

PENERAPAN *NATURAL LANGUAGE PROCESSING* DAN *REGULAR EXPRESSIONS* DALAM VALIDASI PENULISAN ARTIKEL AINET: JURNAL INFORMATIKA

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk
Mendapatkan Gelar Sarjana Komputer (S.Kom)
Program Studi Informatika



FATIMAH AZZAHRA Z

105841100320

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2025**

PENERAPAN *NATURAL LANGUAGE PROCESSING* DAN *REGULAR EXPRESSIONS* DALAM VALIDASI PENULISAN ARTIKEL AINET: JURNAL INFORMATIKA

Diajukan sebagai Salah Satu Syarat untuk
Mendapatkan Gelar Sarjana Komputer (S.Kom)
Program Studi Informatika

Disusun dan Diajukan Oleh:

FATIMAH AZZAHRA Z

105841100320

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2025**



FAKULTAS TEKNIK

PENGESAHAN

Skripsi atas nama Fatimah Azzahra dengan nomor induk Mahasiswa 105 84 11003 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

1. Pengawas Umum

Makassar,

6 Rabi'ul Awa 1447 H

30 Agustus 2025 M

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Ahd. Rakhim Nanda, ST, MT, IPH

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T., ASEAN, Eng.

2. Penguji

a. Ketua : Prof. Dr. Ir. Hafsa Nurwana, M.T.

b. Sekretaris : Darniati, S.Kom., M.T.

3. Anggota

1. Ir. Ida, S.Kom., M.T.

2. Rizki Yuslana Bakti, S.T., M.T.

3. Lukman, S.Kom., M.T.

Mengetahui

Pembimbing I

Pembimbing II

Muhyiddin A. M. Hayat, S.Kom., M.T.

Desi Anggreani, S.Kom., M.T.

Dekan



Dr. Muh. Syarif S. Kuba, S.T., M.T.

NPM : 795288



HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **PENERAPAN NATURAL LANGUAGE PROCESSING DAN
REGULAR EXPRESIONS DALAM VALIDASI PENULISAN AINET:
JURNAL INFORMATIKA**

Nama : **FATIMAH AZZAHRA**

Stambuk : **105 84 11003 20**

Makassar, 10 September 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II

Muhyiddin A.M Hayat, S.Kom., M.T

Desi Anggreani, S.Kom., M.T

Mengetahui,

Ketua Prodi Informatika



Rizki Nurfarida Bakti, S.T., M.T.

NPM. 1307 284



MOTTO DAN PERSEMBAHAN

Motto

"Maka, sesungguhnya beserta kesulitan ada kemudahan. Sesungguhnya beserta kesulitan ada kemudahan"

(Qs. Al- Insyirah 94;5-6)

Persembahan

Dengan penuh rasa syukur ke hadirat Allah SWT, Skripsi ini saya persembahkan kepada kedua orang tua tercinta Bapak Zainuddin dan Ibu Kurnia S.Pd., M.Pd, kedua sosok yang tak pernah berhenti melimpahkan kasih sayang, doa tulus, serta dukungan yang tiada henti. Untuk Adik tercinta, Fakhira Zazkia dan Fathin Nur Athifa dan keluarga besar yang selalu memberikan semangat, serta para dosen yang telah membimbing dan menuntun saya selama menempuh pendidikan. Ucapan terima kasih juga saya tunjukkan kepada sahabat-sahabat seperjuangan yang senantiasa hadir memberi warna dalam perjalanan ini. Untuk keluarga besar informatika 20 khususnya kelas A, terima kasih atas semangat dan perjalanan yang dijalani bersama. Untuk keluarga besar Integrasi angkatan 20, terima kasih atas kebersamaan, dukungan, bantuan, serta semangat yang selalu diberikan di kala suka maupun duka. terima kasih kepada seluruh pihak yang tidak dapat disebutkan satu per satu, atas segala dukungan dan doa yang tulus. Tak lupa, terima kasih kepada seluruh pihak yang tidak dapat disebutkan satu per satu, atas segala dukungan dan doa yang tulus. Dan akhir, untuk diriku sendiri, terima kasih telah bertahan, terus belajar, dan berjuang hingga titik ini, percaya bahwa setiap niat baik dan harapan akan selalu diberikan kemudahan oleh-Nya. Semoga persembahan ini bermanfaat bagi pengembangan ilmu pengetahuan dan menjadi bagian dari amal kebaikan. Selamat bergelar S.Kom.

ABSTRAK

FATIMAH AZZAHRA, Penerapan Natural Language Processing dalam Validasi Penulisan Artikel Jurnal *AINET: Jurnal Informatika* (dibimbing oleh Desi Anggreani S.Kom., M.T dan Muhyiddin A.M. Hayat, S.Kom., M.T).

Penelitian ini mengimplementasikan *Natural Language Processing* (NLP) dengan pendekatan *rule-based* serta *Regular Expressions* (Regex) untuk melakukan validasi aturan penulisan artikel ilmiah pada *AINET: Jurnal Informatika*. Sistem yang dibangun berupa aplikasi web interaktif yang dapat memeriksa kepatuhan artikel terhadap format jurnal, meliputi judul, penulis, abstrak, kata kunci, bagian wajib, serta daftar pustaka. Validasi dilakukan melalui kombinasi preprocessing teks, segmentasi dokumen, serta pencocokan pola dengan Regex. Evaluasi sistem menggunakan metode *black-box testing* terhadap 20 artikel menunjukkan akurasi 90%, dengan 18 artikel dikenali secara tepat sesuai kondisi aktual. Hasil ini menunjukkan bahwa integrasi NLP dan Regex efektif untuk mendukung proses validasi penulisan artikel, meskipun masih diperlukan peningkatan pada deteksi abstrak bilingual, konsistensi bahasa pada heading, serta format referensi.

Kata Kunci: Natural Language Processing, Regular Expressions, Validasi Penulisan, Jurnal Ilmiah, *AINET: Jurnal Informatika*

ABSTRACT

FATIMAH AZZAHRA, *Implementation of Natural Language Processing in the Validation of AINET Journal Article Writing (supervised by Desi Anggreani S.Kom., M.T dan Muhyiddin A.M. Hayat, S.Kom., M.T).*

This study implements *Natural Language Processing* (NLP) with a *rule-based* approach and *Regular Expressions* (Regex) to validate the formatting rules of scientific articles in *AINET: Jurnal Informatika*. The developed system is an interactive web application designed to verify compliance with journal standards, including title, authors, abstract, keywords, mandatory sections, and references. Validation was conducted by combining text preprocessing, document segmentation, and pattern matching using Regex. System evaluation through *black-box testing* on 20 articles achieved 90% accuracy, with 18 articles correctly classified according to their actual condition. These results indicate that integrating NLP and Regex is effective for supporting the validation process of scientific articles, although improvements are still needed in bilingual abstract detection, heading language consistency, and reference formatting.

Keywords: Natural Language Processing, Regular Expressions, Writing Validation, Scientific Journal, AINET: Jurnal Informatika

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu 'alaikum warahmatullahi wabarakatuh,

Alhamdulillah rabbil 'alamin, senantiasa penulis panjatkan rasa syukur kepada Sang Pencipta, Allah Subhanu Wa Ta'ala, atas limpahkan rahmat, ridha, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “PENERAPAN NATURAL LANGUAGE PROCESSING DAN REGULAR EXPRESSION DALAM VALIDASI PENULISAN ARTIKEL JURNAL *AINET: Jurnal Informatika*” Shalawat dan salam penulis panjatkan kepada Nabi Besar Muhammad Shallallahu 'alaihi wa sallam, sebagai *ustawun hasanah dan rahmatan lil'alamin*, yang telah membawa umat manusia dari zaman jahiliah menuju era modern seperti yang kita rasakan saat ini.

Selama proses penyusunan skripsi ini, penulis menerima banyak bimbingan, arahan, motivasi, dan bantuan dari berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak **Dr. Ir. H. Abd. Rakhim Nanda, S.T., M.T., IPU**, selaku Rektor Universitas Muhammadiyah Makassar
2. Bapak **Muh. Syafa'at S Kuba, S.T., M.T.**, selaku Dekan Fakultas Teknik Universitas Muhammadiyah Makassar
3. Ibu **Rizky Yusliana Bakti, S.T., M.T.**, selaku Ketua Program Studi Informatika Universitas Muhammadiyah Makassar
4. Bapak **Muhyiddin AM Hayat S.Kom., M.T.**, selaku Dosen Pembimbing 1 yang telah memberikan arahan dan bimbingan serta saran yang sangat berarti dalam penyusunan skripsi
5. Ibu **Desi Anggraeni S.Kom., M.T.**, selaku Dosen Pembimbing 2 yang telah memberi arahan dan bimbingan serta saran yang sangat berarti dalam penyusunan skripsi.
6. Bapak **Muhyiddin AM Hayat S.Kom., M.T.**, selaku Dosen Pembimbing Akademik yang selama ini turut serta membantu dan memberikan arahan.
7. Seluruh Dosen Fakultas Teknik Program Studi Informatika Universitas Muhammadiyah Makassar yang telah memberikan ilmu dan bantuannya serta dorongannya dalam penulisan skripsi ini.

8. Kepada Kedua Orang Tua saya tercinta, Bapak Zainuddin dan Ibu Kurnia S.Pd., M.Pd, persembahkan terima kasih yang tak terhingga atas segala pengorbanan, kasih sayang, dukungan baik berupa moral, materi, dan spiritual yang senantiasa diberikan kepada penulis.
9. Senior sejurusan saya Kak **Andi Agung Dwi Arya, S.Kom**, atas bantuannya sejak awal saya masuk ke jurusan ini.
10. Teman – teman Integrasi khususnya Angkatan 2020 Fakultas Teknik, Universitas Muhammadiyah Makassar, terimakasih yang sebesar – besarnya atas dukungan dan doanya.
11. Teman – teman saya tercinta di kelas A Angkatan 2020 Program Studi Informatika Universitas Muhammadiyah Makassar.
12. Kepada semua pihak yang tidak bisa penulis tuliskan satu persatu, penulis mengucapkan terima kasih yang sebesar – besarnya.
13. Dan untuk diri sendiri, terima kasih sudah berjuang dan berusaha saba dan tidak menyerah sejauh ini meskipun terlambat bukan berarti gagal.

Semoga kebaikan menjadi Amal Sholeh dan di balas dengan kebaikan pula oleh Allah Subhanahu Wata'ala Aamiin ya Rabbal Alamin. Demikian laporan Skripsi ini, dan penulis sadar bahwa masih banyak kekurangan didalam laporan ini oleh karna itu penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca atas laporan ini. Akhir kata penulis ucapkan terima kasih

Billahi fisabililhaq, fastabiqul khairat.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Makassar, 10 september 2025

Penulis

Fatimah Azzahra Z

DAFTAR ISI

MOTTO DAN PERSEMBAHAN	v
ABSTRAK	vi
ABSTRACT	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR ISTILAH	xiii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah	2
C. Tujuan Penelitian	2
D. Manfaat Penelitian	2
E. Ruang Lingkup Penelitian	3
F. Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	4
A. Landasan Teori	4
B. Penelitian Terkait	9
C. Kerangka Pikir	11
BAB III METODE PENELITIAN	12
A. Tempat dan Waktu Penelitian	12
B. Alat Dan Bahan	12
C. Perancangan Sistem	13
BAB IV HASIL DAN PEMBAHASAN	18
A. Deskripsi Umum Implementasi Sistem	18
B. Hasil Implementasi Sistem	19
C. Hasil Validasi dengan Metode NLP dan REGEX	28
D. Pengujian Sistem	34
BAB V KESIMPULAN DAN SARAN	39
A. Kesimpulan	39
B. Saran	39
DAFTAR PUSTAKA	40
Lampiran	43

DAFTAR GAMBAR

Gambar 1. Perancangan Sistem.....	5
Gambar 2. AINET: Jurnal Informatika.....	8
Gambar 3. Aturan Penulisan Pada AINET: Jurnal Informatika	9
Gambar 4. Kerangka Pikir.....	11
Gambar 5. Perancangan Sistem.....	13
Gambar 6. Flowchart nlp rule-based	14
Gambar 7. Desain antar muka sistem	20
Gambar 8. Unggah template AINET: Jurnal Informatika	21
Gambar 9. Jurnal submission.....	22
Gambar 10. Unggah dokumen	22
Gambar 11. Hasil validasi.	23
Gambar 12. Proses validasi nlp	30
Gambar 13. Validasi regex	32

DAFTAR TABEL

Tabel 1. Penelitian terkait	9
Tabel 2. Template Ainet	25
Tabel 3. Validasi Ainet	26
Tabel 4 Hasil Validasi NLP Artikel	29
Tabel 5 Hasil Validasi Regex Heading Artikel	31
Tabel 6. Ringkasan perbedaan	34
Tabel 7. Black Box Testing	35



DAFTAR ISTILAH

<i>NLP (Natural Language Processing)</i>	Cabang ilmu komputer yang menggabungkan kecerdasan buatan dan linguistik untuk memungkinkan komputer memahami, memproses, dan menganalisis bahasa manusia.
<i>Regex (Regular Expressions)</i>	Pola teks yang digunakan untuk pencocokan string, pencarian, dan validasi format penulisan (misalnya email, ISSN, referensi).
<i>Rule-Based</i>	Pendekatan berbasis aturan yang menggunakan seperangkat pola linguistik atau logika manual untuk menganalisis teks.
<i>AINET: Jurnal Informatika</i>	Proses dekomposisi data untuk Jurnal ilmiah yang diterbitkan oleh Program Studi Informatika Universitas Muhammadiyah Makassar, berfokus pada bidang informatika dan teknologi informasi.
<i>Preprocessing Teks</i>	Tahapan pembersihan dan persiapan teks sebelum diproses lebih lanjut, meliputi lowercasing, tokenisasi, stopword removal, dan stemming.
<i>Tokenisasi</i>	Proses memecah teks menjadi unit-unit kecil seperti kata atau kalimat.
<i>Stopword Removal</i>	Proses menghapus kata-kata umum yang tidak memiliki makna penting (misalnya: yang, dan, di).
<i>Stemming</i>	Proses mengembalikan kata ke bentuk dasarnya (contoh: "berlari" menjadi "lari").
<i>TF-IDF (Term Frequency-Inverse Document Frequency)</i>	Metode pengukuran pentingnya suatu kata dalam dokumen dengan melihat frekuensi kemunculannya dibandingkan seluruh dokumen.
<i>Black-Box Testing</i>	Metode pengujian perangkat lunak yang berfokus pada masukan dan keluaran sistem tanpa memperhatikan struktur internal program.
<i>Accuracy (Akurasi)</i>	Ukuran tingkat ketepatan sistem dalam melakukan prediksi yang benar.
<i>True Positive (TP)</i>	Kondisi di mana artikel yang valid dikenali benar sebagai valid.
<i>True Negative (TN)</i>	Kondisi di mana artikel yang tidak valid dikenali benar

False Positive (FP)

sebagai tidak valid

Kondisi di mana artikel yang tidak valid salah dikenali sebagai valid

False Negative (FN)

Kondisi di mana artikel yang valid salah dikenali sebagai tidak valid.



