



# Integrasi Algoritma Cosine Similarity pada Sistem Informasi LPPM untuk Otomatisasi Deteksi Redundansi Judul

Jeffadha Rhenggajati Umbaratama\*, Wakhid Kurniawan

Fakultas Sains Teknologi dan Peternakan, Program Studi Informatika, Universitas Muhammadiyah Karanganyar, Karanganyar, Indonesia

Email: <sup>1,\*</sup>jeffadharhenggajatiu@gmail.com, <sup>2</sup>kurniawan.wk48@gmail.com

Email Penulis Korespondensi: jeffadharhenggajatiu@gmail.com

**Abstrak**—Duplikasi judul penelitian merupakan permasalahan krusial yang sering terjadi di lingkungan akademik, khususnya pada tata kelola Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM). Proses verifikasi judul yang berjalan secara manual saat ini dinilai tidak efisien, memakan waktu operasional yang besar, dan sangat rentan terhadap *human error*, sehingga berpotensi meloloskan usulan dengan tingkat kemiripan substansial. Penelitian ini bertujuan untuk merancang dan mengembangkan Sistem Informasi LPPM berbasis web yang terintegrasi dengan fitur cerdas pendeteksi kesamaan judul secara otomatis untuk studi kasus di Universitas Muhammadiyah Karanganyar. Solusi yang ditawarkan adalah pembangunan sistem terdistribusi menggunakan arsitektur *microservices*, yang mengintegrasikan layanan web kerangka kerja Laravel dan layanan komputasi *Natural Language Processing* (NLP) Python (FastAPI) melalui protokol komunikasi *REST API*. Metode pengembangan perangkat lunak mengadopsi *Agile Development* guna mengakomodasi perubahan kebutuhan secara iteratif. Kontribusi utama penelitian ini adalah terciptanya instrumen validasi akademik berbasis arsitektur terdistribusi yang mengisolasi beban kerja komputasi teks dari *server* utama. Penerapan algoritma *Cosine Similarity* yang dipadukan dengan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) serta pustaka leksikal Sastrawi digunakan untuk mengkalkulasi tingkat kemiripan teks secara presisi. Hasil pengujian fungsional *Black Box* dengan teknik *Equivalence Partitioning* menunjukkan integrasi API berjalan tanpa galat. Mengingat fokus penelitian pada integrasi arsitektur sistem, evaluasi algoritma dilakukan menggunakan dataset sintesis terkontrol yang menyimulasikan berbagai teknik manipulasi teks. Hasil pengujian menunjukkan sistem mampu mengidentifikasi modifikasi sintaksis dasar secara presisi. Namun demikian, sistem ini secara transparan diakui memiliki keterbatasan komputasi (penurunan deteksi hingga 12,74%) dalam mengenali kemiripan semantik yang murni menggunakan padanan sinonim. Implementasi teknologi ini diharapkan dapat membantu pengelola LPPM dalam memvalidasi orisinalitas usulan proposal secara efisien, objektif, dan *real-time*.

**Kata Kunci:** *Agile Development; Cosine Similarity; Natural Language Processing; REST API; Sistem Informasi*

**Abstract**—Research title duplication is a crucial problem that frequently occurs in the academic environment, particularly in the governance of the Institute for Research and Community Service (LPPM). The current manual title verification process is considered inefficient, time-consuming, and highly prone to human error, potentially allowing proposals with substantial similarity to pass. This study aims to design and develop a web-based LPPM Information System integrated with a smart automatic title similarity detection feature for a case study at Universitas Muhammadiyah Karanganyar. The proposed solution is the development of a distributed system using a microservices architecture, integrating a Laravel framework web service and a Python (FastAPI) Natural Language Processing (NLP) computational service via REST API communication protocol. The software development method adopts Agile Development to accommodate requirement changes iteratively. The primary contribution of this research is the creation of an academic validation instrument based on a distributed architecture that isolates the text computational workload from the main server. The application of the Cosine Similarity algorithm combined with Term Frequency-Inverse Document Frequency (TF-IDF) weighting and the Sastrawi lexical library is used to calculate text similarity precisely. Black Box functional testing using the Equivalence Partitioning technique shows that the API integration runs without errors. Given the research focus on system integration, the algorithm evaluation was conducted using a controlled synthetic dataset simulating various text manipulation techniques. The test results show that the system can precisely identify basic syntactic modifications. However, the system transparently acknowledges a computational limitation (a drop in detection to 12.74%) in recognizing semantic similarities purely based on synonyms. The implementation of this technology is expected to help LPPM administrators validate the originality of proposals efficiently, objectively, and in real-time.

**Keywords:** *Agile Development; Cosine Similarity; Natural Language Processing; REST API; Information System*

## 1. PENDAHULUAN

Perkembangan teknologi informasi di era transformasi digital telah membawa dampak signifikan terhadap efisiensi tata kelola administrasi di lingkungan perguruan tinggi. Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) merupakan unit strategis yang memiliki peran vital dalam mengelola, memantau, dan mengarsipkan seluruh luaran intelektual dosen maupun mahasiswa sebagai bagian dari perwujudan Tri Dharma Perguruan Tinggi (Afriansyah, 2022). Transformasi layanan LPPM dari sistem manual menuju sistem informasi berbasis *web* telah terbukti mampu meningkatkan efisiensi manajemen data secara signifikan (Nababan et al., 2022). Sejalan dengan hal tersebut, digitalisasi layanan LPPM juga berdampak positif pada peningkatan motivasi riset dosen internal (Sugiyatno et al., 2023). Meskipun demikian, seiring dengan meningkatnya volume usulan proposal penelitian setiap semesternya, muncul tantangan baru dalam menjaga integritas akademik, khususnya terkait orisinalitas judul penelitian yang diajukan.

Masalah utama yang sering dihadapi oleh pengelola LPPM adalah sulitnya melakukan validasi judul secara cepat dan akurat. Proses verifikasi judul untuk menghindari redundansi atau kemiripan dengan penelitian terdahulu sering kali masih mengandalkan pengecekan manual atau pencarian kata kunci sederhana pada basis data. Metode manual ini tidak hanya memakan waktu operasional yang besar, tetapi juga memiliki risiko tinggi terhadap *human error* dan



ketidakkonsistenan hasil verifikasi. Di sinilah penerapan *Natural Language Processing* (NLP) di bidang pendidikan menjadi sangat relevan sebagai instrumen cerdas untuk memproses data tekstual secara otomatis dan sistematis (Khurana et al., 2022). Ketangguhan teknik pemrosesan teks semacam ini pada dasarnya memiliki cakupan yang luas dan telah terbukti andal dalam berbagai studi klasifikasi informasi maupun analisis teks menggunakan *machine learning* (Bintang et al., 2025; Romadloni & Kurniawan, 2025).

