2022.
- [19] E. Alves dos Santos and F. Jorge, "The way we cite: common metadata used across disciplines for defining bibliographic references," May 2022.
- [20] A. Munif, F. Ariyani, D. Khairunnisa', and R. Mardiyani, "IDENTIFIKASI KATA KUNCI PADA KONTEN PUBLIKASI JURNAL ILMIAH UNTUK STUDI KASUS PENCARIAN PUBLIKASI ONLINE ITS (POMITS)," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 21, p. 50, Jan. 2023, doi: 10.12962/j24068535.v21i1.a1.
- [21] M. Rizky Suherlan and A. Pambudi, "Jurnal Informatika Terpadu UMMIBOT SEBAGAI MEDIA LAYANAN INFORMASI PENERIMAAN MAHASISWA BARU UNIVERSITAS MUHAMMADIYAH SUKABUMI," *Jurnal Informatika Terpadu*, vol. 9, no. 2, pp. 82–91, 2023, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JIT>
- [22] S. Minaee *et al.*, "Large Language Models: A Survey," Mar. 2025, [Online]. Available: <http://arxiv.org/abs/2402.06196>
- [23] N. Rachmat and D. P. Kesuma, "Implementasi Large Language Models Gemini Pada Pengembangan Aplikasi Chatbot Berbasis Android," 2024. [Online]. Available: <https://journal.umgo.ac.id/index.php/juik/index>

- [24] A. Vaswani *et al.*, “Attention Is All You Need,” Aug. 2023, [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [25] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, “Efficient Transformers: A Survey,” *ACM Comput Surv*, vol. 55, no. 6, Jun. 2023, doi: 10.1145/3530811.
- [26] R. Wolfe and A. Caliskan, “VAST: The Valence-Assessing Semantics Test for Contextualizing Language Models,” 2022. [Online]. Available: www.aaai.org
- [27] S. Raschka, “Build a Large Language Model (From Scratch),” 2024.
- [28] A. Pismamargareta, J. R. Coyanda, and E. Yulianti, “Sistem Informasi Layanan Administrasi Desa Berbasis Web pada Kantor Desa Ulak Kemang Baru,” Jan. 2023.
- [29] R. S. Pressman, “Software Engineering: A Practitioner’s Approach,” 2010. [Online]. Available: www.mhhe.com/pressman.
- [30] S. J. Putri, D. Galih, P. Putri, W. Hayuhardhika, and N. Putra, “Analisis Komparasi pada Teknik Black Box Testing (Studi Kasus: Website Lars),” *Journal of Internet and Software Engineering*, vol. 5, no. 1, 2024.
- [31] A. D. Frayudha, I. R. Pande, and M. B. Juwita, “Implementation of Black Box Testing with the Application of Equivalence Partitioning Techniques in the M-Magazine Android Application at Semen Gresik High School,” *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 9, no. 1, pp. 134–143, Jun. 2024, doi: 10.21831/elinvo.v9i1.70382.
- [32] M. C. Júnior, D. Amalfitano, L. Garcés, A. R. Fasolino, S. A. Andrade, and M. Delamaro, “Dynamic Testing Techniques of Non-functional Requirements in Mobile Apps: A Systematic Mapping Study,” *ACM Comput Surv*, vol. 54, no. 10, Jan. 2022, doi: 10.1145/3507903.

- [33] M. E. Khan, "Different approaches to white box testing technique for finding errors," *International Journal of Software Engineering and its Applications*, vol. 5, no. 3, pp. 1–14, 2011, doi: 10.5121/ijsea.2011.2404.

LAMPIRAN

Lampiran berisi dokumen-dokumen yang digunakan sebagai pendukung penelitian, seperti dokumen pengujian. Berikut contoh penulisan lampiran untuk Skripsi Berbasis Publikasi.

Lampiran 1. Bukti Pernyataan Diterima (ACCEPTED)

Lampiran 2. Bukti Terakreditasi atau Terindeks SCOPUS

Lampiran 3. Bukti Bayar (untuk jurnal yang berbayar)