BAB I PENDAHULUAN

A. Latar Belakang

Penulisan ilmiah merupakan elemen krusial dalam pendidikan yang bertujuan untuk meningkatkan akreditasi institusi dan mengembangkan keterampilan berpikir kritis mahasiswa (Riana dkk., 2024). Jurnal ilmiah merupakan media utama untuk menyajikan hasil penelitian secara metodis, terstruktur, dan jelas (Lazuardi, 2023). Di tengah globalisasi dan kemudahan akses informasi saat ini, akademisi tidak hanya mampu menghasilkan karya ilmiah, tetapi juga mampu menganalisis dan menerapkan informasi secara efektif (Susanti dkk., 2023).

Dalam pengembangan sistem berbasis NLP, metode *rule-based* menjadi salah satu pendekatan yang relevan dan efektif. Metode ini menggunakan seperangkat aturan linguistik atau pola struktural teks yang ditentukan secara manual dan sangat cocok diterapkan pada domain dengan format tetap, seperti penulisan jurnal ilmiah. Penelitian oleh Priandana et al. (2020) menunjukkan bahwa pendekatan *rule-based* berhasil mengekstraksi informasi penting dari laporan abu vulkanik dengan akurasi mencapai 84%, membuktikan bahwa metode ini efektif dalam menangani teks yang memiliki pola struktur yang konsisten.

Selain itu, Ekspresi Reguler memiliki peran penting dalam memvalidasi format penulisan. REGEX adalah alat penyunting string yang dapat secara akurat menginterpretasikan format seperti email, ISSN, struktur judul, dan elemen jurnal lainnya (Muhammad Isfa Hany dkk., 2021). Meskipun penyusunan regex manual dapat menghemat waktu dan biaya terkait masalah, penggunaannya sangat efektif dalam mengidentifikasi teks yang tidak biasa dan tidak menentu.

Dengan menggabungkan kemampuan pemrosesan bahasa alami (NLP) berbasis aturan dan ekspresi reguler (REGEX) untuk mendeteksi format teks, sistem validasi otomatis dapat dikembangkan yang tidak hanya akurat tetapi juga efisien dalam menilai struktur artikel jurnal ilmiah. Sistem ini diharapkan dapat membantu mahasiswa, akademisi, dan editor jurnal untuk memastikan bahwa karya yang dihasilkan memenuhi persyaratan dan gaya jurnal, serta meningkatkan jumlah publikasi ilmiah secara keseluruhan.

B. Rumusan Masalah

1. Bagaimana implementasi metode *Natural Language Processing* dan *Regular Expression* dengan pendekatan *rule-based* dapat digunakan untuk mengenali memvalidasi penulisan artikel *AINET: Jurnal Informatika*
2. Bagaimana analisis pengujian metode *Natural Language Processing* dan *Regular Expression* dengan pendekatan *rule-based* dapat digunakan untuk mengenali memvalidasi penulisan artikel *AINET: Jurnal Informatika*

C. Tujuan Penelitian

Mengimplementasikan metode *Natural Language Processing* dan *Regular Expression* dengan pendekatan *rule-based* guna mengenali dan memvalidasi kesesuaian penulisan jurnal ilmiah terhadap aturan baku yang telah ditetapkan di *AINET: Jurnal Informatika*.

D. Manfaat Penelitian

Penelitian tentang penerapan *Natural Language Processing* dan *Regular Expressions* dalam Validasi Penulisan Jurnal *AINET: Jurnal Informatika* ini diharapkan dapat bermanfaat baik secara teoritis maupun praktis.

1. Secara Teoris

- a. Untuk pengembangan ilmu pengetahuan, terutama bagi dunia Teknik Informatika dan Ilmu Komputer.

2. Secara Praktis

a. Bagi Peneliti

- 1) Memahami cara kerja metode *Natural Language Processing* dan *Regular Expression*

b. Bagi Mahasiswa

- 1) Mempermudah dan membantu mahasiswa dalam proses validasi aturan penulisan jurnal secara otomatis dan efisien.

c. Bagi Universitas

- 1) Sebagai bahan referensi untuk penelitian yang akan datang.

E. Ruang Lingkup Penelitian

Penelitian ini berfokus pada pengembangan sistem validasi penulisan *AINET: Jurnal Informatika* dengan menggunakan teknologi *Natural Language Processing* dan *Regular Expressions*.

1. Penelitian ini lebih berfokus pada metode *Natural Language Processing* untuk menganalisis teks jurnal dan mengekstrak informasi penting.
2. Penelitian ini hanya fokus pada validasi penulisan jurnal artikel dengan format teks, penelitian ini tidak mencakup validasi penulisan *AINET: Jurnal Informatika* dengan format multimedia (contoh: gambar, tabel) dan format PDF
3. Data yang digunakan yaitu jurnal prodi informatika
4. Template jurnal yang digunakan adalah template *AINET: Jurnal Informatika*

F. Sistematika Penulisan

Secara garis besar penulisan laporan tugas akhir ini terbagi menjadi beberapa bagian bab yang tersusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang teori – teori yang saling berhubungan dengan penelitian seperti metode dan algoritma yang melandasi penulis dalam melaksanakan skripsi.

BAB III METODE PENELITIAN

Bab ini Menguraikan metode penelitian yang digunakan, mulai dari tempat dan waktu penelitian, data, alat dan bahan, perancangan sistem, perancangan model, teknik pengujian, hingga teknik analisis data.

BAB IV HASIL DAN PEMBAHASAN

Menyajikan hasil penelitian, evaluasi model, visualisasi, dan analisis hasil.

BAB V PENUTUP

Memuat kesimpulan yang diambil dari penelitian serta saran yang dapat diberikan untuk penelitian selanjutnya

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Teknologi

Teknologi merupakan sarana yang diciptakan manusia untuk mempermudah aktivitas sehari-hari, baik di bidang pendidikan maupun pekerjaan. Dalam perkembangannya, teknologi tidak hanya sebatas perangkat keras, tetapi juga mencakup perangkat lunak dan aplikasi yang mendukung berbagai kegiatan, termasuk pembelajaran. Perubahan digital saat ini memungkinkan institusi pendidikan menggunakan teknologi sebagai media pendukung, baik untuk mengakses informasi maupun sebagai alat bantu dalam proses belajar-mengajar (Maritsa dkk., 2021; Nurillahwaty, 2021).

2. *Natural Language Processing*

NLP adalah bidang interdisipliner yang menggabungkan kecerdasan buatan dan linguistik dengan tujuan agar komputer mampu memahami serta memproses bahasa manusia. Penerapannya dapat berupa analisis teks seperti klasifikasi dokumen, ekstraksi informasi, dan analisis sentimen, maupun berbasis percakapan seperti chatbot dan asisten virtual. Tujuan utamanya adalah memungkinkan interaksi manusia-komputer menggunakan bahasa alami yang dapat dimengerti oleh mesin (Rumapea, 2021; Yuneфри dkk., 2021; Ningtyas dkk., 2023).

3. *Regular Expressions*

Regular Expressions merupakan teknik klasik yang dipakai untuk mencocokkan pola dalam teks. Alat ini sangat efisien untuk proses pencarian string, ekstraksi data, serta validasi format seperti email, ISSN, atau penulisan referensi. Meski penyusunan pola regex secara manual dapat menyita waktu, metode ini terbukti efektif dalam menemukan kesalahan penulisan yang bersifat struktural. Penelitian terbaru bahkan mengembangkan otomatisasi regex agar lebih cepat dan minim kesalahan (Muhammad Isfa Hany dkk., 2021; Irham Gufroni dkk., 2023).

4. Metode *Rule-Based* dalam NLP

Metode *Rule-Based* adalah pendekatan dalam Pemrosesan Bahasa Alami (NLP) yang menggunakan seperangkat aturan linguistik atau pola spesifik yang dirancang secara manual untuk menganalisis atau mengekstrak informasi dari teks. Aturan-aturan ini dapat mencakup struktur kalimat, urutan kata, tanda baca, dan format penulisan (Priandana dkk., 2020).

Dalam konteks penelitian yang dilakukan oleh Priandana dkk. (2020), pendekatan *Rule-based* diterapkan untuk mengekstrak informasi dari teks laporan BMKG SIGMET. Sistem ini mencapai akurasi 84% dalam mengidentifikasi informasi yang relevan, menunjukkan bahwa metode NLP *rule-based* ini efektif dalam domain dengan format atau struktur data yang konsisten.



Gambar 1. Perancangan Sistem

Untuk memudahkan proses perancangan penelitian ini menggunakan flowchart perancangan sistem sebagai berikut :

Proses dimulai pada tahap Mulai, di mana sistem siap menerima *input* dokumen dari pengguna. Selanjutnya, pengguna mengunggah file jurnal yang akan divalidasi melalui tahap *Upload File*. Setelah file berhasil diunggah, sistem

melakukan *Ekstraksi Teks* untuk mengambil seluruh isi teks dari dokumen, baik yang berformat .docx, .pdf, maupun format teks lainnya.

Tahap berikutnya adalah *Preprocessing NLP*, yaitu proses pembersihan dan normalisasi teks, seperti menghapus karakter yang tidak relevan, mengubah huruf menjadi *lowercase*, melakukan *tokenisasi*, menghapus *stopwords*, serta melakukan *stemming* atau *lemmatization* agar teks siap dianalisis. Setelah itu, sistem memasuki tahap *Regular Expressions*, yaitu pencocokan pola teks berdasarkan aturan penulisan jurnal, misalnya format sitasi, daftar pustaka, penomoran tabel, atau penulisan judul.

Hasil dari proses ini ditampilkan pada tahap Hasil Skor Validasi, berupa persentase kesesuaian dokumen terhadap standar penulisan, disertai informasi kesalahan yang ditemukan. Tahap terakhir adalah Selesai, di mana sistem menyajikan laporan validasi sehingga pengguna dapat melakukan revisi dokumen sesuai rekomendasi yang diberikan.

5. Jurnal

Jurnal merupakan salah satu bentuk publikasi ilmiah yang digunakan untuk menyebarkan hasil penelitian secara cepat dan tepat waktu, sehingga menjadikannya sumber referensi utama dalam dunia akademik (Zakiah & Rohani, 2023). Umumnya, jurnal memuat artikel ilmiah yang disusun secara sistematis, meliputi bagian-bagian seperti abstrak, metode, hasil, dan simpulan (Karomah & Rukmana, 2022).

Dengan kemajuan teknologi, jurnal ilmiah kini banyak diterbitkan dalam bentuk elektronik (e-journal), yang lebih efisien dan mudah diakses dibandingkan jurnal cetak (Society dkk., 2023). Jurnal dikategorikan menjadi jurnal bereputasi nasional dan internasional, masing-masing dengan standar dan sistem pengindeksannya sendiri, seperti SINTA untuk jurnal nasional dan Scopus untuk jurnal internasional.

Publikasi di jurnal bereputasi mencerminkan kompetensi akademik penulis. Namun, proses ini menghadapi tantangan, karena penulis harus memenuhi berbagai persyaratan terkait struktur penulisan, gaya bahasa, dan kontribusi ilmiah. Oleh karena itu, pelatihan dan sistem pendukung sangat penting untuk meningkatkan kualitas karya ilmiah (Firmansyah dkk., 2020)

6. *AINET*: Jurnal Informatika

AINET: Jurnal Informatika adalah jurnal ilmiah yang diterbitkan oleh Program Studi Informatika, Universitas Muhammadiyah Makassar, yang berfokus pada bidang informatika dan teknologi informasi. Jurnal ini berfungsi sebagai wadah publikasi bagi para peneliti, dosen, mahasiswa, dan praktisi untuk menyebarluaskan hasil penelitian, kajian ilmiah, serta pengembangan teknologi terbaru. Ruang lingkup jurnal ini mencakup berbagai topik, seperti pemrosesan bahasa alami (*Natural Language Processing*), kecerdasan buatan (*Artificial Intelligence*), data mining, pengolahan citra digital, sistem informasi, keamanan jaringan, *Internet of Things (IoT)*, serta pengembangan aplikasi berbasis *web* dan *mobile*.

AINET: Jurnal Informatika bersifat *open access*, yang berarti seluruh artikel yang diterbitkan dapat diakses secara gratis oleh semua orang. Ruang lingkup artikel yang dimuat di *AINET*: Jurnal Informatika mencakup berbagai bidang dalam ilmu komputer dan teknologi informasi, antara lain *computer vision*, pemrosesan bahasa alami (NLP), data mining, kecerdasan buatan, pemrosesan sinyal digital, robotika, sistem cerdas, antarmuka manusia-mesin, teori informasi, tata kelola TI, teknologi jaringan, serta proses bisnis berbasis IT.

Jurnal ini juga memegang teguh prinsip-prinsip etika publikasi ilmiah, termasuk keharusan *orisinalitas* karya, larangan *plagiarisme*, transparansi kontribusi penulis, serta penanganan konflik kepentingan. Semua artikel dapat diakses dan diunduh secara gratis oleh semua orang tanpa memerlukan langganan, selama digunakan untuk kepentingan *non-komersial* dan tetap mencantumkan sumber.



Gambar 2. AINET: Jurnal Informatika

7. Aturan Penulisan AINET: Jurnal Informatika

Penulisan akademis berperan penting dalam menyebarkan ilmu pengetahuan, sehingga kualitasnya sangat bergantung pada isi dan keakuratan penyampaiannya. Salah satu aspek penting dalam penulisan akademis adalah penggunaan tanda baca yang tepat, karena kesalahan dalam penggunaannya dapat menyebabkan kebingungan dan mengaburkan makna (Harahap dkk., 2025).

Kesalahan berbahasa seringkali muncul akibat kurangnya pemahaman penulis terhadap kaidah-kaidah kebahasaan. Bahkan penulis yang dianggap mahir pun dapat melakukan kesalahan, baik dalam morfologi, semantik, maupun sintaksis. Hal ini juga sering ditemukan dalam karya ilmiah seperti tesis, di mana kesalahan yang umum terjadi meliputi penggunaan kata baku, partikel, dan struktur kalimat yang tidak efektif (Untung dkk., 2024).