Dalam upaya mengatasi masalah duplikasi dokumen, berbagai algoritma kemiripan teks telah dikembangkan dan diuji. Sebagai contoh, sebuah penelitian membuktikan efektivitas algoritma *Cosine Similarity* dalam mendeteksi kemiripan topik judul secara komputasional (Sutikno, 2021). Lebih lanjut, implementasi metode yang sama juga berhasil mengidentifikasi indikasi plagiarisme pada dokumen jurnal ilmiah secara presisi (Lumbansiantar et al., 2023). Implementasi nyata dari algoritma ini juga telah dikembangkan dalam membangun aplikasi deteksi kemiripan teks khusus untuk dokumen tugas akhir mahasiswa (Prismadana, 2023). Berdasarkan analisis perbandingan, *Cosine Similarity* yang dipadukan dengan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) menunjukkan stabilitas akurasi yang lebih unggul untuk teks berbahasa Indonesia dibandingkan metode kemiripan lainnya (Pawestri & Suyanto, 2024). Hal ini dikarenakan *Cosine Similarity* mampu melakukan perhitungan berdasarkan sudut vektor antarkata, sehingga hasil kalkulasinya tetap konsisten meskipun terdapat perbedaan panjang karakter pada dokumen yang dibandingkan (Ayash et al., 2025). Keunggulan matematis ini memberikan performa yang lebih baik dalam menangkap kemiripan leksikal judul jika dibandingkan dengan pendekatan *Jaro-Winkler Distance* yang lebih terfokus pada deteksi kesalahan pengetikan atau *typo* (Syahrullah et al., 2024).

Selain aspek algoritma, tantangan dalam pengembangan sistem informasi modern terletak pada arsitektur perangkat lunak dan infrastruktur *server*. Saat ini, banyak pengembang menggunakan *framework* Laravel untuk membangun aplikasi akademik yang kompleks karena fitur keamanannya yang mumpuni (Nasution & Maulana, 2024). Fleksibilitas Laravel juga memungkinkan pengembangan *REST API* yang mempermudah sistem untuk berkomunikasi dengan layanan komputasi lain secara terintegrasi (Otieno & Araka, 2025). Namun, untuk mengatasi kelemahan metodologi pengembangan yang bersifat sekuensial kaku, tren saat ini lebih mengarah pada adopsi kerangka kerja Agile (Pertiwi et al., 2023). Kerangka kerja ini dipilih karena metodologi konvensional sering kali kurang adaptif jika terdapat perubahan kebutuhan di tengah proses pengembangan, seperti adanya permintaan revisi fitur mendadak dari pihak LPPM. Untuk mengatasi isu fleksibilitas dan integrasi antarbahasa pemrograman, sebuah studi menekankan pentingnya arsitektur sistem yang modular guna meningkatkan kinerja aplikasi *web* (Ekaputra et al., 2023). Pemanfaatan arsitektur terdistribusi memungkinkan layanan utama dan layanan komputasi berjalan secara independen tanpa hambatan konfigurasi yang rumit.

Tinjauan terhadap penelitian terdahulu menunjukkan adanya celah penelitian (*gap analysis*) yang perlu diisi. Beberapa sistem informasi LPPM yang telah dikembangkan sebelumnya memiliki fokus yang berbeda-beda. Terdapat penelitian yang berfokus pada manajemen data penelitian secara umum pada Pusat Penelitian dan Pengabdian Masyarakat (P3M) di perguruan tinggi (Rizal et al., 2024). Sementara itu, penelitian lain lebih menitikberatkan pada pengembangan alur pelaporan riset agar lebih terstruktur secara digital (Toscany et al., 2022). Pendekatan administratif serupa juga diusulkan untuk mempercepat birokrasi pengabdian di lingkungan universitas (Syafii & Ridha, 2023). Upaya penyempurnaan sistem informasi juga dilakukan melalui pembuatan modul skema penelitian spesifik (Martua & Toba, 2024). Akan tetapi, meskipun sistem-sistem tersebut sudah sangat tangguh secara administratif, mayoritas belum dilengkapi dengan fitur cerdas yang mampu mendeteksi kemiripan judul secara otomatis dan *real-time*. Di sisi lain, riset-riset mengenai *Cosine Similarity* sering kali hanya berhenti pada tahap pengujian algoritma secara teoretis tanpa diintegrasikan ke dalam arsitektur sistem informasi yang mampu menangani beban kerja nyata. Dalam konteks pemrosesan *Natural Language Processing* (NLP), pendekatan sistem monolitik memiliki kelemahan komputasional pada manajemen beban kerja (*workload*). Apabila komputasi matriks yang berat dipaksakan berjalan pada *server web* utama yang beroperasi secara sinkronus, hal tersebut berisiko memicu *bottleneck* memori, terutama pada masa puncak (*peak time*) pengajuan proposal di LPPM. Oleh karena itu, integrasi arsitektur terdistribusi antara Laravel dan FastAPI (Python) secara spesifik diperlukan guna memecahkan kesenjangan beban kerja tersebut.

Kesenjangan tersebut memicu kebutuhan akan sebuah platform yang menggabungkan keandalan manajemen administratif dengan kecerdasan komputasi teks. Penelitian ini hadir untuk merancang bangun Sistem Informasi LPPM yang tidak hanya mengelola alur pengusulan proposal, tetapi juga memiliki modul deteksi kemiripan judul otomatis yang terintegrasi di dalamnya. Perangkat lunak ini dikembangkan menggunakan metodologi *Agile Development* agar proses pengembangannya lebih responsif terhadap kebutuhan pengguna dan perubahan fitur secara iteratif. Untuk memastikan kualitas fungsional sistem, pengujian dilakukan melalui metode *Black Box* dengan teknik *Equivalence Partitioning* guna menjamin setiap *input* data menghasilkan *output* yang sesuai dengan logika bisnis (Maulana et al., 2024).

Menanggapi kelemahan arsitektur monolitik sebelumnya, pembaruan sekaligus kontribusi utama pada penelitian ini terletak pada integrasi arsitektur sistem hibrida terdistribusi yang memisahkan beban kerja komputasi teks. Layanan antarmuka *web* yang dibangun menggunakan Laravel dapat berjalan beriringan dengan layanan komputasi *Natural Language Processing* (NLP) berbasis Python (FastAPI) secara efisien. Penggunaan algoritma *Cosine Similarity* pada sistem ini memberikan visualisasi persentase kemiripan judul secara otomatis dan transparan. Adapun kontribusi lanjutan dari penelitian ini adalah terciptanya sebuah instrumen validasi akademik yang objektif, terukur, dan cerdas secara komputasional bagi pengelola LPPM di Universitas Muhammadiyah Karanganyar. Melalui pemisahan beban kerja komputasi ini, sistem tidak hanya mengatasi permasalahan duplikasi usulan proposal, tetapi juga memastikan

infrastruktur *server* utama tetap stabil meskipun memproses matriks teks dalam jumlah besar. Tujuan akhir dari penelitian ini adalah menjaga integritas, kualitas, serta orisinalitas publikasi ilmiah sejak tahap awal pengajuan proposal.