**Judul Naskah Publikasi Maksimum 15 Kata dalam
Bahasa Indonesia, Huruf Kapital di Awal Kata
(Center, Arial 14 pt, Bold)**

Penulis Pertama¹, Penulis kedua², Penulis ketiga³ (Arial Bold 10pt)

¹Departemen, Institusi, kota, kodepos, negara (Arial 9 pt italic)

²Departemen, Institusi, kota, kodepos, negara (Arial 9 pt italic)

e-mail koresponden: e-mail³ (Arial 9 pt italic)

Received: Month XX, 20XX, Accepted: Month XX, 20XX, Published: Month XX, 20XX (Arial 9 pt)

Abstrak

Abstrak terdiri dari 150-250 kata, kata berbahasa Indonesia dicetak miring dengan Arial 10pt, spasi 1.0. Abstrak harus jelas, deskriptif dan harus memberikan gambaran singkat masalah yang diteliti. Abstrak meliputi tujuan penelitian, metode yang dipakai untuk menyelesaikan masalah, dampak dan hasil penelitian. Abstrak sebaiknya diakhiri dengan komentar tentang pentingnya hasil atau kesimpulan singkat. (Arial 10pt italic)

Kata kunci: 3-5 Kata Kunci, Algoritma, Pengolahan Gambar

Abstract

Abstract is written in English in maximum of 150-250 words in italics with Times New Roman 10 point, 1.0 space. Abstract should be clear, descriptive, and should provide the aim of research, the method how to solve the problem, and the result(s). Abstract preferably end with a comment about the importance of the results or conclusions briefly. (Arial 10pt italic)

Gambar 3. Aturan Penulisan Pada AINET: Jurnal Informatika

B. Penelitian Terkait

Peneliti memperoleh banyak inspirasi dan referensi untuk penyusunan skripsi ini dari penelitian sebelumnya, terkait dengan latar belakang masalah pada skripsi ini. Penelitian sebelumnya yang terkait meliputi

Tabel 1. Penelitian terkait

Peneliti	Tujuan/Kasus	Metode/Algoritma	Hasil
Cano, et.al., (2024)	Meninjau pemanfaatan NLP pada grammar checker akademik	etode <i>systematic literature review</i> dengan fokus pada kombinasi <i>rule-based NLP</i> dan <i>machine learning</i>	Sistem efektif mendeteksi kesalahan format, rule-based unggul dalam mendeteksi struktur sesuai pedoman jurnal

Peneliti	Tujuan/Kasus	Metode/Algoritma	Hasil
Helen, Pradana, & Afif (2024)	Mengonversi format referensi otomatis ke gaya APA/IEEE/Chicago	<i>Rule-Based Machine Learning dengan Regular Expressions (Regex)</i>	Tingkat kesesuaian 97,8%, sistem mampu memproses ratusan referensi dengan konsistensi tinggi
Ladas et al. (2023)	Peningkatan keterbacaan & akurasi pipeline rule-based NLP	<i>Modularisasi aturan menggunakan pustaka NLP seperti spaCy, UIMA, dan CoreNLP</i>	Akurasi ekstraksi meningkat 10–15%, pemeliharaan lebih mudah
Irham Gufroni et al., 2023	Peningkatan akurasi klasifikasi opini Twitter	<i>Naïve Bayes Classifier</i>	Akurasi naik dari 79,26% → 88,05% setelah menggunakan regex
Ismail et al. (2022)	Otomatisasi pengecekan regulasi kompleks	<i>Kombinasi NLP, Regular Expressions, dan Context-Free Grammar (CFG)</i>	Akurasi pemeriksaan dokumen mencapai 93%; mampu menangani aturan yang kompleks dan mengurangi waktu pemeriksaan dari jam menjadi menit
Rahman, M et.al(2022)	Ekstraksi data medis PSG	Metode rule-based NLP dengan regex	Akurasi >90% untuk parameter tidur pada catatan EMR
Irkham Huda (2021)	Untuk mengembangkan dan mengimplementasikan Sistem pencarian lokasi berbasis NLP	Natural Language Processing Based Map	Presisi 0,95 dan recall 0,91

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian dilakukan di Universitas Muhammadiyah Makassar yang beralamat di Jl. Sultan Alauddin No. 259. Proses penelitian dimulai pada bulan juli 2025 sampai semua proses pengumpulan data selesai. Selama rentang waktu tersebut, penelitian ini akan melalui tahap pengumpulan data, pengolahan data, pelatihan model, pengujian, serta analisis data..

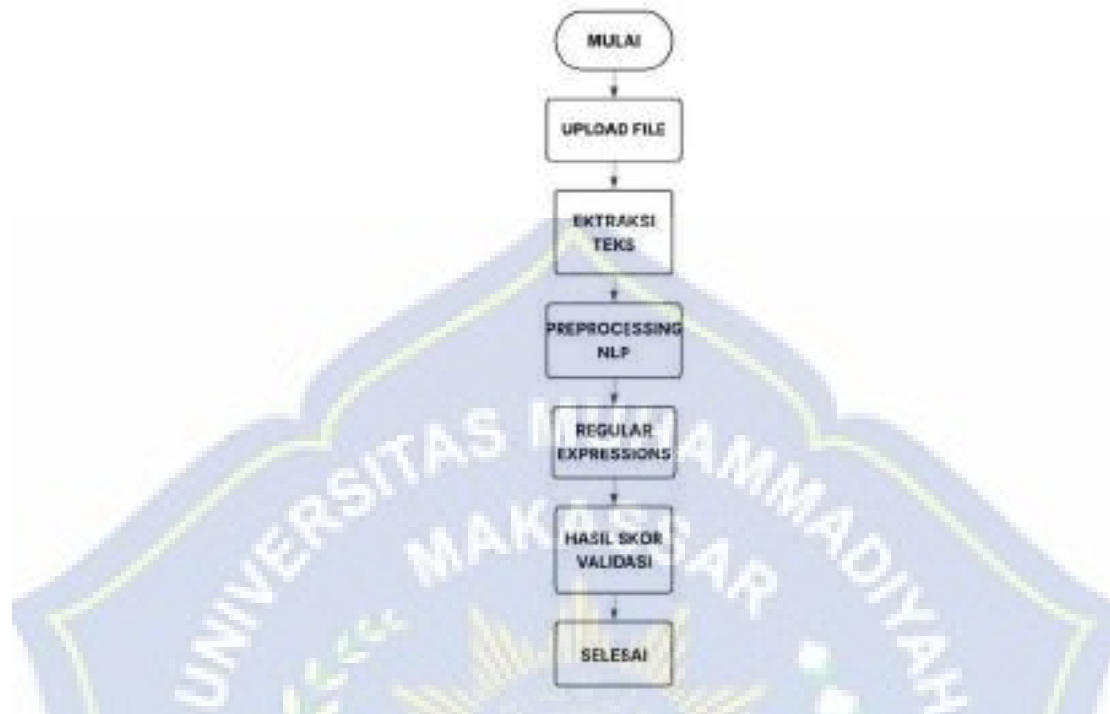
B. Alat Dan Bahan

Untuk melaksanakan penelitian diperlukan alat dan bahan yang akan digunakan pada penelitian ini yaitu :

1. Hardware (Perangkat Keras)
 - a) Laptop ASUS A455L
 - b) RAM 4,00 GB
2. Intel Core i5-5200U 2.7GHz
3. Software (Perangkat Lunak)
 - a) Visual Studio Code
 - b) Python
4. *AINET*: Jurnal Informatika

C. Perancangan Sistem

1. Perancangan sistem



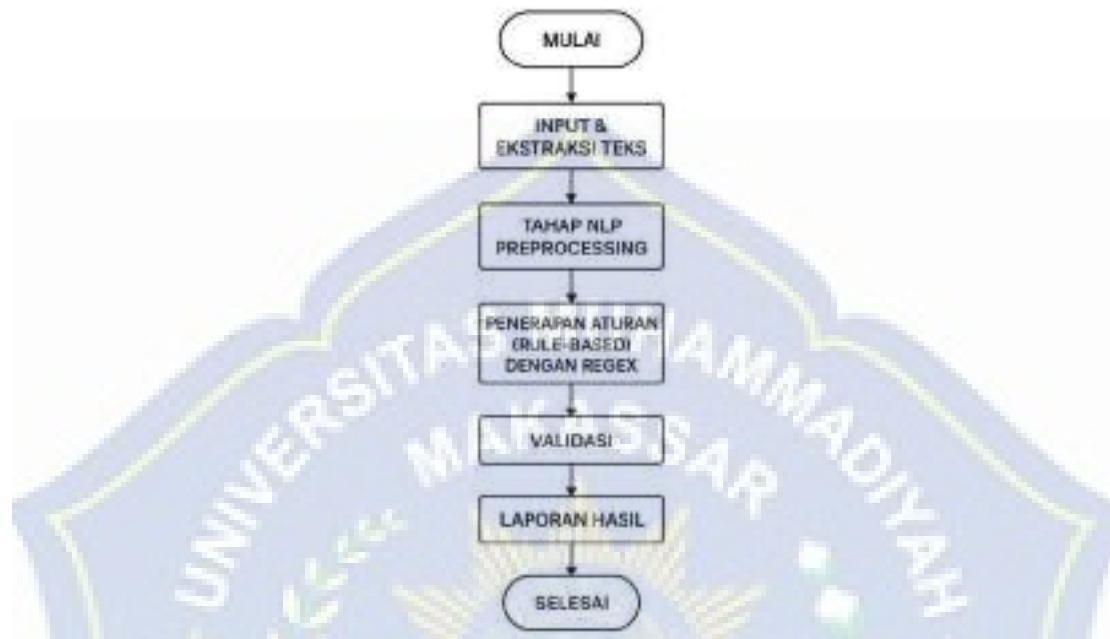
Gambar 5. Perancangan Sistem

Untuk memudahkan proses perancangan penelitian ini menggunakan flowchart perancangan sistem sebagai berikut :

Proses dimulai pada tahap *Mulai*, di mana sistem siap menerima input dokumen dari pengguna. Selanjutnya, pengguna mengunggah file jurnal yang akan divalidasi melalui tahap *Upload File*. Setelah file berhasil diunggah, sistem melakukan *Ekstraksi Teks* untuk mengambil seluruh isi teks dari dokumen, baik yang berformat .docx, .pdf, maupun format teks lainnya. Tahap berikutnya adalah *Preprocessing NLP*, yaitu proses pembersihan dan normalisasi teks, seperti menghapus karakter yang tidak relevan, mengubah huruf menjadi *lowercase*, melakukan tokenisasi, menghapus *stopwords*, serta melakukan stemming atau lemmatization agar teks siap dianalisis. Setelah itu, sistem memasuki tahap *Regular Expressions*, yaitu pencocokan pola teks berdasarkan aturan penulisan jurnal, misalnya format sitasi, daftar pustaka, penomoran tabel, atau penulisan judul. Hasil dari proses ini ditampilkan pada tahap *Hasil Skor Validasi*, berupa persentase kesesuaian dokumen terhadap standar penulisan, disertai informasi kesalahan yang ditemukan. Tahap

terakhir adalah *Selesai*, di mana sistem menyajikan laporan validasi sehingga pengguna dapat melakukan revisi dokumen sesuai rekomendasi yang diberikan.

2. Flowchart pada metode NLP Rule-Based



Gambar 6. flowchart nlp rule-based

Flowchart NLP Rule-Based menggambarkan alur kerja system seperti berikut :

Proses diawali pada tahap *mulai* ketika sistem siap menerima masukan. Pengguna mengunggah dokumen jurnal dalam format .docx atau .txt melalui tahap input dokumen. Setelah dokumen diterima, sistem melakukan *preprocessing* menggunakan teknik Natural Language Processing (NLP), meliputi perubahan semua huruf menjadi kecil (*lowercasing*), pemecahan teks menjadi kata-kata (*tokenisasi*), penghapusan kata-kata umum yang tidak relevan (*stopword removal*), serta pengubahan kata menjadi bentuk dasarnya (*stemming*).

Tahap berikutnya adalah segmentasi bagian jurnal, di mana sistem memisahkan struktur dokumen menjadi bagian-bagian seperti judul, abstrak (Bahasa Indonesia dan Bahasa Inggris), kata kunci, pendahuluan, metode, hasil dan pembahasan, kesimpulan, serta daftar pustaka. Setelah bagian-bagian ini teridentifikasi, sistem menerapkan aturan berbasis *rule-based*

yang telah ditentukan, misalnya pengecekan panjang kata pada abstrak, jumlah minimal kata kunci, kesesuaian format penulisan dengan pedoman *AINET: Jurnal Informatika*, dan tata cara penulisan daftar pustaka.

Selanjutnya, proses *validasi regex* dijalankan untuk memeriksa pola teks yang spesifik, seperti format huruf pada judul, pola tahun pada daftar pustaka, atau struktur penomoran bab. Hasil dari semua pemeriksaan ini disajikan pada tahap output validasi, di mana sistem menandai bagian yang sesuai aturan sebagai valid, dan bagian yang tidak sesuai aturan sebagai perlu revisi. Alur kemudian diakhiri pada tahap selesai, menghasilkan laporan yang dapat digunakan penulis untuk memperbaiki naskah jurnal sebelum pengiriman ke penerbit.

D. Teknik Pengujian Sistem

Pengujian sistem dalam penelitian ini dilakukan menggunakan metode *Black Box Testing*. Metode ini berfokus pada hubungan antara *input* dan *output* aplikasi tanpa perlu melihat kode sumber atau struktur internal program. Tujuan utama pengujian ini adalah untuk memastikan bahwa sistem memberikan hasil keluaran sesuai dengan yang diharapkan berdasarkan data masukan yang diberikan.

Pada prinsipnya, *Black Box Testing* hanya menilai kinerja sistem dari sisi pengguna. Apabila hasil keluaran sudah sesuai dengan spesifikasi, maka sistem dianggap berfungsi dengan benar. Sebaliknya, jika keluaran tidak sesuai, maka perlu dilakukan perbaikan pada mekanisme sistem.

Dalam penelitian ini, beberapa teknik pengujian black box yang digunakan antara lain:

1. *Equivalence Partitioning*

Equivalence Partitioning adalah teknik pengujian yang membagi input sistem ke dalam kelas data valid dan tidak valid. Setiap kelas cukup diuji dengan satu data perwakilan, karena data dalam kelas yang sama akan memberikan hasil yang sama.

2. *Boundary Value Analysis*

Boundary Value Analysis digunakan untuk menguji nilai batas bawah dan batas atas suatu input, karena pada titik inilah sistem sering kali mengalami kesalahan. Teknik ini relevan untuk aturan yang memiliki rentang nilai

tertentu.

3. *Decision Table Testing*

Decision Table Testing digunakan untuk menguji kombinasi kondisi yang mungkin terjadi pada sistem. Teknik ini membantu memastikan bahwa setiap kemungkinan kombinasi input menghasilkan output yang benar.

4. *State Transition Testing*

State Transition Testing berfokus pada perubahan status atau keadaan sistem akibat suatu input. Teknik ini digunakan ketika sistem memiliki alur proses tertentu yang dipengaruhi oleh tindakan pengguna.

5. *Cause-Effect Graphing*

Cause-Effect Graphing adalah teknik yang memetakan hubungan antara sebab (input atau kondisi) dan akibat (output sistem).

6. *Error Guessing*

Error Guessing merupakan teknik pengujian yang didasarkan pada pengalaman penguji dalam menebak kesalahan umum yang mungkin terjadi. Teknik ini sering digunakan untuk menemukan kesalahan yang tidak teridentifikasi oleh metode formal lainnya.

E. Teknik Analisis Data

Analisis data merupakan tahapan penting dalam penelitian yang bertujuan mengubah data mentah menjadi informasi yang bermakna. Proses ini mencakup pengumpulan, pembersihan, transformasi, hingga pengolahan data sehingga dapat digunakan untuk menemukan pola, tren, serta hubungan antarvariabel. Hasil analisis selanjutnya dijadikan dasar dalam pengambilan keputusan serta evaluasi efektivitas sistem yang dikembangkan.

Langkah-langkah analisis data dalam penelitian ini dilakukan melalui beberapa tahapan berikut:

1. Pengumpulan data

Data yang akan di analisis berasal dari teks jurnal mahasiswa prodi informatika informatika Data yang akan di analisis berasal dari teks jurnal mahasiswa prodi informatika dan berasal dari repositori skripsi

2. Preprocessing Data

Data skripsi melalui tahap *preprocessing* untuk mempersiapkan teks sebelum di

proses lebih lanjut. Langkah-langkah ini meliputi:

- a. Pembersihan teks: Menghapus karakter yang tidak relevan seperti tanda baca, angka, dan simbol.
- b. *Tokenisasi*: Memecah teks menjadi unit-unit kecil seperti kata atau kalimat.

3. Ekstraksi informasi

Setelah *preprocessing*, informasi penting dari teks skripsi diekstraksi menggunakan model NLP. Informasi yang diekstraksi meliputi bagian: pendahuluan, tinjauan pustaka, metodologi, hasil dan pembahasan, serta kesimpulan.

4. Evaluasi Model NLP

Evaluasi dilakukan untuk mengukur performa model NLP dalam mengekstraksi informasi dan memformat teks sesuai template jurnal. Metode evaluasi meliputi:

- a. *Akurasi*: Mengukur ketepatan informasi yang diekstraksi.
- b. *Recall dan Precision*: Menilai seberapa banyak informasi relevan yang berhasil diekstraksi (*recall*) dan seberapa banyak informasi yang diekstraksi benar-benar relevan (*precision*).

5. Validasi Hasil

Validasi dilakukan dengan membandingkan hasil validasi aturan penulisan jurnal yang dihasilkan oleh aplikasi dengan hasil validasi manual yang dilakukan oleh mahasiswa.

- a. Perbandingan Kualitas Teks: Melihat apakah teks yang dihasilkan aplikasi memenuhi standar jurnal ilmiah.
- b. Penilaian Subjektif: Meminta penilaian dari dosen atau ahli terkait untuk menilai kualitas teks yang dihasilkan.

6. Visualisasi Data

Visualisasi digunakan untuk menyajikan hasil analisis secara grafis agar lebih mudah dipahami dan diinterpretasikan.

BAB IV

HASIL DAN PEMBAHASAN

A. Deskripsi Umum Implementasi Sistem

Implementasi sistem pada penelitian ini dilakukan dengan membangun sebuah aplikasi *web* yang diberi nama *Interactive AINET Journal Format Validator*. Sistem ini dirancang untuk membantu pengguna, khususnya mahasiswa dan dosen, dalam memvalidasi artikel ilmiah agar sesuai dengan standar penulisan *AINET: Jurnal Informatika*. Aplikasi dibangun menggunakan *Flask (Python)* sebagai *backend*, sedangkan tampilan antarmuka memanfaatkan *HTML*, *CSS (TailwindCSS)*, dan *JavaScript* sehingga mudah diakses melalui *browser* tanpa perlu instalasi tambahan.

Secara umum, implementasi dilakukan melalui beberapa tahapan. Pertama, pengguna dapat mengunggah dua dokumen yaitu template *AINET: Jurnal Informatika* dan artikel *submission* yang akan divalidasi. Fitur unggah file dilengkapi dengan *drag & drop* serta validasi *client-side* untuk memastikan ukuran file tidak melebihi 16 MB dan hanya menerima format *.docx* atau *.txt*. File yang berhasil diunggah dikirim ke server melalui *endpoint/upload_files*, kemudian disimpan sementara menggunakan *session ID* unik agar tidak terjadi bentrok antar pengguna. Implementasi ini membuat sistem lebih praktis karena pengguna tidak perlu melakukan pengaturan manual maupun mengedit kode program.

Setelah file diterima, sistem menjalankan tahap ekstraksi teks. Untuk dokumen berformat *.docx*, teks diekstraksi menggunakan pustaka *python-docx*, sedangkan file *.txt* diproses langsung melalui pembacaan *string*. Hasil ekstraksi ini kemudian masuk ke tahap *Natural Language Processing (NLP) preprocessing pipeline*, yang meliputi *lowercasing*, *segmentasi* kalimat, tokenisasi kata, penghapusan tanda baca, *stopword removal* (bahasa Inggris dan Indonesia), serta *stemming*. Tahap ini memastikan teks dalam kondisi bersih, seragam, dan siap dianalisis lebih lanjut.

Tahap berikutnya adalah validasi berbasis aturan (*rule-based*) yang dipadukan dengan *Regular Expressions (Regex)*. Validasi meliputi:

- 1) pengecekan *header* ISSN (e-ISSN dan p-ISSN) sesuai standar AINET,
- 2) kesesuaian format judul artikel
- 3) struktur penulis, email, dan afiliasi

- 4) keberadaan abstrak serta kata kunci dalam dua Bahasa
- 5) kelengkapan bagian wajib seperti Pendahuluan, Metode, Hasil dan Pembahasan, Kesimpulan, dan Referensi, serta
- 6) format daftar pustaka apakah menggunakan gaya IEEE atau APA. Hasil validasi ditampilkan secara detail, termasuk bagian yang benar, kesalahan yang ditemukan, serta posisi teks yang tidak sesuai.
- 7) Terakhir, sistem menyajikan hasil berupa skor kepatuhan yang dihitung berdasarkan bobot tiap aspek (ISSN 10%, Judul 10%, Penulis 15%, Abstrak & Keywords 25%, Struktur Bagian Wajib 30%, dan Referensi 10%). Skor ini dilengkapi dengan laporan validasi interaktif sehingga pengguna dapat langsung mengetahui kekurangan naskah dan melakukan revisi sebelum proses publikasi.

Dengan implementasi ini, sistem berhasil memadukan teknologi Flask, NLP, dan Regex menjadi sebuah aplikasi validasi otomatis yang bersifat interaktif, praktis, dan transparan. Hal ini diharapkan dapat meningkatkan efisiensi penulisan artikel, menjaga konsistensi format, serta mendukung peningkatan kualitas publikasi ilmiah di lingkungan akademik.

B. Hasil Implementasi Sistem

Hasil implementasi sistem menunjukkan bagaimana aplikasi Interactive AINET Journal Format Validator bekerja dalam membantu pengguna memvalidasi penulisan artikel sesuai standar *AINET: Jurnal Informatika*. Pada tampilan awal, sistem menampilkan halaman utama dengan antarmuka sederhana yang dilengkapi menu untuk mengunggah file template dan artikel submission. Antarmuka ini dirancang ramah pengguna dengan adanya fitur *drag and drop* serta tombol *upload file* sehingga memudahkan mahasiswa maupun dosen untuk memulai proses validasi.

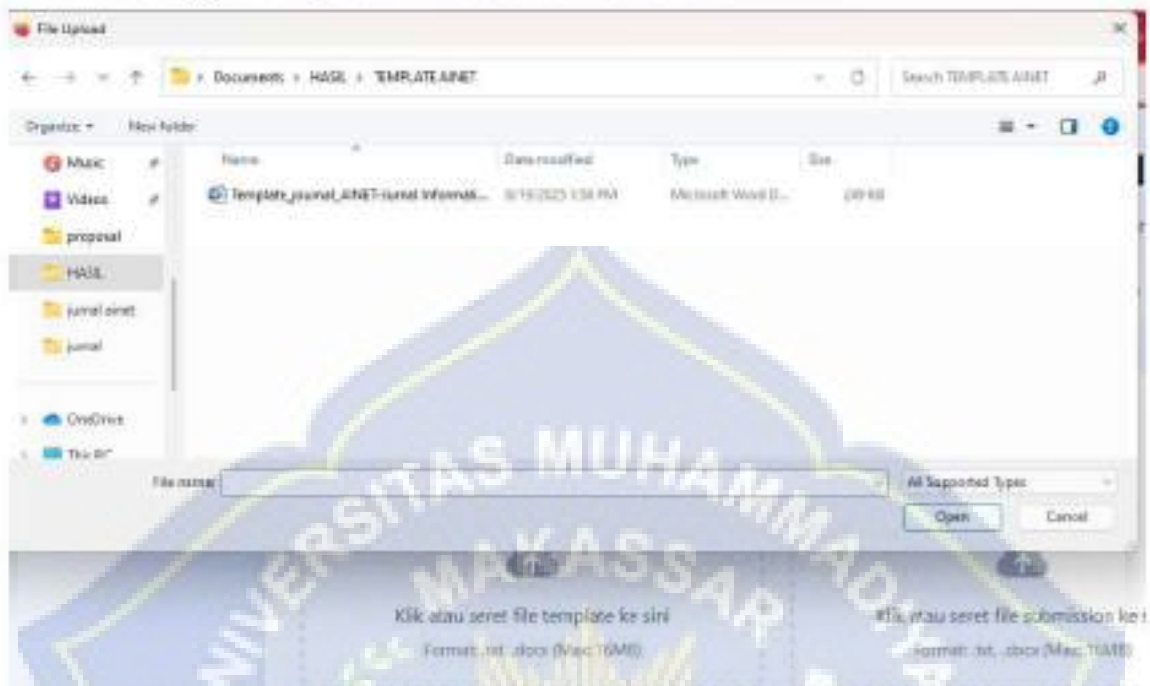
I. Tampilan Awal Aplikasi



Gambar 7. desain antar muka sistem

Langkah pertama yang dilakukan pengguna adalah mengunggah file template *AINET: Jurnal Informatika* dalam format .docx. File ini berfungsi sebagai acuan pembandingan agar sistem dapat menilai kesesuaian format artikel. Setelah itu, pengguna diminta mengunggah dokumen submission atau artikel yang akan divalidasi. Sistem hanya menerima dokumen berformat .docx atau .txt dengan ukuran maksimal 16 MB, sehingga jika pengguna mengunggah file dengan format tidak sesuai seperti .pdf atau .jpg, sistem akan menolaknya dan menampilkan pesan kesalahan. Hal ini memastikan bahwa input yang diproses benar-benar sesuai dengan standar.

2. Unggah Template *AINET: Jurnal Informatika*

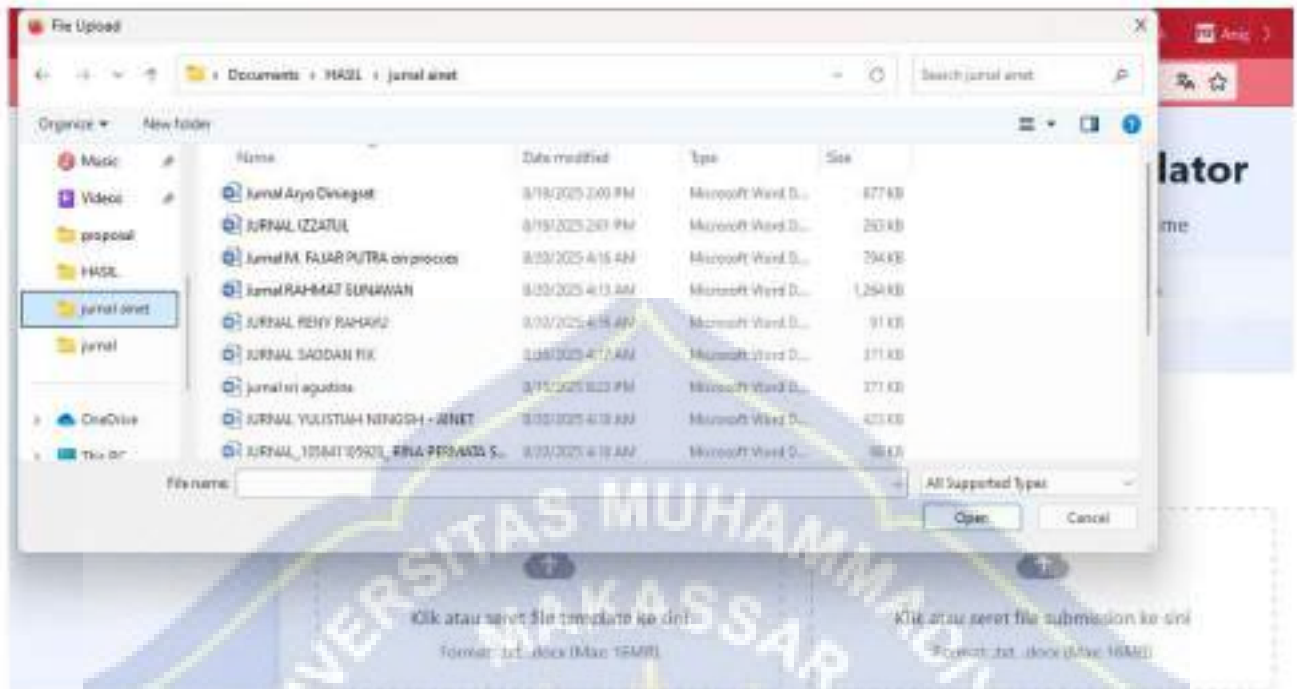


Gambar 8. unggah template *AINET: Jurnal Informatika*

Pada tahap ini, pengguna diminta untuk mengunggah file template *AINET: Jurnal Informatika* dalam format .docx. File template berfungsi sebagai acuan pembandingan agar sistem dapat menilai kesesuaian format artikel. Setelah berhasil diunggah, sistem akan menampilkan status file template yang diterima. Gambar tersebut merupakan tampilan awal aplikasi "validasi penulisan artikel *AINET: Jurnal Informatika*". Pengguna diminta memilih format template jurnal dan artikel submissions berdasarkan kebutuhannya. Pada halaman utama juga terdapat penjelasan tentang bagaimana cara menggunakan aplikasi validasi penulisan artikel *AINET: Jurnal Informatika*.

Pada tampilan template ainet pengguna dapat mengklik dan mengunggah file tersebut

1. Unggah Dokumen *Submission*



Gambar 9.. jurnal submission

Selanjutnya, pengguna mengunggah artikel/jurnal submission yang akan divalidasi. Dokumen harus berformat .docx atau .txt dengan ukuran maksimal 16 MB. Sistem otomatis menolak file dalam format lain seperti .pdf atau .jpg dengan menampilkan pesan error. Proses ini menjamin bahwa hanya dokumen yang sesuai ketentuan yang bisa diproses.