## 2. METODOLOGI PENELITIAN

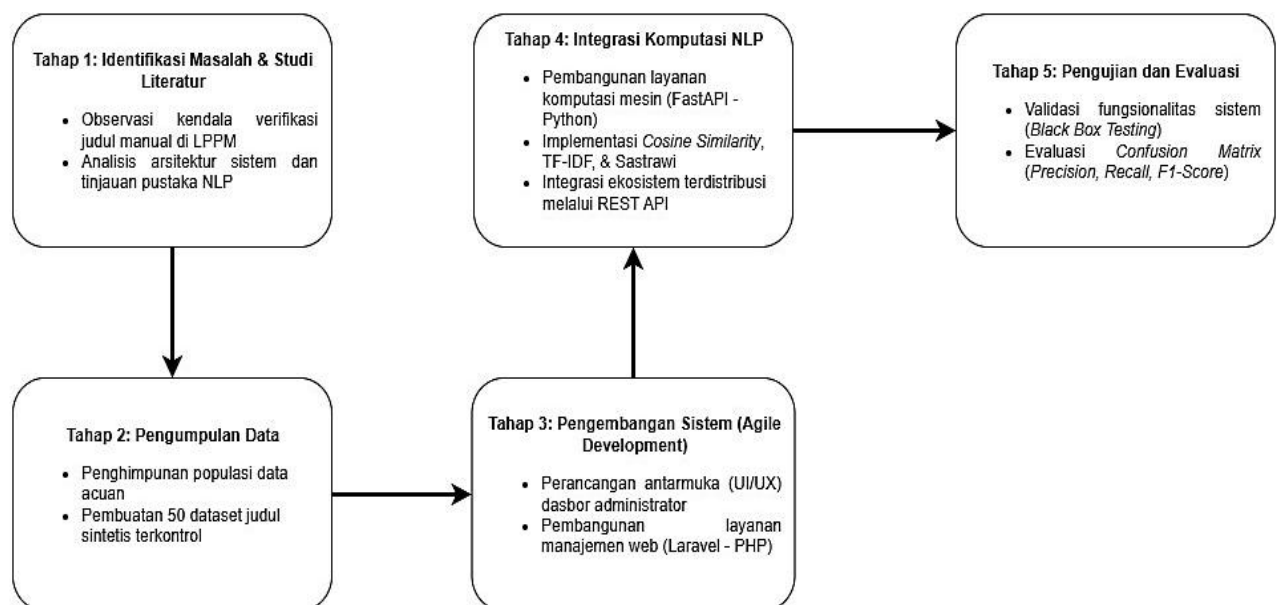
Penelitian ini dilaksanakan melalui serangkaian tahapan sistematis yang mencakup pengembangan perangkat lunak, perancangan arsitektur sistem, pemrosesan teks, serta pengujian algoritma. Untuk memastikan sistem informasi yang dibangun relevan dengan kebutuhan pengguna yang dinamis, pendekatan metodologi konvensional seperti model Waterfall ditinggalkan, dan digantikan dengan metodologi Agile yang terbukti lebih adaptif terhadap perubahan (Rahmawan et al., 2025).

### 2.1 Kerangka Dasar Penelitian

Penelitian ini merupakan penelitian terapan (*applied research*) yang dilaksanakan di Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Muhammadiyah Karanganyar. Objek utama dalam penelitian ini adalah data tekstual berupa usulan judul proposal penelitian. Variabel yang diukur adalah tingkat persentase kemiripan (*similarity*) antardokumen proposal. Teknik analisis data yang digunakan dalam pengujian ini bertumpu pada pemrosesan *Natural Language Processing* (NLP) dengan menerapkan algoritma *Cosine Similarity* dan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF). Adapun populasi data yang digunakan dalam evaluasi komputasi algoritma ini berjumlah 50 dokumen judul proposal sintetis yang di-*generate* dan disimpan dalam basis data untuk mensimulasikan berbagai teknik manipulasi teks yang umum dilakukan oleh pengusul.

### 2.2 Tahapan Penelitian

Pelaksanaan penelitian ini dilakukan melalui alur metodologi ilmiah yang terstruktur, bukan sekadar siklus pengembangan perangkat lunak. Secara garis besar, tahapan penelitian diilustrasikan melalui bagan alur pada Gambar 1.



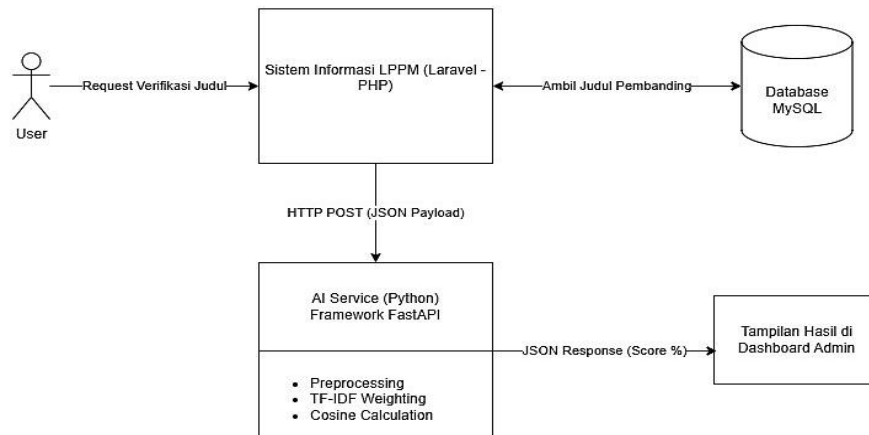
**Gambar 1.** Bagan Alur Tahapan Penelitian

Berdasarkan Gambar 1, penelitian diawali dengan (1) Identifikasi Masalah dan Studi Literatur, di mana observasi dilakukan terhadap kendala verifikasi manual di LPPM. Tahap selanjutnya adalah (2) Pengumpulan Data, yang berfokus pada penghimpunan populasi data judul sintetis untuk skenario pengujian. Setelah kebutuhan data terpenuhi, masuk ke tahap (3) Pengembangan Sistem (Agile Development). Pada fase inilah metodologi *Agile* diimplementasikan untuk merekayasa perangkat lunak melalui siklus iteratif (*Sprint*). Tahap berikutnya adalah (4) Integrasi Komputasi NLP, di mana mesin pemrosesan teks Python disambungkan ke *server* utama melalui REST API. Tahap terakhir adalah (5) Pengujian dan Evaluasi, yang mencakup validasi *Black Box* fungsional serta pengukuran sensitivitas akurasi algoritma.

### 2.3 Arsitektur Sistem Terdistribusi via REST API

Guna meningkatkan performa dan modularitas sistem, penelitian ini menerapkan arsitektur sistem terdistribusi. Layanan manajemen administrasi (Laravel) dan layanan komputasi kemiripan (FastAPI) dipisahkan secara logika namun diintegrasikan melalui protokol *Representational State Transfer Application Programming Interface* (REST API) (Otieno & Araka, 2025). Arsitektur ini dirancang sedemikian rupa agar aplikasi Laravel bertugas penuh menangani

request HTTP dari pengguna dan menyajikan antarmuka visual. Apabila pengguna melakukan *input* judul proposal baru, sistem Laravel akan mengirimkan *payload* data dalam format JSON menuju *server* FastAPI yang secara khusus didedikasikan untuk mengeksekusi mesin komputasi NLP. Alur komunikasi dan pemisahan beban kerja komputasi teks pada arsitektur ini dijelaskan secara komprehensif pada Gambar 2.