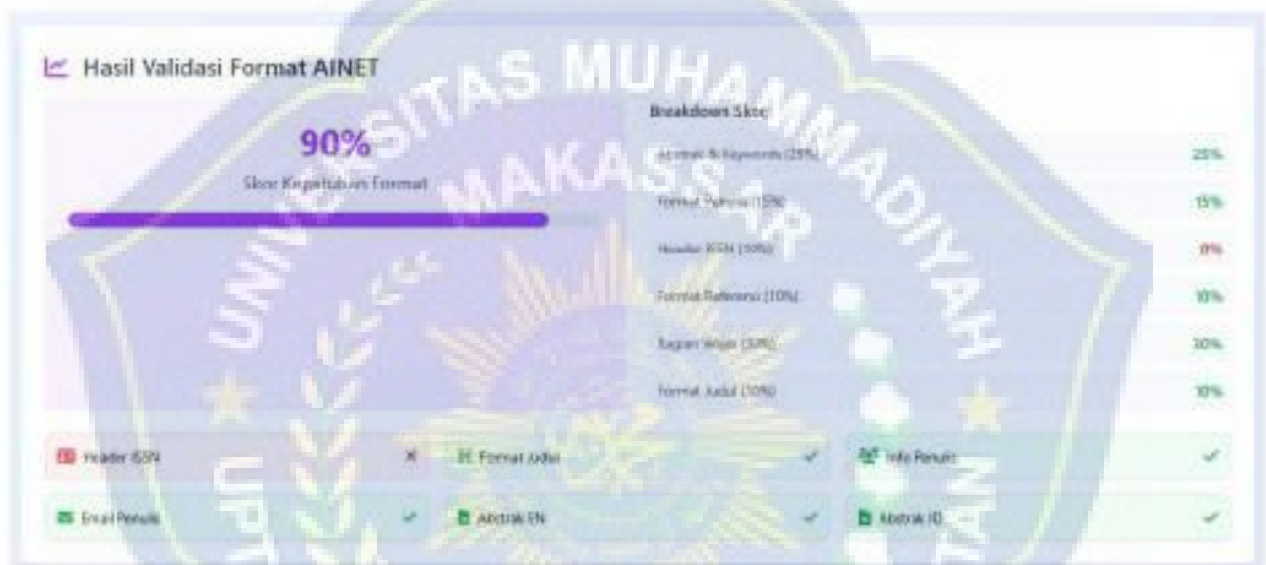
2. Proses Analisis & Validasi



Gambar 10. unggah dokumen

Setelah kedua file berhasil diunggah, pengguna menekan tombol “Analisis & Validasi Format AINET”. Pada tahap ini, sistem melakukan serangkaian proses otomatis:

- Ekstraksi teks dari dokumen submission.
- Preprocessing NLP, meliputi normalisasi teks, tokenisasi, penghapusan stopwords, dan stemming.
- Validasi Regex, untuk memeriksa pola penulisan seperti ISSN, judul, format penulis, abstrak, keywords, dan daftar pustaka.



Gambar 11. hasil validasi.

Hasil validasi seperti pada gambar tersebut di atas menjelaskan bahwa skor kepatuhan format mencapai 90% berikut penjelasannya :

- Validasi Header ISSN*

Bertujuan untuk memastikan bahwa artikel memiliki identitas jurnal yang benar, yaitu mencantumkan e-ISSN dan p-ISSN sesuai dengan nomor resmi *AINET: Jurnal Informatika*. Hal ini penting karena keberadaan ISSN menjadi tanda pengenal resmi jurnal ilmiah, sehingga apabila tidak sesuai, dokumen dianggap tidak valid.

- Validasi Format Judul*

Dilakukan untuk mengecek apakah judul artikel ditulis dengan benar, misalnya menggunakan huruf kapital pada setiap kata penting (Title Case), memiliki panjang yang ideal (tidak terlalu pendek atau panjang).

serta posisinya benar-benar berada di bagian atas naskah. Judul yang sesuai aturan akan memudahkan pembaca dan menambah kesan profesional.

c) Validasi Format Penulis

Digunakan untuk memastikan penulisan nama penulis, alamat email, dan afiliasi sesuai dengan pedoman penulisan. Biasanya nama penulis harus lengkap, afiliasi ditulis jelas, serta email korespondensi ditandai dengan benar. Selain itu, format penomoran afiliasi seperti penggunaan superscript juga dicek agar konsisten.

d) Validasi Abstrak dan *Keywords*

Berfungsi untuk memastikan bahwa artikel memiliki abstrak dalam bahasa Indonesia dan bahasa Inggris, sesuai standar jurnal internasional. Abstrak harus ditulis dengan panjang tertentu (umumnya 150–250 kata), sedangkan *keywords* atau kata kunci harus tersedia dalam dua bahasa dan berjumlah 3–5 kata atau frasa agar memudahkan indeksasi.

e) Validasi Bagian Wajib

Dilakukan untuk memastikan naskah memiliki struktur utama yang lengkap, yaitu bagian Pendahuluan, Metode, Hasil dan Pembahasan, Kesimpulan, serta Referensi. Jika salah satu bagian ini tidak ditemukan, maka artikel tidak sesuai dengan format *AINET: Jurnal Informatika*.

f) Validasi Format Referensi

Bertujuan untuk mengecek apakah daftar pustaka ditulis dengan gaya sitasi yang benar. *AINET: Jurnal Informatika* umumnya menggunakan format IEEE, yaitu dengan penomoran [1], [2], dan seterusnya, bukan format lain seperti APA. Dengan validasi ini, kualitas penulisan daftar pustaka lebih konsisten dan sesuai standar internasional.

Tabel 2. Template Ainet

Tahap Proses		Jumlah	Contoh (Sample)
Teks Asli		-	<i>Judul Naskah Publikasi Maksimum 15 Kata dalam Bahasa Indonesia, Huruf Kapital di Awal Kata (Center, Arial 14 pt, Bold) Penulis Pertama1, Penulis kedua2, Penulis ketiga3...*</i>
Segmentasi	Kalimat	140	<i>Judul Naskah Publikasi Maksimum 15 Kata dalam Bahasa Indonesia, Huruf Kapital di Awal Kata (Center, Arial 14 pt, Bold) Penulis Pertama1, Penulis kedua2, Penulis ketiga3...*</i>
Tokenisasi		2.048	judul, naskah, publikasi, maksimum, 15, kata, dalam, bahasa, indonesia, ...
Pembersihan Token		1.544	judul, naskah, publikasi, maksimum, 15, kata, dalam, bahasa, indonesia, huruf...
Penghapusan Stopwords		1.119	judul, naskah, publikasi, maksimum, kata, bahasa, indonesia, huruf, kapital, awal...
Stemming		1.119	judul, naskah, publikasi, maksimum, kata, bahasa, indonesia, huruf, kapit, awal...

Tabel 3. Validasi Ainet

Tahap proses	Jumlah	Sample
Teks asli	-	<i>PERANCANGAN DAN IMPLEMENTASI KONSULTASI AKADEMIK BERBASIS WEBSITE MAHASISWA PRODI INFORMATIKA UNISMUH MAKASSAR Rahmat Gunawan1, Muhyiddin AM Hayat2, Titin Wahyuni3...</i>

Tahap proses	Jumlah	Sample
Tokenisasi	2.213	perancangan, dan, implementasi, konsultasi, akademik, berbasis, website, mahasiswa, prodi, informatika...
Pembersihan Token	1.774	perancangan, dan, implementasi, konsultasi, akademik, berbasis, website, mahasiswa, prodi, informatika...
Penghapusan Stopwords	1.349	perancangan, implementasi, konsultasi, akademik, berbasis, website, mahasiswa, prodi, informatika, unismuh...

C. Hasil Validasi dengan Metode NLP dan REGEX

1. Proses Validasi *NLP*

Selain menggunakan Regex untuk memeriksa kesesuaian pola heading, sistem ini juga melakukan validasi berbasis Natural Language Processing (NLP). Tujuan dari pengujian NLP adalah untuk memastikan bahwa isi artikel tidak hanya sesuai dengan format, tetapi juga memenuhi standar kualitas penulisan, seperti panjang teks, kesesuaian kata kunci, serta konsistensi penggunaan bahasa.

Tahap awal validasi NLP dilakukan melalui tokenisasi, yaitu proses pemisahan artikel menjadi unit kata dan kalimat. Tokenisasi memungkinkan sistem untuk menghitung jumlah kata dalam judul, abstrak, maupun bagian lain yang memiliki batasan panjang tertentu. Selanjutnya dilakukan stopword removal, yaitu penghapusan kata-kata umum yang tidak bermakna penting (misalnya *yang*, *dan*, *di*, *dari*). Proses ini bertujuan untuk menekankan kata kunci utama yang relevan dengan isi artikel.

Hasil tokenisasi kemudian diproses dengan stemming, di mana kata-kata dikembalikan ke bentuk dasarnya. Contohnya, kata *menganalisis*, *dianalisis*, *analisis* semuanya direduksi menjadi kata dasar *analisis*. Hal ini membuat sistem lebih efektif dalam menghitung frekuensi kata kunci. Untuk menilai tingkat kepentingan kata tertentu dalam dokumen, digunakan metode TF-IDF (Term Frequency – Inverse Document Frequency). Dengan TF-IDF, sistem dapat mendeteksi apakah kata kunci yang ditentukan penulis memang muncul secara dominan di bagian abstrak atau isi artikel.

Sebagai ilustrasi hasil pengujian, judul artikel yang diuji terdiri atas 12 kata, sehingga berada dalam rentang ketentuan 10–15 kata. Abstrak berjumlah 132 kata, yang berarti lebih pendek dari batas minimum 150 kata sehingga dinyatakan tidak valid. Pada bagian kata kunci, artikel mencantumkan 4 kata kunci, yang sudah sesuai dengan ketentuan minimal tiga kata. Hasil perhitungan TF-IDF menunjukkan bahwa kata kunci utama seperti *NLP*, *Regex*, dan *Validasi* muncul lebih dari tiga kali di bagian abstrak, sehingga dapat dinyatakan konsisten. Dari sisi kualitas bahasa, persentase stopwords berada pada angka 25% dari total kata, yang masih dalam batas wajar (<40%) sehingga dianggap valid.

Secara umum, hasil validasi NLP menunjukkan bahwa artikel telah memenuhi sebagian besar aturan kualitas penulisan, khususnya dari aspek jumlah kata kunci, relevansi isi dengan kata kunci, serta tingkat penggunaan stopwords. Namun, terdapat kelemahan pada panjang abstrak yang masih di bawah batas minimal, sehingga artikel perlu direvisi untuk menyesuaikan dengan standar penulisan. Hal ini membuktikan bahwa validasi berbasis NLP dapat memberikan evaluasi yang lebih menyeluruh terhadap kualitas isi artikel, sekaligus melengkapi hasil validasi berbasis Regex yang berfokus pada struktur dan format.

Tabel 4 Hasil Validasi NLP Artikel

Komponen	Aturan	Hasil validasi	Status
	AINET: Jurnal Informatika		
Judul	10-15 kata	12 kata	✓ Valid
Abstrak	150-250 kata	132 kata	✗ Tidak Valid (kurang panjang)
Kata kunci	Minimal 3 kata, maksimal 5 kata	4 kata kunci	✓ Valid
Relevansi kata kunci	Kata kunci harus muncul ≥ 3 kali di abstrak	"NLP" (5x), "Regex" (3x), "Validasi" (6x)	✓ Valid
Stopwords	Tidak lebih dari 40% total kata	25% dari total kata	✓ Valid

Submission NLP Processing

Teks Asli

ISTEM BAKAR DALAM MENDIAGNOSIS PENYAKIT MATA DENGAN MENGGUNAKAN METODE FUZZY CHAINING (S. Agrestal, Trian Wal-rail, Rida Yuliana Rahel)
IT Informatika, Universitas Muhammadiyah Makassar, Makassar

Segara Wana Kalimat

Frequency: 0.4%

Sampel: 50000 PARAGRAF DALAM BENTUKAN DIRAS. PERHATIKAN AGAK BANYAK BENTUKAN PARAGRAF YANG TERDAPAT DALAM 50000 PARAGRAF. Untuk lebih jelasnya perhatikan Gambar 1.2.3 dan Gambar 1.2.4.

Identities

January 1997

Small note: this table contains text that has been converted from a PDF.

Gambar 12 .proses validasi nlp

Gambar tersebut di atas menjelaskan tentang Validasi Metode algoritma NLP dirancang untuk melakukan serangkaian tahapan *preprocessing* teks agar data mentah menjadi lebih terstruktur sebelum dianalisis lebih lanjut.

2. Proses Validasi Regex

Pengujian dengan menggunakan Regular Expressions (Regex) dilakukan untuk memvalidasi struktur artikel berdasarkan kesesuaian heading utama. Regex dipilih karena mampu mendeteksi pola teks tertentu, menghitung jumlah kemunculan, dan menentukan posisi keberadaan pola di dalam dokumen. Hasil pengujian ditampilkan dalam bentuk log dan dirangkum ke dalam tabel validasi.

Berdasarkan tabel hasil validasi, pada bagian Pendahuluan sistem berhasil mendeteksi pola *Pendahuluan* sebanyak satu kali pada posisi [3616]. Pola alternatif berupa *Introduction* tidak ditemukan, sehingga dapat disimpulkan bahwa artikel ini konsisten menggunakan heading dalam bahasa Indonesia. Hal ini sesuai dengan format penulisan artikel pada *AINET: Jurnal Informatika* yang menetapkan struktur artikel dalam bahasa Indonesia.

Pada bagian Metode, sistem menemukan adanya dua pola yang berbeda. Heading *Metode Penelitian* terdeteksi sebanyak satu kali pada posisi [5506], yang menunjukkan kesesuaian dengan format. Namun demikian, ditemukan

pula istilah *Methods* pada posisi [2666]. Pola *Methodology* tidak terdeteksi sama sekali. Temuan ini mengindikasikan adanya inkonsistensi penulisan, karena penulis menggunakan campuran bahasa Indonesia dan Inggris dalam penamaan bagian metode.

Hasil pengujian pada bagian Hasil dan Pembahasan menunjukkan bahwa heading utama *Hasil dan Pembahasan* berhasil dideteksi sebanyak satu kali pada posisi [10966]. Pola *Results and Discussion* tidak ditemukan, tetapi istilah *Results* muncul sebanyak tiga kali pada posisi [3017, 3133, 3231]. Hal ini kembali memperlihatkan adanya campuran bahasa antara Indonesia dan Inggris. Meskipun heading utama sudah sesuai dengan format AINET, keberadaan istilah *Results* sebaiknya diperbaiki agar konsisten.

Sementara itu, pada bagian Kesimpulan, sistem mendeteksi heading *Kesimpulan* sebanyak delapan kali dengan sebaran posisi [8425, 8493, 8554, 8667, 17390, 17401, 17472, 17901]. Hal ini menandakan konsistensi penggunaan bahasa Indonesia untuk heading kesimpulan. Pola alternatif *Conclusion* tidak ditemukan dalam dokumen. Walaupun kemunculannya cukup banyak, hal ini dapat disebabkan oleh pengulangan kata *Kesimpulan* di dalam paragraf penutup.

Secara umum, hasil pengujian menunjukkan bahwa artikel telah memenuhi sebagian besar aturan format yang ditentukan, terutama dalam penggunaan heading bahasa Indonesia. Namun, terdapat beberapa inkonsistensi berupa penggunaan istilah berbahasa Inggris seperti *Methods* dan *Results*. Dengan demikian, sistem berbasis regex terbukti mampu mengidentifikasi bagian yang valid maupun tidak valid, serta memberikan umpan balik yang jelas mengenai posisi dan jumlah kesalahan yang perlu diperbaiki.

Tabel 5 Hasil Validasi Regex Heading Artikel

	Pola regex	Jumlah match	Posisi	Status
Pendahuluan	\bPendahuluan\b	1	[3616]	✓ Valid
	\bIntroduction\b	0	-	✗ Tidak ditemukan
Metode	\bMetode	1	[5506]	✓ Valid

	Pola regex	Jumlah match	Posisi	Status
	Penelitian\b			✗ Tidak
	\bMethodology\b			ditemukan
	\bMethods\b	0	-	△ Campuran
		1	[2666]	Bahasa
Hasil	\bHasil dan	1	[10966]	✓ Valid
	Pembahasan\b			
	\bResults and	0	-	
	Discussion\b			
	\bResults\b			
		3	[3017, 3133, 3231]	
Kesimpulan	\bKesimpulan\b	8	[8425, 8493, 8554, 8667, 17390, 17401, 17472, 17901]	✓ Valid

Regex Pattern Matching Logs			
PENDAHULUAN			✓
Pattern: \bIntroduction\b			
Matches: 1 Positions: [3018]			
PENDAHULUAN			✗
Pattern: \bIntroduction\b			
Matches: 0 Positions: []			
METODE			✓
Pattern: \bMetode Penelitian\b			
Matches: 1 Positions: [1104]			
METODE			✗
Pattern: \bMethodology\b			

Gambar 13. validasi regex

Gambar ini menunjukkan hasil pencocokan pola (pattern) dengan teks naskah menggunakan metode Regular Expressions (Regex)

3. Analisis Perbandingan Hasil Validasi Regex dan NLP

Hasil validasi dengan menggunakan Regex menunjukkan bahwa sistem mampu memeriksa struktur dan format artikel secara tepat. Regex berhasil mendeteksi keberadaan heading utama, seperti *Pendahuluan*, *Metode Penelitian*, *Hasil dan Pembahasan*, serta *Kesimpulan*. Pada pengujian juga ditemukan adanya inkonsistensi dalam penggunaan bahasa, khususnya pada bagian metode yang selain menggunakan istilah *Metode Penelitian* juga memuat istilah *Methods*, serta pada bagian hasil yang selain menggunakan heading *Hasil dan Pembahasan* juga terdapat istilah *Results*. Temuan ini membuktikan bahwa Regex efektif untuk mengidentifikasi kesalahan struktural berupa campuran bahasa maupun ketidaksesuaian heading dengan format *AINET: Jurnal Informatika*.

Sementara itu, hasil validasi dengan NLP lebih berfokus pada isi artikel. NLP memproses teks dengan tahapan tokenisasi, stopword removal, stemming, hingga perhitungan bobot kata menggunakan TF-IDF. Berdasarkan hasil validasi, judul artikel dengan jumlah 12 kata dinyatakan valid karena sesuai aturan 10–15 kata. Kata kunci yang digunakan berjumlah 4, sehingga juga sesuai ketentuan minimal 3 kata dan maksimal 5 kata. Relevansi kata kunci dengan abstrak juga valid, karena kata kunci utama muncul lebih dari tiga kali di dalam abstrak. Persentase stopwords sebesar 25% masih berada dalam batas toleransi <40%, sehingga dinyatakan valid. Namun, pada bagian abstrak ditemukan jumlah kata hanya 132, sehingga tidak memenuhi aturan panjang minimal 150 kata. Hal ini menyebabkan abstrak dinyatakan tidak valid dan perlu dilakukan revisi.

Perbedaan utama dari kedua metode validasi ini terletak pada fokus pemeriksaan. Regex bersifat deterministik, yaitu hanya memeriksa keberadaan pola tertentu dan memberikan hasil valid atau tidak valid berdasarkan kecocokan pola. Dengan demikian, Regex sangat efektif untuk mengevaluasi aspek struktur penulisan artikel. Sebaliknya, NLP bersifat analitis dan evaluatif karena mampu menilai kualitas isi artikel secara lebih mendalam, misalnya panjang teks, konsistensi kata kunci, hingga proporsi penggunaan stopwords.

Dengan demikian, dapat disimpulkan bahwa validasi menggunakan Regex dan NLP memiliki fungsi yang saling melengkapi. Regex berperan dalam menjaga kepatuhan artikel terhadap aturan format penulisan, sedangkan

NLP berperan dalam memastikan kualitas isi artikel sesuai dengan standar yang berlaku. Kombinasi keduanya menghasilkan sistem validasi yang lebih komprehensif, sehingga dapat membantu penulis artikel meminimalisir kesalahan baik dari sisi struktur maupun isi.

Tabel 6. ringkasan perbedaan

Aspek	Regex	Nlp
Focus	Struktur & format heading	Isi & kualitas teks
Ouput	Match, jumlah kemunculan, posisi	Perhitungan kata, frekuensi, persentase
Sifat	Ya/tidak(deterministik)	Analisis, evaluatif
Temutama utama	Healing valid tapi da campuran bahasa	Abstrak terlalu pendek, isi lain validasi

D. Pengujian Sistem

Pengujian sistem dilakukan dengan metode **black-box testing** menggunakan 20 sampel artikel AINET: jurnal informatika. Tujuan pengujian ini adalah untuk mengevaluasi kinerja sistem dalam mendeteksi kepatuhan format artikel berdasarkan aturan yang telah ditentukan.

1. Pengujian *black box* dengan teknik Tabel Keputusan

Teknik ini dilakukan dengan pendekatan sistematis, di mana kombinasi input dirangkum dalam sebuah tabel. Teknik tabel keputusan cocok untuk menguji fungsi yang memiliki hubungan logistik antara dua atau lebih input. Berikut adalah hasil pengujian *black box* dengan teknik tabel keputusan

Tabel 7. Black Box Testing

NO	ARTIKEL	KONDISI AKTUAL	SKENARIO UJI	OUTPUT YANG DI HARAPKAN	OUTPUT SISTEM	STATUS
1	A1	Valid	Artikel sesuai template penulisan judul	Valid	Valid	✓
2	A2	Valid	Artikel sesuai template tapi ada bagian yang hilang (referensinya)	Valid	Tidak valid	✗
3	A3	Valid	Artikel lengkap absrak ED & ID	Valid	Valid	✓
4	A4	Valid	Artikel sesuai aturan format judul	Valid	Valid	✓
5	A5	Valid	Artikel format benar (judu; kapital,abstrak sesuai)	Valid	Valid	✓
NO	ARTIKEL	KONDISI AKTUAL	SKENARIO UJI	OUTPUT YANG DI HARAPKAN	OUTPUT SISTEM	STATUS
6	A6	Valid	Artikel sesuai template bahasa	Valid	Valid	✓
7	A7	Valid	Artikel sesuai format jurul dan abstrak	Valid	Valid	✓
8	A8	Valid	Artikel format judul benar dan sesuai template	Valid	Valid	✓
9	A9	Valid	Artikel sesuai template judul & abstrak	Valid	Valid	✓

10	A10	Valid	Artikel format benar	Valid	Valid	✓
11	B1	Tidak valid	Artikel tidak memiliki abstrak EN & ID	Tidak Valid	Tidak valid	✓
12	B2	Tidak valid	Artikel tidak lengkap bagian abstrak EN & ID dan ada bagian yang hilang (metodenya)	Tidak Valid	Tidak valid	✓
13	B3	Tidak valid	Artikel lengkap format penulisan tapi bagian abstrak EN & ID tidak di temukan	Tidak Valid	Tidak valid	✓
14	B4	Tidak valid	Artikel lengkap dengan format judul dan abstrak tapi ada bagian yang hilang (metode)	Tidak valid	Tidak valid	✓
NO	ARTIKEL	KONDISI AKTUAL	SKENARIO UJI	OUTPUT YANG HARAPKAN	OUTPUT DI SISTEM	STATUS
15	B5	Tidak valid	Artikel tidak sesuai template abstrak dan bagian hilang (pendahuluan, metode, hasil, kesimpulan, referensi)	Valid	Tidak valid	✗
16	B6	Tidak valid	Artikel sesuai template format judul	Tidak valid	Tidak valid	✓
17	B7	Tidak valid	Artikel sesuai	Tidak Valid	Tidak valid	✓

			template format judul tapi abstrak EN & ID tidak di temukan			
18	B8 meli	Tidak valid	Artikel sesuai format judul tetapi ada bagian hilang (metode)	Tidak Valid	Tidak valid	✓
19	B9 akram	Tidak valid	Artikel sesuai format judul tapi tidak di temukan bagian abstrak EN & ID tidak di temukan	Tidak valid	Tidak valid	✓
20	B10 yass	Tidak valid	Artikel sesuai template format judul dan format penulis tapi abstrak tidak di temukan	Tidak valid	Tidak valid	✓

2. Pengujian Akurasi (Accuracy)

Mengukur persentase prediksi yang benar dibandingkan dengan total data uji.

$$Accuracy = \frac{Jumlah\ Prediksi\ Benar}{Total\ Seluruh\ Pengujian} \times 100\%$$

Total artikel uji = 20 (10 sebenarnya valid, 10 sebenarnya tidak valid)

1. Artikel Valid & dikenali Valid (True Positive, TP) = 9
2. Artikel Tidak Valid & dikenali Tidak Valid (True Negative, TN) = 9
3. Artikel Valid tapi salah dikenali Tidak Valid (False Negative, FN)=1
4. Artikel Tidak Valid tapi salah dikenali Valid (False Positive, FP) = 1

Akurasi

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \%$$

$$Accuracy = \frac{9 + 9}{9 + 9 + 1 + 1} \times 100\%$$

$$Accuracy = \frac{18}{20} \times 100\% = 90\%$$

Dengan demikian, sistem mencapai akurasi sebesar 90%, yang menunjukkan bahwa dari 20 artikel yang diuji, sebanyak 18 artikel berhasil diklasifikasikan dengan benar sesuai kondisi aktualnya. Tingkat kesalahan (error rate) sebesar 10% disebabkan oleh adanya 1 kasus false positive dan 1 kasus false negative.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

1. Implementasi Sistem

Sistem validasi penulisan artikel ilmiah berhasil diimplementasikan dengan menggabungkan metode Natural Language Processing (NLP) berbasis *rule-based* untuk preprocessing teks, segmentasi dokumen, serta ekstraksi bagian penting, dan Regular Expressions (Regex) untuk memvalidasi pola penulisan sesuai aturan AINET: Jurnal Informatika. Aplikasi yang dibangun mampu menampilkan skor kepatuhan format, log kesalahan, serta rekomendasi perbaikan secara interaktif, sehingga memudahkan pengguna dalam melakukan revisi artikel.

2. Hasil Pengujian Sistem

Pengujian sistem menggunakan metode Black Box Testing terhadap 20 artikel menunjukkan tingkat akurasi 90%, dengan 18 artikel terklasifikasi sesuai kondisi aktual. Hasil ini membuktikan bahwa kombinasi metode NLP dan Regex mampu mendeteksi kesalahan baik pada struktur maupun isi artikel secara cukup andal, meskipun masih terdapat keterbatasan pada deteksi abstrak bilingual, konsistensi heading, dan format referensi.

B. Saran

1. Pengembangan Sistem

Untuk meningkatkan kualitas implementasi, sistem dapat dikembangkan lebih lanjut dengan menambahkan dukungan format dokumen lain (PDF, LaTeX, OCR), serta memperluas dataset uji pdari berbagai jurnal selain AINET. Selain itu, metode NLP dapat diperkuat dengan algoritma berbasis machine learning atau deep learning agar mampu menangani variasi bahasa yang lebih kompleks.

2. Peningkatan Validasi

Disarankan agar validasi tidak hanya terbatas pada teks, tetapi juga mencakup elemen non-teks seperti tabel, gambar, dan sitasi otomatis. Integrasi sistem dengan Open Journal System (OJS) juga penting agar proses validasi dapat dilakukan langsung sebelum artikel dikirim ke penerbit, sehingga mendukung peningkatan kualitas publikasi ilmiah secara lebih luas.

DAFTAR PUSTAKA

- Riana, R., Lase, F., Halawa, N., Hura, D., Zega, I., & Lase, B. P. (2024). STRATEGI EFEKTIF UNTUK MENGUBAH SKRIPSI MENJADI ARTIKEL ILMIAH: PANDUAN PELATIHAN KONVERSI YANG SUKSES. *JURNAL EDUCATION AND DEVELOPMENT*, 12(1), 206-211.
- Susanti, R. H. (2023). Penulisan Karya Ilmiah sebagai Salah Satu Tools Meningkatkan Kemampuan Berpikir Kritis.
- Bakri, S., Rahayu, S., & Wardari, T. (2023). *Pengabdian Kepada Masyarakat Pembekalan Penulisan Skripsi dan Karya Ilmiah Bagi Mahasiswa Fakultas Ekonomi dan Bisnis Institut Bakti Nusantara*.
- Darmalaksana, W. (2022). *PANDUAN PENULISAN SKRIPSI DAN TUGAS AKHIR*.
- Huda, I. (2021). IMPLEMENTASI NATURAL LANGUAGE PROCESSING (NLP) UNTUK APLIKASI PENCARIAN LOKASI. *Jurnal Nasional Teknologi Terapan (JNTT)*, 3(2), 15. <https://doi.org/10.22146/jntt.35036>
- Ningtyas, K. D., Kurniawan, R., & Armansyah, A. (2023). Penerapan Natural Language Processing Pada Aplikasi Chatbot Info Layanan Kantor Menggunakan Naive Bayes Algorithm. *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, 6(1), 266-273.
- Irham Gufroni, A., Yuliyanti, S., & Nur Fitriani Dewi, E. (2023). MIND (Multimedia Artificial Intelligent Networking Database Ekstraksi Fitur menggunakan Regular Expression pada Naïve Bayes Classifier untuk Analisis Sentimen. *Journal MIND Journal | ISSN*, 8(2), 230–241.
- Maritsa, A., Hanifah Salsabila, U., Wafiq, M., Rahma Anindya, P., & Azhar Ma'shum, M. (2021). Pengaruh Teknologi Dalam Dunia Pendidikan. *Al-Mutharahah: Jurnal Penelitian Dan Kajian Sosial Keagamaan*, 18(2), 91–100. <https://doi.org/10.46781/al-mutharahah.v18i2.303>
- Muhammad Isfa Hany, Adhitya Bhawiyuga, & Ari Kusyanti. (2021). Implementasi Cross Site Scripting Vulnerability Assessment Tools berdasarkan OWASP Code Review. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 5(9), 3745–3753.
- Nurillahwaty, E. (2021). Peran Teknologi dalam Dunia Pendidikan. *Jurnal Keislaman Dan Ilmu Pendidikan*, 3(1), 123–133.
- Pratikasari, C. R., Khairani, E. N., Digananda, S. K., & Ulya, C. (2021). Analisis kesalahan ejaan

- pada Jurnal Imajeri Pendidikan Bahasa Dan Sastra Indonesia Universitas Muhammadiyah Prof. Dr. Hamka. *Jurnal Genre (Bahasa, Sastra, Dan Pembelajarannya)*, 2(2), 71–78. <https://doi.org/10.26555/jg.v2i2.3296>
- Rumapea, H. (2021). Deteksi Kemiripan Artikel Melalui Keywords Dengan Metode Fuzzy String Matching Dalam Natural Language Processing. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, 5(1), 60–66. <https://doi.org/10.46880/jmika.vol5no1.pp60-66>
- Society, A. C. (2023). *KERANGKA ACUAN KERJA (KAK) PENGADAAN JURNAL ONLINE TAHUN 2023*.
- Yunefri, Y., Fadrial, Y. E., & Sutejo, S. (2021). Chatbot Pada Smart Cooperative Oriented Problem Menggunakan Natural Language Processing dan Naive Bayes Classifier. *INTECOMS: Journal of Information Technology and Computer Science*, 4(2), 131–140. <https://doi.org/10.31539/intecom.v4i2.2704>
- Untung, S. (2024). ANALISIS KESALAHAN BERBAHASA PADA LATAR BELAKANG SKRIPSI MAHASISWA UNIVERSITAS KHAIRUN TERNATE. *ESTETIKA: Jurnal Bahasa, Sastra, dan Pengajarannya*, 1(1), 1–11.
- Atiawardani, N. A., Anjaswarni, T., Sulastyawati, S., & Bahari, K. (2025). Hubungan Dukungan Keluarga dengan Tingkat Stres Mahasiswa Rantau yang Mengerjakan Skripsi. *JUMANTIK (Jurnal Ilmiah Penelitian Kesehatan)*, 10(1), 34–41.
- Daawi, M. M., & Nisa, W. I. (2021). Pengaruh dukungan sosial terhadap tingkat stres dalam penyusunan tugas akhir skripsi. *Psikodinamika: Jurnal Literasi Psikologi*, 1(1), 67–75.
- Darmalaksana, W. (2022). *PANDUAN PENULISAN SKRIPSI DAN TUGAS AKHIR*.
- Harahap, N. A. S., Siallagan, L., Simanjuntak, D., Ginting, D. J. P., & Akbar, A. (2025). Analisis Kesalahan Penggunaan Tanda Baca Pada Tugas Makalah Mahasiswa Pendidikan Jasmani Kesehatan Dan Rekreasi Universitas Negeri Medan (UNIMED). *Jurnal Inovasi Penelitian Ilmu Pendidikan Indonesia*, 86–90.
- Priandana, A. D., Andri, A. W., & Ariani, N. (2020). *Rule-Based Natural Language Processing in Volcanic Ash Data Searching System*. *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, 14(2), 99–108. <https://doi.org/10.22146/ijccs.88081>
- Rahman, M., Nowakowski, S., Agrawal, R., Naik, A., Sharafkhaneh, A., & Razjouyan, J.