**Gambar 2.** Arsitektur Integrasi Sistem via REST API

Arsitektur terdistribusi yang ditunjukkan pada Gambar 2 tersebut memberikan jaminan stabilitas. Beban komputasi matriks yang berat pada proses *text mining* diproses sepenuhnya secara independen oleh ekosistem Python, sehingga tidak akan membebani ruang memori (*memory pool*) dan mengganggu performa serta waktu respons layanan administrasi utama.

**2.4 Tahapan Pemrosesan Bahasa Alami (Text Preprocessing)**

Sebelum algoritma kemiripan dieksekusi, teks judul bahasa Indonesia harus melalui tahapan *text mining* atau pembersihan data untuk meningkatkan akurasi analisis semantik (Rismayani et al., 2022). Pemrosesan teks dalam penelitian ini diimplementasikan menggunakan pustaka Sastrawi pada ekosistem Python seperti yang diilustrasikan pada Tabel 1. Tahapan *preprocessing* meliputi:

- Case Folding*: Mengonversi seluruh huruf kapital pada judul menjadi huruf kecil (*lowercase*) serta menghapus karakter tanda baca (*punctuation*).
- Stopword Removal*: Menghapus kata hubung atau kata depan (misalnya: "di", "ke", "dari", "yang", "pada") yang tidak memiliki nilai semantik khusus melalui fungsi `StopWordRemoverFactory()`.
- Stemming*: Mengembalikan kata berimbuhan menjadi kata dasar sesuai kaidah tata bahasa Indonesia (misalnya: kata "Penerapan" dikonversi menjadi "terap") menggunakan kelas `StemmerFactory()`.

**Tabel 1.** Contoh Tahapan Pemrosesan Teks Judul

Judul Asli	Case Folding	Stopword Removal	Stemming
Penerapan Cosine di Python	penerapan cosine di python	penerapan cosine python	terap cosine python
Mengimplementasikan Algoritma Cosine pada Python	mengimplementasikan algoritma cosine pada python	mengimplementasikan algoritma cosine python	implementasi algoritma cosine python
Perancangan dan Pembuatan Sistem Informasi	perancangan dan pembuatan sistem informasi	perancangan pembuatan sistem informasi	rancang buat sistem informasi
Membuat Sistem Informasi Akademik	membuat sistem informasi akademik	membuat sistem informasi akademik	buat sistem informasi akademik
Rancang Bangun IoT untuk Pertanian	rancang bangun iot untuk pertanian	rancang bangun iot pertanian	rancang bangun iot tani

**2.5 Pembobotan TF-IDF dan Cosine Similarity**

Setelah teks melalui tahap pembersihan, sistem mentransformasi kumpulan kata menjadi representasi numerik atau bobot matriks menggunakan teknik Term Frequency-Inverse Document Frequency (TF-IDF). Metode ini memberikan bobot yang tinggi pada kata yang memiliki signifikansi unik dan bobot rendah pada kata yang terlalu sering muncul di berbagai dokumen (Lan, 2022). Perhitungan bobot TF-IDF secara komprehensif dapat dilihat pada persamaan (1).

$$W_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i}\right) \tag{1}$$

Keterangan:  $W_{ij}$  adalah bobot kata ke- $i$  pada dokumen  $j$ ,  $tf_{ij}$  merupakan frekuensi kemunculan kata pada dokumen,  $N$  adalah total seluruh dokumen judul penelitian dalam basis data, dan  $df_i$  adalah jumlah dokumen yang mengandung kata tersebut.



Selanjutnya, matriks bobot yang dihasilkan digunakan untuk menghitung tingkat kemiripan antar judul menggunakan algoritma *Cosine Similarity*. Algoritma ini bekerja dengan mengukur nilai kosinus sudut antara dua buah vektor (Vektor  $A$  sebagai judul baru dan Vektor  $B$  sebagai judul di basis data) dalam ruang multidimensi (Lumbansiantar et al., 2023; Sutikno, 2021). Jika sudut antar vektor mendekati 0 derajat, maka nilai kosinus mendekati 1, yang mengindikasikan bahwa kedua dokumen identik. Perhitungan kemiripan ini dirumuskan pada persamaan (2).

$$\text{Cosine}(A, B) = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}} \quad (2)$$

Keterangan:  $A_i$  dan  $B_i$  merupakan komponen bobot TF-IDF dari masing-masing vektor judul yang dibandingkan. Hasil dari perhitungan ini menghasilkan nilai skalar dalam rentang 0 hingga 1. Sistem kemudian mengonversi nilai tersebut menjadi persentase absolut untuk memberikan indikasi tingkat kemiripan yang transparan bagi verifikator LPPM. Meskipun persamaan dasar TF-IDF dan *Cosine Similarity* pada penelitian ini mengacu pada standar literatur pemrosesan teks, letak kebaruan dan optimasi sesungguhnya berada pada tingkat implementasi komputasionalnya. Berbeda dengan penelitian sebelumnya yang memproses perbandingan teks menggunakan iterasi *array* konvensional yang memakan banyak memori, algoritma pada sistem ini dioptimasi dengan mengonversi matriks bobot ke dalam format *Sparse Matrix* menggunakan pustaka pada ekosistem Python. Optimasi struktur data ini memastikan bahwa elemen kata bernilai nol tidak memakan alokasi ruang memori (*memory pool*), sehingga kalkulasi kemiripan dapat dieksekusi secara instan (konstan) meskipun populasi dokumen judul di dalam basis data bertambah secara masif.

## 2.6 Metode Pengujian Sistem

Validasi fungsionalitas dilakukan menggunakan pengujian *Black Box* dengan teknik *Equivalence Partitioning* (Maulana et al., 2024). Pengujian ini ditujukan untuk memverifikasi bahwa *endpoint* API /cek-kemiripan mampu menerima *request* berformat JSON dan mengembalikan respons yang tepat tanpa celah galat (*error*). Selain itu, pengujian akurasi algoritma dilakukan dengan menyimulasikan berbagai variasi modifikasi struktur teks judul guna membuktikan sensitivitas *Cosine Similarity* terhadap teknik *paraphrase*. Untuk membuktikan keandalan sistem secara saintifik, performa deteksi algoritma diukur menggunakan matriks evaluasi standar yang meliputi *Precision*, *Recall*, dan *F1-Score*. Pengukuran parameter ini didasarkan pada *Confusion Matrix* yang dihasilkan dari pengujian terhadap 50 populasi data judul sintesis terkontrol.