- (2022). Validation of a natural language processing algorithm for the extraction of the sleep parameters from the polysomnography reports. *Healthcare*, 10(10), 1837. <https://doi.org/10.3390/healthcare10101837>
- Cano, P. M. A., Catigay, J. M. L., Florida, J. M. M., Pua, J. M. R., & Samonte, M. J. C. (2024). Natural language processing of grammar checker tools for academic writing: A systematic literature review. *Journal of Electrical Systems*, 20(7s), 1151–1156.
- Priandana, R. R. K., & Indra. (2024). Rule-Based Natural Language Processing in Volcanic Ash Data Searching Application. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 18(1), 49–60. <https://doi.org/10.22146/ijccs.88081>
- Gufroni, A. I., Prasetya, M. A., & Purnama, A. (2023). Ekstraksi Fitur Menggunakan Regular Expression pada Naïve Bayes Classifier untuk Analisis Sentimen. *Jurnal Teknologi dan Sistem Komputer*, 11(1), 34–40. <https://doi.org/10.14710/jtsiskom.11.1.34-40>
- Ismail, et al. (2022). Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking. *Computers in Industry*.
- Ladas, N., Borchert, F., Franz, S., Rehberg, A., Strauch, N., Sommer, K. K., Marschollek, M., & Gietzelt, M. (2023). Programming techniques for improving rule readability for rule-based information extraction natural language processing pipelines of unstructured and semi-structured medical texts.
- Helen, A., Pradana, A., & Afif, M. (2024). Scientific reference style using rule-based machine learning. *International Journal of Advances in Intelligent Informatics*. <https://doi.org/10.26555/ijain.v9i3.1056>

Lampiran

- Lampiran data mentah AINET : jurnal informatika

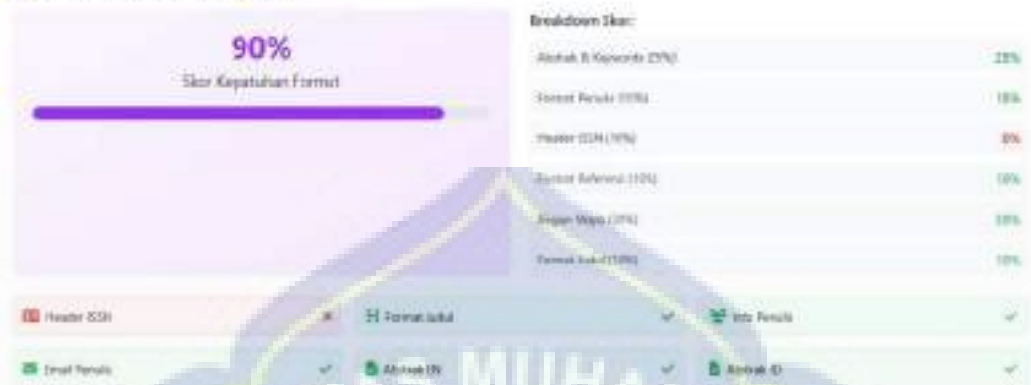


- Lampiran data template ainet



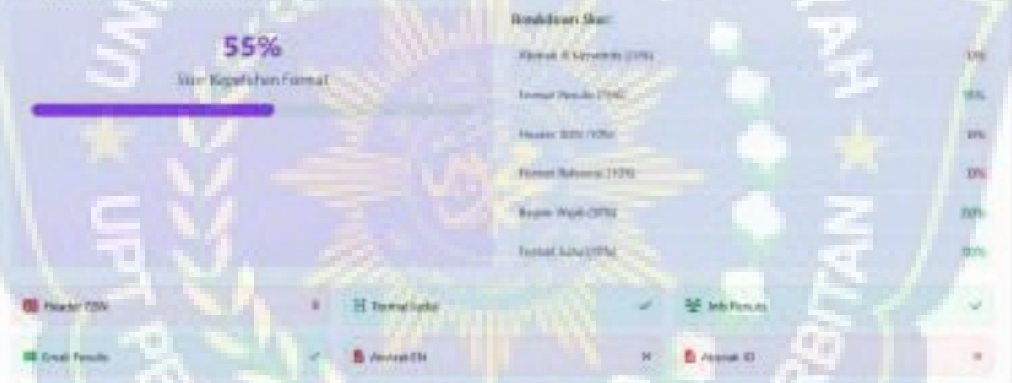
- Lampiran hasil validasi jurnal yang valid

Hasil Validasi Format AINET



- Lampiran hasil validasi jurnal yang tidak valid

Hasil Validasi Format AINET



- Lampiran Source Code

```
import os
import re
import uuid
import time

from flask import Flask, request, render_template,
jsonify, send_from_directory
```

```

from werkzeug.utils import secure_filename

import docx

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize, sent_tokenize

from nltk.stem import PorterStemmer

from sklearn.feature_extraction.text import
TfidfVectorizer

from sklearn.metrics.pairwise import cosine_similarity

import numpy as np

import json

# Download required NLTK data
def ensure_nltk_data():
    """Ensure all required NLTK data is available"""
    resources = [
        ('tokenizers/punkt', 'punkt'),
        ('tokenizers/punkt_tab', 'punkt_tab'),
        ('corpora/stopwords', 'stopwords')
    ]

    for resource_path, resource_name in resources:
        try:

```

```

        nltk.data.find(resource_path)

    except LookupError:

        print(f"Downloading NLTK resource:
{resource_name}")

        nltk.download(resource_name, quiet=True)

# Initialize NLTK data
ensure_nltk_data()

app = Flask(__name__)
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024 #
16MB max file size
app.config['UPLOAD_FOLDER'] = 'uploads'

# Create uploads directory if it doesn't exist
os.makedirs(app.config['UPLOAD_FOLDER'], exist_ok=True)

ALLOWED_EXTENSIONS = {'txt', 'docx'}

# Store uploaded files temporarily (in production, use
Redis or database)
uploaded_files = {}

def allowed_file(filename):

    return '.' in filename and filename.rsplit('.',
1)[1].lower() in ALLOWED_EXTENSIONS

def extract_text_from_file(filepath):

```

```

"""Extract text from .txt or .docx files"""

if filepath.endswith('.docx'):

    try:

        doc = docx.Document(filepath)

        text = '\n'.join([paragraph.text for
        paragraph in doc.paragraphs])

        return text

    except Exception as e:

        return f"Error reading docx: {str(e)}"

elif filepath.endswith('.txt'):

    try:

        with open(filepath, 'r', encoding='utf-8') as
file:

            return file.read()

    except Exception as e:

        return f"Error reading txt: {str(e)}"

    return ""

def nlp_preprocessing_pipeline(text):

    """Comprehensive NLP preprocessing with detailed
    logging"""

    steps = {}

    # Step 1: Original text

```

```

    steps['original_text'] = text[:500] + "..." if
len(text) > 500 else text

    steps['original_length'] = len(text)

# Step 2: Lowercasing
lowercased = text.lower()

    steps['lowercased'] = lowercased[:500] + "..." if
len(lowercased) > 500 else lowercased

# Step 3: Sentence segmentation
try:
    sentences = sent_tokenize(text)
except LookupError:
    # Fallback to simple sentence splitting if punkt
tokenizer fails
    sentences = text.split('. ')
    sentences = [s.strip() + '.' for s in sentences
if s.strip()]

    steps['sentences'] = {
        'count': len(sentences),
        'sample': sentences[:3] if sentences else []
    }

# Step 4: Tokenization
try:

```

```

        tokens = word_tokenize(lowercased)

    except LookupError:

        # Fallback to simple word splitting if punkt
tokenizer fails

        import re

        tokens = re.findall(r'\b\w+\b', lowercased)

    steps['tokens'] = {
        'count': len(tokens),
        'sample': tokens[:20]
    }

    # Step 5: Remove punctuation and special characters
    clean_tokens = [token for token in tokens if
token.isalnum()]

    steps['clean_tokens'] = {
        'count': len(clean_tokens),
        'sample': clean_tokens[:20],
        'removed_count': len(tokens) - len(clean_tokens)
    }

    # Step 6: Remove stopwords

    try:

        stop_words = set(stopwords.words('english'))

```

```

except LookupError:

    # Fallback stopwords if NLTK resource is not
    available

    stop_words = {'i', 'me', 'my', 'myself', 'we',
                  'our', 'ours', 'ourselves', 'you', 'your', 'yours',
                  'yourself', 'yourselves', 'he',
                  'him', 'his', 'himself', 'she', 'her', 'hers',
                  'herself', 'it', 'its', 'itself',
                  'they', 'them', 'their', 'theirs', 'themselves',
                  'what', 'which', 'who', 'whom',
                  'this', 'that', 'these', 'those', 'am', 'is', 'are',
                  'was', 'were', 'be', 'been',
                  'being', 'have', 'has', 'had', 'having', 'do', 'does',
                  'did', 'doing', 'a', 'an', 'the',
                  'and', 'but', 'if', 'or', 'because', 'as', 'until',
                  'while', 'of', 'at', 'by', 'for',
                  'with', 'through', 'during', 'before', 'after',
                  'above', 'below', 'up', 'down',
                  'in', 'out', 'on', 'off', 'over', 'under', 'again',
                  'further', 'then', 'once'}

    # Add Indonesian stopwords

    indonesian_stopwords = {'dan', 'atau', 'yang',
                             'adalah', 'dengan', 'untuk', 'pada', 'dalam', 'dari',

```

```

'ke', 'ini', 'itu', 'akan', 'dapat', 'telah', 'sudah',
'juga', 'tidak', 'ada', 'sebagai'}

stop_words.update(indonesian_stopwords)

filtered_tokens = [word for word in clean_tokens if
word not in stop_words and len(word) > 2]

steps['filtered_tokens'] = {
    'count': len(filtered_tokens),
    'sample': filtered_tokens[:20],
    'stopwords_removed': len(clean_tokens) -
len(filtered_tokens)
}

# Step 7: Stemming
stemmer = PorterStemmer()

stemmed_tokens = [stemmer.stem(word) for word in
filtered_tokens]

steps['stemmed_tokens'] = {
    'count': len(stemmed_tokens),
    'sample': stemmed_tokens[:20]
}

# Step 8: Section segmentation

sections = segment_document_sections(text)

steps['sections'] = sections

```

```

    return steps, ' '.join(filtered_tokens)

def segment_document_sections(text):
    """Segment document into logical sections"""
    sections = {}
    text_lower = text.lower()

    # Define section patterns with positions
    section_patterns = {
        'header': r'(e-issn|p-issn)',
        'title': r'^[A-Z][^.!?]*$', # First capitalized
        'authors': r'[A-Z][a-z]+\s+[A-Z][a-z]+.*@.*',
        'abstract_en': r'abstract\s*[:\n]',
        'abstract_id': r'abstrak\s*[:\n]',
        'keywords_en': r'keywords\s*:',
        'keywords_id': r'kata kunci\s*:',
        'introduction': r'(introduction|pendahuluan)',
        'methodology': r'(methodology|methods?|metode
        penelitian)',

        'results':
        r'(results?|hasil).*?(discussion|pembahasan)',

        'conclusion': r'(conclusion|kesimpulan)',

```

```

        'references': r'(references?|referensi|daftar
pustaka)'
    }

    for section_name, pattern in
section_patterns.items():

        matches = list(re.finditer(pattern, text_lower,
re.IGNORECASE | re.MULTILINE))

        sections[section_name] = {
            'found': len(matches) > 0,
            'count': len(matches),
            'positions': [{'start': m.start(), 'end':
m.end(), 'text': m.group()} for m in matches]
        }

    return sections

def validate_ainet_header(text):

    """Validate AINET journal header format with detailed
logging"""

    patterns = [

        r'e-ISSN\s*:\s*2686\s*[--]\s*1917',

        r'p-ISSN\s*:\s*2657\s*[--]\s*0653'

    ]

    results = []

```

```

for i, pattern in enumerate(patterns):

    match = re.search(pattern, text, re.IGNORECASE)

    results.append({

        'pattern': pattern,

        'type': 'e-ISSN' if i == 0 else 'p-ISSN',

        'match': bool(match),

        'position': match.start() if match else None,

        'text': match.group() if match else None

    })

return {

    'valid': all(r['match'] for r in results),

    'details': results

}

def validate_title_format(text):

    """Validate title format with detailed analysis"""

    lines = text.split('\n')

    title_candidates = []

    for i, line in enumerate(lines[:15]): # Check first
15 lines

        line = line.strip()

```

```

        if 10 < len(line) < 200 and not
re.search(r'(abstract|author|email|@)', line,
re.IGNORECASE):

        words = line.split()

        if len(words) > 2:

            # Check title case

            title_case_words = sum(1 for word in
words if word[0].isupper() or word.lower() in ['and',
'or', 'the', 'of', 'in', 'on', 'at', 'to', 'for',
'with'])

            title_case_ratio = title_case_words /
len(words)

            title_candidates.append({

                'line_number': i + 1,

                'text': line,

                'word_count': len(words),

                'title_case_ratio': title_case_ratio,

                'is_valid': title_case_ratio > 0.6

            })

        valid_titles = [t for t in title_candidates if
t['is_valid']]

        return {

            'valid': len(valid_titles) > 0,

```

```

        'candidates': title_candidates,

        'best_match': valid_titles[0] if valid_titles
    else None

```

```

    }

def validate_authors_format(text):

    """Validate authors format with detailed pattern
    matching"""

    patterns = {

        'author_names': r'([A-Z][a-zA-Z\s]+(?:\d+|{1:2:4567890}))+',

        'email_addresses': r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}',

        'affiliations': r'\d+\s*[A-Za-z\s,]+(University|Institut|College|Department)',

        'numbered_authors': r'([A-Z][a-zA-Z\s]+(?:\d+|{1:2:4567890}\d))+',

    }

    results = {}

    for pattern_name, pattern in patterns.items():

        matches = list(re.finditer(pattern, text,
            re.IGNORECASE))

        results[pattern_name] = {

            'count': len(matches),

```

```

        'matches': [{'text': m.group(), 'position':
m.start()} for m in matches[:5]] # Limit to first 5
    }

    return {

        'authors_detected':
results['author_names']['count'] > 0,

        'emails_found':
results['email_addresses']['count'] > 0,

        'affiliations_found':
results['affiliations']['count'] > 0,

        'numbered_format':
results['numbered_authors']['count'] > 0,

        'details': results
    }

def validate_abstract_keywords(text):
    """Validate abstract and keywords with position
tracking"""

    patterns = {

        'abstract_en':
r'Abstract\s*[:\n] (.*) (?=Keywords|Kata
kunci|Abstrak|Introduction|Pendahuluan|\n\n|\Z)',

        'abstract_id':
r'Abstrak\s*[:\n] (.*) (?=Keywords|Kata
kunci|Abstract|Introduction|Pendahuluan|\n\n|\Z)',

```

```

        'keywords_en': r'Keywords\s*:(.*) (?=Kata
kunci|Introduction|Pendahuluan| \n\n| \Z)',

        'keywords_id': r'Kata
kunci\s*:(.*) (?=Keywords|Introduction|Pendahuluan| \n\n| \
Z)'

    }

    results = {}
    for pattern_name, pattern in patterns.items():
        match = re.search(pattern, text, re.IGNORECASE |
re.DOTALL)
        if match:
            content = match.group(1).strip() if
len(match.groups()) > 0 else match.group()
            word_count = len(content.split()) if content
else 0
            results[pattern_name] = {
                'found': True,
                'content': content[:200] + "..." if
len(content) > 200 else content,
                'word_count': word_count,
                'position': match.start()
            }
        else:

```

```

        results[pattern_name] = {

            'found': False,

            'content': '',

            'word_count': 0,

            'position': None

        )

    return results

def validate_required_sections(text):

    """Validate presence of required sections with regex
    logging"""

    sections = {

        'pendahuluan': {

            'patterns': [r'\bPendahuluan\b',
r'\bIntroduction\b'],

            'required': True

        },

        'metode': {

            'patterns': [r'\bMetode Penelitian\b',
r'\bMethodology\b', r'\bMethods\b'],

            'required': True

        },

        'hasil': {

```

```
        'patterns': [r'\bHasil dan Pembahasan\b',  
r'\bResults and Discussion\b', r'\bResults\b'],  
  
        'required': True
```

```
    },
```

```
    'kesimpulan': {
```

```
        'patterns': [r'\bKesimpulan\b',  
r'\bConclusion\b'],
```

```
        'required': True
```

```
    },
```

```
    'referensi': {
```

```
        'patterns': [r'\bReferensi\b',  
r'\bReferences\b', r'\bDaftar Pustaka\b'],
```

```
        'required': True
```

```
    }
```

```
}
```

```
regex_log = []
```

```
for section_key, section_data in sections.items():
```

```
    section_found = False
```

```
    section_matches = []
```

```
    for pattern in section_data['patterns']:
```

```

        matches = list(re.finditer(pattern, text,
re.IGNORECASE))

        for match in matches:

            section_matches.append({
                'pattern': pattern,
                'position': match.start(),
                'text': match.group()
            })

            section_found = True

            regex_log.append({
                'section': section_key,
                'pattern': pattern,
                'match': len(matches) > 0,
                'count': len(matches),
                'positions': [m.start() for m in matches]
            })

            sections[section_key]['found'] = section_found

            sections[section_key]['matches'] =
section_matches

        return sections, regex_log

def validate_reference_format(text):

```

```

"""Validate reference formatting with detailed
pattern analysis"""

patterns = {
    'ieee_numbered': r'\{\d+\}',
    'apa_style': r'\([A-Za-z]+,?\s*\d{4}\)',
    'numbered_list': r'^\s*\d+\.\s+',
    'doi_pattern': r'doi:\s*[0-9.]+/[^\s]+',
    'url_pattern': r'https?:\/\/[^\s]+'
}

results = {}

for pattern_name, pattern in patterns.items():
    matches = list(re.finditer(pattern, text,
re.IGNORECASE | re.MULTILINE))

    results[pattern_name] = {
        'count': len(matches),
        'matches': [{'text': m.group(), 'position':
m.start()} for m in matches[:10]] # Limit to first 10
    }

# Determine primary reference style

ieee_count = results['ieee_numbered']['count']

apa_count = results['apa_style']['count']

```

```

        primary_style = 'ieee' if ieee_count > apa_count else
'apa' if apa_count > 0 else 'unknown'

    return {

        'ieee_style': ieee_count > 0,

        'apa_style': apa_count > 0,

        'primary_style': primary_style,

        'ieee_count': ieee_count,

        'apa_count': apa_count,

        'details': results

    }

def perform_comprehensive_ainet_validation(template_text,
submission_text):

    """Perform complete AINET format validation with
detailed logging"""

    # NLP preprocessing for both documents
    template_nlp, template_processed =
nlp_preprocessing_pipeline(template_text)

    submission_nlp, submission_processed =
nlp_preprocessing_pipeline(submission_text)

    # Validate submission document with detailed logging

    validation_results = {

        'header_issn':
validate_ainet_header(submission_text),

```

```

        'title_format':
validate_title_format(submission_text),

        'authors_format':
validate_authors_format(submission_text),

        'abstract_keywords':
validate_abstract_keywords(submission_text),

        'required_sections': {},
        'reference_format':
validate_reference_format(submission_text)
    }

    # Get sections and regex log
    sections, regex_log =
validate_required_sections(submission_text)
    validation_results['required_sections'] = sections

    # Calculate overall compliance score
    score_components = []

    # Header ISSN (10%)
    score_components.append(10 if
validation_results['header_issn']['valid'] else 0)

    # Title format (10%)
    score_components.append(10 if
validation_results['title_format']['valid'] else 0)

```

```

# Authors format (15%)

authors = validation_results['authors_format']

author_score = 0

if authors['authors_detected']:
    author_score += 5

if authors['emails_found']:
    author_score += 5

if authors['numbered_format']:
    author_score += 5

score_components.append(author_score)

# Abstract & Keywords (25%)

abstract_kw = validation_results['abstract_keywords']

abstract_score = 0

if abstract_kw['abstract_en']['found']:
    abstract_score += 8

if abstract_kw['abstract_id']['found']:
    abstract_score += 8

if abstract_kw['keywords_en']['found']:
    abstract_score += 4.5

if abstract_kw['keywords_id']['found']:

```

```

        abstract_score += 4.5

    score_components.append(abstract_score)

    # Required sections (30%)

    section_score = sum(6 for section in
sections.values() if section['found'])

    score_components.append(section_score)

    # Reference format (10%)

    ref_format = validation_results['reference_format']
    ref_score = 10 if ref_format['ieee_style'] else 0

    score_components.append(ref_score)

    overall_score = sum(score_components)

    # Prepare comprehensive output
    output = {
        'overall_score': overall_score,
        'score_breakdown': {
            'header_issn': score_components[0],
            'title_format': score_components[1],
            'authors_format': score_components[2],
            'abstract_keywords': score_components[3],
            'required_sections': score_components[4],
            'reference_format': score_components[5]

```

```

    },
    'validation_results': validation_results,
    'regex_log': regex_log,
    'nlp_steps': {
        'template': template_nlp,
        'submission': submission_nlp
    },
    'missing_sections': [k for k, v in
sections.items() if not v['found']],
    'template_word_count':
len(template_text.split()),
    'submission_word_count':
len(submission_text.split())
}
return output

def cleanup_old_files():
    """Clean up files older than 1 hour"""
    current_time = time.time()

    for session_id, file_info in
list(uploaded_files.items()):

        if current_time - file_info['timestamp'] >
3600: # 1 hour

```

```

try:
    if
os.path.exists(file_info['template_path']):

        os.remove(file_info['template_path'])

    if
os.path.exists(file_info['target_path']):

        os.remove(file_info['target_path'])
        del uploaded_files[session_id]
except Exception as e:
    print(f"Error cleaning up files: {e}")

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/upload_files', methods=['POST'])
def upload_files():
    """Handle file upload separately"""

    if 'template' not in request.files or 'target' not in
request.files:

        return jsonify({'success': False, 'error': 'Kedua
file template dan submission diperlukan'}), 400

    template_file = request.files['template']

    target_file = request.files['target']

```

```

        if template_file.filename == '' or
target_file.filename == '':

            return jsonify({'success': False, 'error': 'Tidak
ada file yang dipilih'}), 400

        if not (allowed_file(template_file.filename) and
allowed_file(target_file.filename)):

            return jsonify({'success': False, 'error': 'Hanya
file .txt dan .docx yang diizinkan'}), 400

    try:

        # Clean up old files periodically
cleanup_old_files()

        # Generate unique session ID
session_id = str(uuid.uuid4())

        # Save uploaded files with unique names

        template_filename =
f"{session_id}_template_{secure_filename(template_file.fi
lename)}"

        target_filename =
f"{session_id}_target_{secure_filename(target_file.filena
me)}"

        template_path =
os.path.join(app.config['UPLOAD_FOLDER'],
template_filename)

```

```

        target_path =
os.path.join(app.config['UPLOAD_FOLDER'],
target_filename)

        template_file.save(template_path)

        target_file.save(target_path)

        # Store file paths in memory (use Redis or
database in production)
        uploaded_files[session_id] = {
            'template_path': template_path,
            'target_path': target_path,
            'timestamp': time.time()
        }

        return jsonify({
            'success': True,
            'filePaths': {
                'session_id': session_id,
                'template_name': template_file.filename,
                'target_name': target_file.filename
            }
        })

    except Exception as e:

```

```

        return jsonify({'success': False, 'error':
f'Gagal mengunggah file: {str(e)}'}), 500

@app.route('/validate', methods=['POST'])
def validate_documents():
    """Handle document validation after upload"""
    try:
        data = request.get_json()
        session_id = data.get('session_id')
        if not session_id or session_id not in
uploaded_files:
            return jsonify({'success': False, 'error':
'File tidak ditemukan atau sudah kadaluarsa'}),
400

        file_info = uploaded_files[session_id]
        template_path = file_info['template_path']
        target_path = file_info['target_path']

        # Check if files still exist
        if not os.path.exists(template_path) or not
os.path.exists(target_path):

            return jsonify({'success': False, 'error':
'File tidak ditemukan'}), 400

        # Extract text from files

```

```

        template_text =
extract_text_from_file(template_path)

        target_text =
extract_text_from_file(target_path)

        if template_text.startswith('Error') or
target_text.startswith('Error'):

            return jsonify({'success': False, 'error':
'Gagal mengekstrak teks dari file'}), 500

            # Perform comprehensive AINET validation
            results =
perform_comprehensive_ainet_validation(template_text,
target_text)

            # Clean up files after validation
            try:

                os.remove(template_path)

                os.remove(target_path)

                del uploaded_files[session_id]

            except Exception as e:

                print(f"Error cleaning up files: {e}")

            return jsonify({

                'success': True,

                'results': results

```

```
1)

except Exception as e:

    return jsonify({'success': False, 'error':
f'Pemrosesan gagal: {str(e)}'}, 500

if __name__ == '__main__':

    app.run(debug=True, host='0.0.0.0', port=5000)
```



Lampiran surat keterangan bebas plagiat



MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN

Alamat Kantor: Jl. Sultan Alauddin No. 255 Makassar 90222 Telp (0411) 881583, 881583, Fax (0411) 881588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menyatakan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Fatimah Az Zahra

Nim : 105841100320

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	9%	10 %
2	Bab 2	4%	25 %
3	Bab 3	3%	10 %
4	Bab 4	0%	10 %
5	Bab 5	3%	5 %

Dinyatakan telah lulus cek plagiat yang dilakukan oleh UPT- Perpustakaan dan Penerbitan
Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan
seperlunya.

Makassar, 23 Agustus 2025

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



Jl. Sultan Alauddin no 255 makassar 90222
Telepon (0411)881583, 881583, fax (0411)881588
Website: www.library.umh.ac.id
E-mail: perpus@umh.ac.id

Lampiran Hasil uji plagiasi



Submission date: 28-Aug-2025 08:58AM (UTC+0700)

Submission ID: 2736535860

File name: BAB_1_6.docx (78.98K)

Word count: 719

Character count: 4845

BAB I FATIMAH AZZAHRA 105841100320

ORIGINALITY REPORT

9%

SIMILARITY INDEX

9%

INTERNET SOURCES

0%

PUBLICATIONS

5%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

4%

★ repository.untag-sby.ac.id

Internet Source

Exclude quotes

On

Exclude matches

+ 2%

Exclude bibliography

On

BAB II FATIMAH AZZAHRA

105841100320

by Tahap Tutup

Submission date: 28-Aug-2015 08:59AM (UTC+0700)

Submission ID: 2736596475

File name: BAB_II_4.docx (201.16K)

Word count: 1172

Character count: 8074

BAB II FATIMAH AZZAHRA 105841100320

ORIGINALITY REPORT

4%

SIMILARITY INDEX

4%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

WATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

4%

★ digilibadmin.unismuh.ac.id

Internet Source

Exclude quotes

On

Exclude matches

On

Exclude bibliography

On



● BAB III FATIMAH AZZAHRA

105841100320

by Tahap Tutup

●

Submission date: 28-Aug-2025 09:00AM (UTC+0100)

Submission ID: 2736396940

File name: BAB_III_4.docx (959.9K)

Word count: 1090

Character count: 7209

BAB III FATIMAH AZZAHRA 105841100320

ORIGINALITY REPORT

3%	2%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCES PRINTED)

3%

★ ejournal.unisbablitar.ac.id

Internet Source

Exclude quotes

On

Exclude bibliography

On

Exclude matches

< 2%

The logo of Universitas Muhammadiyah Makassar is a blue pentagon with a yellow border. Inside the pentagon, there is a yellow sunburst in the center. The text "UNIVERSITAS MUHAMMADIYAH MAKASSARA" is written in yellow around the top half of the pentagon. The text "UPT PERPUSTAKAAN DAN PENERBITAN" is written in yellow around the bottom half of the pentagon. There are two yellow stars on the left and right sides of the pentagon.

● BAB IV FATIMAH AZZAHRA

105841100320

by Tahap Tutup

Submission date: 27-Aug-2025 09:55AM (UTC+0700)

Submission ID: 2735948683

File name: BAB_IV_1.docx (482.5K)

Word count: 3156

Character count: 19895

BAB IV FATIMAH AZZAHRA 105841100320

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes

On

Exclude bibliography

On

Exclude matches

< 2%



BAB V FATIMAH AZZAHRA

105841100320

by Tahap Tutup

Submission date: 27-Aug-2025 09:56AM (UTC+0700)

Submission ID: 2735948891

File name: BAB_V_1.docx (73.68K)

Word count: 234

Character count: 1582

BAB V FATIMAH AZZAHRA 105841100320

ORIGINALITY REPORT

3%

SIMILARITY INDEX

3%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

3%

★ github.com

Internet Source

Exclude quotes

On

Exclude bibliography

On

Exclude matches

< 2%