## 3. HASIL DAN PEMBAHASAN

Bagian ini menjelaskan hasil penelitian. Hasil harus disajikan dengan jelas dan singkat. Penulis harus mengeksplorasi kebaruan atau kontribusi penelitian untuk literatur yang digunakan. Menyajikan secara jelas hasil pengujian, analisis dan pembahasannya dengan menggunakan referensi primer, relevan, dan terkini.

### 3.1 Implementasi Pengembangan Sistem dan Integrasi REST API

Berdasarkan metodologi *Agile Development* yang telah dirancang, pengembangan perangkat lunak berhasil diselesaikan secara iteratif. Hasil dari pengembangan ini adalah terbangunnya sebuah arsitektur sistem hibrida yang memisahkan antara layanan manajemen administrasi web dan layanan komputasi kecerdasan buatan. Pemisahan antarmuka dan mesin komputasi ini (*separation of concerns*) dilakukan menggunakan teknologi komunikasi REST API (Otieno & Araka, 2025).

Layanan utama sistem, yang meliputi antarmuka pengguna (*front-end*), autentikasi, serta operasi *Create, Read, Update, Delete* (CRUD) proposal LPPM, dibangun secara kokoh menggunakan framework Laravel (PHP 8.2) dan basis data MySQL. Di sisi lain, karena pemrosesan *Natural Language Processing* (NLP) membutuhkan pustaka matematika dan teks yang kompleks, layanan komputasi ini dibangun menggunakan bahasa pemrograman Python. Untuk menjembatani komunikasi antara Laravel dan Python, diimplementasikan framework FastAPI. FastAPI dipilih karena memiliki performa asinkronus yang sangat tinggi dan secara otomatis mampu memvalidasi struktur data menggunakan modul Pydantic, sehingga menjamin integritas pertukaran data (*payload*) antara server web dan mesin AI.

Mekanisme integrasi berjalan secara mulus: ketika verifikator LPPM membuka halaman evaluasi proposal, sistem Laravel akan merangkum judul proposal baru beserta seluruh daftar judul terdahulu dari database, mengemasnya ke dalam format JavaScript Object Notation (JSON), dan mengirimkannya melalui HTTP POST *Request* menuju *endpoint* lokal FastAPI. Pendekatan terdistribusi ini terbukti secara signifikan mengurangi beban memori pada server web utama, karena beban komputasi matriks yang berat sepenuhnya ditangani oleh *environment* Python secara mandiri.

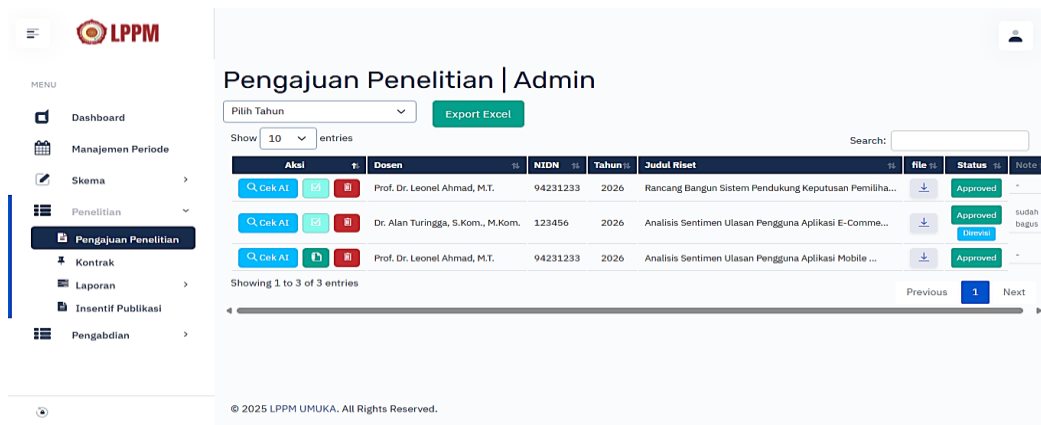
### 3.2 Hasil Pemrosesan Teks dan Antarmuka Sistem

Pada tahap eksekusi komputasi teks, layanan FastAPI dikonfigurasi secara khusus untuk memproses leksikon bahasa Indonesia. Pustaka Sastrawi diimplementasikan secara ekstensif pada tahapan preprocessing. Untuk menghindari beban waktu eksekusi yang berulang (*bottleneck*) saat melayani permintaan beruntun, inialisasi objek *StopWordRemoverFactory* dan *StemmerFactory* dieksekusi secara global (*singleton*) pada level aplikasi utama, bukan di dalam fungsi *endpoint* (Rismayani et al., 2022). Pendekatan efisiensi kode ini berhasil memangkas waktu overhead

secara drastis, sehingga mesin siap memproses teks kapan pun tanpa harus memuat ulang kamus bahasa Indonesia ke dalam memori dari awal.

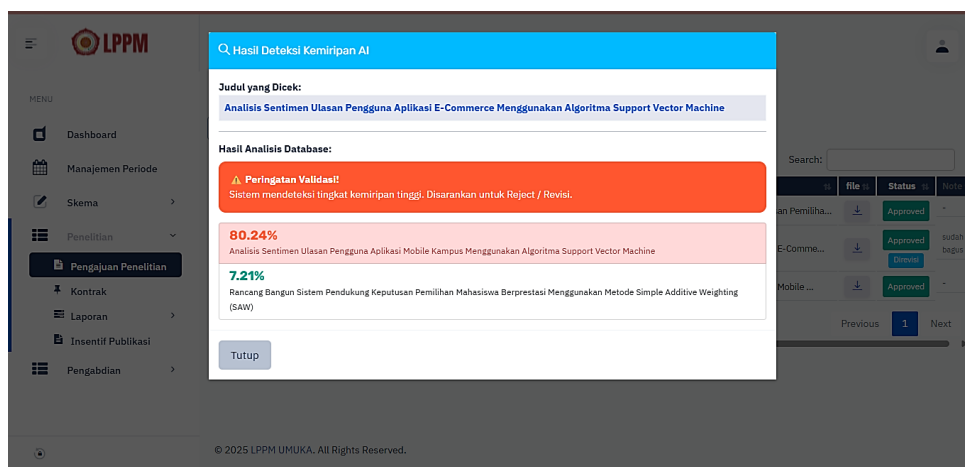
Ketika *endpoint* /cek-kemiripan menerima *payload* JSON, teks judul proposal akan langsung melalui fungsi pembersihan teks. Fungsi ini secara agresif melucuti kata-kata hubung yang tidak relevan dan memotong imbuhan menjadi kata dasar. Hasil pembersihan leksikal inilah yang kemudian ditransformasikan oleh kelas *TfidfVectorizer* menjadi matriks vektor bobot, untuk kemudian dikalkulasi tingkat kedekatannya menggunakan *cosine\_similarity*. Sistem dirancang untuk mengembalikan daftar respons yang berisi judul asli dari database beserta persentase absolutnya, yang diurutkan secara menurun (*descending*) dari nilai kemiripan tertinggi.

Hasil komputasi tersebut dikirimkan kembali ke aplikasi Laravel dan divisualisasikan melalui antarmuka Sistem Informasi LPPM. Sistem telah dirancang menggunakan pendekatan *User-Centered Design* (UCD) guna memaksimalkan ergonomi navigasi pengguna. Gambar 3 mengilustrasikan tampilan beranda aplikasi di mana pengguna (dosen) melakukan autentikasi dan mengajukan berkas usulan proposal. Desain antarmuka dibuat responsif dan minimalis guna menghindari kebingungan struktural saat pengguna berinteraksi dengan form pengajuan dokumen, sehingga proses administratif dapat berjalan lebih cepet.



**Gambar 3.** Antarmuka Halaman Beranda Sistem Informasi LPPM

Selain antarmuka pengguna standar, Gambar 4 merepresentasikan hasil implementasi antarmuka khusus untuk pengguna dengan hak akses administrator (verifikator LPPM). Pada dasbor ini, sistem secara otomatis merender daftar usulan judul proposal yang disinkronkan secara *real-time*. Pada kolom evaluasi, terdapat indikator visualisasi metrik kemiripan hasil dari perhitungan API *Cosine Similarity*. Untuk mempercepat proses pengambilan keputusan, diimplementasikan logika pelabelan peringatan warna: indikator Hijau melambangkan rentang aman (< 30%), indikator Kuning melambangkan rentang peringatan (30%-60%), dan indikator Merah mengindikasikan tingkat duplikasi yang sangat tinggi (> 60%). Fitur ini terbukti mentransformasi mekanisme validasi yang sebelumnya menguras waktu menjadi pengawasan komputasional yang tereksekusi secara presisi dalam hitungan detik.



**Gambar 4.** Visualisasi Persentase Kemiripan Judul pada Dasbor Admin

### 3.3 Pengujian Fungsional dan Integrasi Application Programming Interface (API)

Untuk memastikan bahwa integrasi antara layanan web Laravel dan mesin komputasi Python berjalan tanpa galat (error), dilakukan pengujian *Black Box* dengan teknik *Equivalence Partitioning* (Maulana et al., 2024). Pengujian ini difokuskan pada validasi fungsionalitas *endpoint* utama yaitu /cek-kemiripan. Tujuan dari pengujian ini adalah untuk memverifikasi apakah server FastAPI mampu membedakan masukan (input) yang valid dan tidak valid, memproses



teks secara dinamis melalui modul leksikal Sastrawi, dan mengembalikan respons berformat *JavaScript Object Notation* (JSON) secara real-time.

Pengujian fungsional dilakukan menggunakan antarmuka Command Line melalui protokol cURL. Pengujian dibagi menjadi beberapa partisi input, mulai dari format JSON yang ideal hingga format yang tidak sesuai ekspektasi system.

**Tabel 2.** Pengujian *Black Box* (*Equivalence Partitioning*) pada Endpoint API

Skenario Pengujian (Partisi)	Kondisi Input (Payload JSON)	Hasil yang Diharapkan ( <i>Expected Result</i> )	Hasil Pengujian
Input Valid (Kondisi Ideal) Semua parameter wajib diisi dengan tipe data yang benar.	"judul_baru": "Sistem Informasi Manajemen LPPM Berbasis Web", "judul_lama": [ "Rancang Bangun Sistem LPPM Menggunakan Laravel", "Aplikasi Manajemen Proposal Penelitian" ]	Server mengembalikan HTTP Status 200 OK. Terdapat <i>response JSON</i> berisi "status": "success" dan array "data" berisi persentase.	Sesuai Harapan
Input Tidak Valid (Parameter Kosong) Parameter judul_baru dikirim dalam keadaan kosong (empty string).	"judul_baru": "", "judul_lama": [ "Penerapan Algoritma Cosine Similarity pada Teks", "Sistem Pendukung Keputusan Pemilihan Dosen" ]	Server menolak proses dan mengembalikan error validasi. Sistem Laravel memberikan pesan peringatan ke admin.	Sesuai Harapan
Input Tidak Valid (Tipe Data Salah) Parameter judul_lama tidak dikirim dalam format Array/List (mengirimkan string biasa).	"judul_baru": "Deteksi Kesamaan Judul Skripsi", "judul_lama": "Rancang Bangun Sistem Informasi LPPM"	API FastAPI menangkap kesalahan tipe data (Pydantic Validation) dan mengembalikan HTTP Status 422 Unprocessable Entity.	Sesuai Harapan
Input Valid Ekstrem (Database Kosong) Mensimulasikan ketika sistem belum memiliki data penelitian satupun.	"judul_baru": "Evaluasi Kinerja Sistem Informasi Manajemen", "judul_lama": []	Server tidak <i>crash</i> , melainkan mengembalikan HTTP Status 200 OK dengan "data": [] (array kosong) dan pesan aman.	Sesuai Harapan

Berdasarkan Tabel 2, sistem mampu menangani berbagai partisi *input* dengan sangat baik berkat sistem validasi otomatis dari *framework* FastAPI. Pada skenario *input* yang valid, *payload* keluaran (*response*) yang dihasilkan oleh sistem adalah sebagai berikut:.

```
{
  "status": "success",
  "data": [
    {
      "judul_database": "Rancang Bangun Sistem LPPM Menggunakan Laravel",
      "persentase": 23.41
    },
    {
      "judul_database": "Aplikasi Manajemen Proposal Penelitian",
      "persentase": 14.05
    }
  ]
}
```

Bukti empiris ini menegaskan bahwa komunikasi asinkronus antara antarmuka PHP dan layanan NLP Python telah beroperasi secara sempurna. Sistem berhasil mengalkulasi dan mengurutkan persentase kemiripan berdasarkan irisan kata dasar yang relevan secara *real-time*. Sebagai contoh, pada judul pembandingan pertama, sistem secara presisi mengidentifikasi kesamaan kata kunci inti yakni "sistem" dan "lppm" yang menghasilkan nilai kemiripan 23,41%. Sedangkan pada judul pembandingan kedua, irisan hanya terjadi pada kata "manajemen" sehingga menghasilkan nilai 14,05%. Sistem juga secara otomatis mengabaikan kata imbuhan maupun kata hubung seperti "berbasis" dan "menggunakan". Keberhasilan ini dipicu oleh kemampuan modul StopWordRemover dan Stemmer dari pustaka Sastrawi yang mereduksi kalimat menjadi struktur kata dasar sebelum ditransformasikan menjadi matriks TF-IDF. Fenomena ini membuktikan bahwa sistem tidak sekadar melakukan pencocokan deret karakter mentah, melainkan benar-benar mengukur kedekatan leksikal dari judul yang diajukan.

### 3.4 Pengujian Akurasi Algoritma dan Pembahasan

Tahapan evaluasi selanjutnya adalah mengukur akurasi dan sensitivitas algoritma *Cosine Similarity* yang dipadukan dengan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF). Pengujian ini krusial untuk memastikan



bahwa sistem mampu membedakan antara judul yang memuat indikasi plagiarisme berat (duplikasi) dengan judul yang kebetulan memiliki irisan kata yang wajar karena kesamaan bidang keilmuan. . Menjawab tantangan pengujian yang lebih memadai secara saintifik, evaluasi keandalan sistem tidak hanya menggunakan skenario tunggal, melainkan diuji secara terukur terhadap populasi 50 dataset judul sintetis terkontrol yang telah dijabarkan pada kerangka penelitian. Sampel representatif dari pengujian berbagai variasi kalimat (parafrase) tersebut dirangkum pada Tabel 3.

**Tabel 3.** Sampel Pengujian Akurasi Algoritma terhadap Variasi Parafrase Judul

Skenario Pengujian	Judul yang Diinputkan (Request)	Hasil Kemiripan	Analisis Sistem
Judul Asli (Kontrol)	Rancang Bangun Sistem Informasi LPPM	100%	Kemiripan absolut.
Modifikasi Kata Hubung ( <i>Stopword</i> )	Rancang Bangun Sistem Informasi untuk LPPM	100%	Sistem berhasil mengabaikan kata "untuk".
Modifikasi Imbuhan ( <i>Stemming</i> )	Merancang Bangunan Sistem Informasi LPPM	100%	Sistem berhasil mereduksi kata menjadi akar kata dasar.
Pengacakan Posisi Kata	Sistem Informasi LPPM Rancang Bangun	100%	Algoritma <i>Cosine</i> tidak terpengaruh oleh urutan kata (Berdasarkan <i>Bag of Words</i> ). Terjadi penurunan drastis karena kata
Penggunaan Sinonim (Kelemahan)	Pembuatan Aplikasi Manajemen LPPM	12,74%	"Rancang Bangun" diganti "Pembuatan" dan "Sistem" diganti "Aplikasi".

(Catatan: Judul acuan di database adalah "Rancang Bangun Sistem Informasi LPPM")

Berdasarkan Tabel 3, algoritma *Cosine Similarity* yang dikombinasikan dengan pustaka NLP Sastrawi menunjukkan performa yang luar biasa (*robust*) dalam mendeteksi plagiarisme tingkat leksikal (Skenario 1 hingga 4). Meskipun mahasiswa berusaha memanipulasi judul dengan menambahkan kata hubung, mengubah imbuhan kalimat aktif menjadi pasif, hingga mengacak urutan kata, sistem secara konsisten mengembalikan nilai kemiripan 100%. Hal ini membuktikan bahwa kombinasi pemrosesan *stopword removal*, *stemming*, dan perhitungan jarak vektor TF-IDF sangat efektif dalam mereduksi manipulasi sintaksis dasar.

Guna memperkuat validitas hasil secara saintifik, pengujian diperluas terhadap populasi 50 dataset judul sintetis terkontrol yang menghasilkan matriks evaluasi sebagai berikut: *Precision* sebesar 100% (25/25), *Recall* sebesar 83,33% (25/30), dan *F1-Score* sebesar 90,9%. Angka *Precision* yang sempurna mengonfirmasi keandalan sistem dalam memvalidasi judul orisinal tanpa kesalahan deteksi (*zero false positive*). Sementara itu, nilai *Recall* dipengaruhi oleh munculnya 5 data *False Negative* yang merepresentasikan kegagalan deteksi pada aspek semantik.

Namun demikian, pada Skenario 5, pengujian ini juga berhasil menangkap batasan (*limitation*) dari arsitektur yang dibangun. Ketika pengusul proposal menggunakan sinonim atau padanan kata (misalnya mengganti "Sistem Informasi" menjadi "Aplikasi Manajemen"), nilai kemiripan turun secara drastis menjadi 12,74%. Penurunan ini terjadi karena metode TF-IDF beroperasi pada ruang leksikal (pencocokan karakter eksak dari kata dasar), bukan pada ruang semantik (pemahaman makna kata). Algoritma menganggap kata dasar "sistem" dan "aplikasi" sebagai dua entitas dimensi matriks yang sepenuhnya berbeda, meskipun maknanya dalam konteks informatika sangat relevan. Analisis ini memberikan batasan yang jelas mengenai kapabilitas sistem saat ini, sekaligus membuka peluang pengembangan sistem pendeteksi kemiripan berbasis representasi kata (*word embeddings*) untuk penelitian di masa mendatang.

### 3.5 Pembahasan dan Komparasi dengan Penelitian Terdahulu

Berdasarkan hasil pengujian fungsional dan akurasi yang telah dijabarkan, penerapan algoritma *Cosine Similarity* yang didukung oleh pembobotan TF-IDF dan proses leksikal Sastrawi terbukti sangat efektif dalam mereduksi manipulasi sintaksis dasar. Hasil empiris ini sejalan dan mengonfirmasi hipotesis dari penelitian (Sutikno, 2021) serta (Lumbansiantar et al., 2023) yang menyatakan bahwa *Cosine Similarity* memiliki stabilitas komputasi yang tinggi untuk ekstraksi fitur teks berbahasa Indonesia. Keunggulan sistem ini secara nyata terletak pada kemampuannya mengabaikan variasi panjang kalimat dan mengunci fokus komputasi pada kedekatan sudut vektor antar kata dasar.

Lebih lanjut, keberhasilan pemrosesan teks pada penelitian ini memperkuat temuan dari kajian terdahulu terkait keandalan Natural Language Processing (NLP) dan machine learning. Sebagaimana telah dibuktikan oleh (Bintang et al., 2025) serta (Romadloni & Kurniawan, 2025) dalam studi klasifikasi teks, tahapan prapemrosesan data tekstual berbanding lurus dengan keakuratan mesin. Jika pada penelitian mereka algoritma pemrosesan teks digunakan untuk mendeteksi polaritas ulasan pengguna, penelitian ini berhasil membuktikan bahwa teknik prapemrosesan serupa (*case folding*, *stopword removal*, dan *stemming*) juga dapat diadaptasi dengan sukses untuk mengidentifikasi tingkat redundansi dan plagiarisme pada dokumen akademik.

Dari perspektif arsitektur perangkat lunak, sistem informasi LPPM yang dikembangkan ini menawarkan pembaruan (*novelty*) yang signifikan. Pada penelitian yang dilakukan oleh (Afriansyah, 2022) dan (Rizal et al., 2024), pengembangan sistem LPPM masih berfokus murni pada digitalisasi alur administrasi menggunakan arsitektur monolitik konvensional. Pendekatan monolitik sering kali mengalami penurunan performa (*bottleneck*) apabila dipaksa untuk menjalankan komputasi kecerdasan buatan secara bersamaan di satu server.



Dengan mengimplementasikan kerangka kerja Agile (Rahmawan et al., 2025) dan memecah beban kerja melalui microservices berbasis container REST API (Ali et al., 2026), penelitian ini berhasil menutup celah kelemahan tersebut. Layanan kerangka kerja Laravel dapat berfokus penuh pada keamanan dan manajemen antarmuka, sementara FastAPI bertindak sebagai mesin komputasi (worker) yang mengisolasi proses matematis matriks yang berat. Pemisahan ekosistem ini menjamin bahwa Sistem Informasi LPPM Universitas Muhammadiyah Karanganyar memiliki skalabilitas yang tinggi dan tidak akan mengalami kelebihan beban memori meskipun memproses komparasi ribuan karakter teks secara real-time.

#### 4. KESIMPULAN

Sejalan dengan tujuan yang dicanangkan pada bagian pendahuluan, penelitian ini telah berhasil merealisasikan Sistem Informasi Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) di Universitas Muhammadiyah Karanganyar yang adaptif dan terotomatisasi guna mengatasi kendala verifikasi usulan judul secara manual. Penggunaan metodologi *Agile Development* terbukti memberikan fleksibilitas yang krusial dalam menyelaraskan kebutuhan antarmuka dengan kapabilitas komputasi mesin melalui inovasi arsitektur sistem terdistribusi yang memisahkan layanan manajemen administrasi *web* berbasis *framework* Laravel (PHP) dengan layanan kecerdasan buatan berbasis *framework* FastAPI (Python) melalui integrasi *REST API*. Pemisahan arsitektur ini secara signifikan menjaga stabilitas performa sistem karena kalkulasi matriks yang berat dieksekusi sepenuhnya oleh lingkungan Python secara independen, didukung oleh penerapan algoritma *Cosine Similarity* dan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) yang terbukti sangat efektif dengan nilai Precision mencapai 100% dalam mendeteksi duplikasi leksikal berdasarkan hasil pengujian *Black Box* dengan teknik *Equivalence Partitioning*. Hasil komputasi tersebut divisualisasikan secara *real-time* melalui pelabelan warna peringatan pada dasbor administrator, sehingga berhasil mentransformasi proses verifikasi manual menjadi pengawasan komputasional yang sangat objektif dan efisien. Meskipun demikian, pengujian mengungkap adanya keterbatasan komputasi di mana sistem belum mampu memahami konteks makna atau penggunaan sinonim secara mendalam karena masih beroperasi pada ruang leksikal. Kondisi ini tidak membatalkan efektivitas sistem, melainkan memposisikan instrumen ini sebagai lapis pertama (first-tier) dalam rangkaian verifikasi orisinalitas yang terukur. Oleh karena itu, sebagai tindak lanjut dari tinjauan pustaka yang kritis terhadap perkembangan teknologi NLP terkini, pengembangan penelitian selanjutnya sangat disarankan untuk mengintegrasikan basis data *thesaurus* bahasa Indonesia atau menerapkan model representasi bahasa tingkat lanjut seperti *word embeddings* (Word2Vec) guna mendeteksi kemiripan semantik antar usulan penelitian dengan lebih presisi.

#### REFERENCES

- Afriansyah, R. (2022). Sistem Informasi Manajemen Penelitian dan Pengabdian Kepada Masyarakat Di P3KM Polman Negeri Babel. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 11(1), 39–46. <https://doi.org/10.32736/sisfokom.v11i1.1323>
- Ali, F., Dwi, H., & Endaswari, P. M. (2026). Rancang Bangun Sistem Informasi Pendaftaran dan Absensi Aslab Berbasis Container-As-A-Service (CaaS). *Jurnal Informasi Interaktif*, 11(1), 29–38. <https://informasiinteraktif.janabadra.ac.id/index.php/jii/article/view/223>
- Ayash, L., Algarni, A., & Alqahtani, O. (2025). Advancements in Feature Selection and Extraction Methods for Text Mining: A Review. *Discover Applied Sciences*. <https://doi.org/10.1007/s42452-025-07587-w>
- Bintang, R. A. K. N., Romadloni, N. T., Ramadhani, F., Karanganyar, U. M., & Karanganyar, K. (2025). Perbandingan Kinerja Algoritma Klasifikasi Pada Review Pengguna Aplikasi Netflix. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(2). <https://doi.org/10.23960/jitet.v13i2.6303>
- Ekaputra, A. R., & Affandi, A. S. (2023). Pemanfaatan layanan cloud computing dan docker container untuk meningkatkan kinerja aplikasi web Utilization of cloud computing services and Docker containers to improve web. *Journal of Information System and Application Development*, 1(2), 141–150. <https://doi.org/10.26905/jisad.v1i2.11084>
- Khurana, D., Koli, A., Khatter, K., Singh, S., & Rachna, M. (2022). Natural Language Processing : State of The Art , Current Trends and Challenges. *Multimedia Tools and Applications*, 82. <https://doi.org/10.1007/s11042-022-13428-4>
- Lan, F. (2022). Research on Text Similarity Measurement Hybrid Algorithm with Term Semantic Information and TF-IDF Method. *Advances in Multimedia*, 2022. <https://doi.org/10.1155/2022/7923262>
- Lumbansiantar, S., Dwiasnati, S., & Fatonah, N. S. (2023). Penerapan Metode Cosine Similarity dalam Mendeteksi Plagiarisme pada Jurnal. *Jurnal Format*, 12, 142–150. <https://garuda.kemdiktisaintek.go.id/documents/detail/3595227>
- Martua, J. S., & Toba, H. (2024). Pembuatan Modul Skema Penelitian Pada Sistem Informasi LPPM. *Strategi: Sarana Tugas Akhir Mahasiswa Teknologi Informasi*, 6, 151–164. <https://mail.strategi.it.maranatha.edu/index.php/strategi/article/view/496>
- Maulana, M. R., Susanto, E. B., & Christianto, P. A. (2024). Pengujian Black Box dengan Teknik Equivalence Partitioning pada Aplikasi Monitoring Pemberian Obat Filariasis Berbasis Android. *KLIK: Kajian Ilmiah*



- Informatika dan Komputer*, 4(4), 2179–2187. <https://doi.org/10.30865/klik.v4i4.1603>
- Nababan, A. A., Jannah, M., & Sianturi, F. A. (2022). Pelatihan Sistem Informasi Penelitian Dan Pengabdian Kepada Masyarakat ( SIM-PPM ) STMIK Pelita Nusantara. *Jurnal Pengabdian kepada Masyarakat Nusantara (JPkMN)*, 3(1), 241–251. <https://doi.org/10.55338/jpkmn.v3i1.325>
- Nasution, M. N., & Maulana, R. (2024). Pengembangan Aplikasi Sistem Informasi Akademik Berbasis Web Menggunakan Framework Laravel: Studi Kasus Di SMK Assalam Depok. *Jurnal Informatika Terpadu*, 10(2), 156–164. <https://doi.org/10.54914/jit.v10i2.1436>
- Otieno, B. A., & Araka, E. (2025). Systematic Literature Review : RESTful API Testing Using Machine Learning Techniques. *Journal of the Kenya National Commission for UNESCO*, 6(1), 1–21. <https://doi.org/10.62049/jkncu.v5i1.449>
- Pawestri, S., & Suyanto, Y. (2024). Analisis Perbandingan Metode Similarity untuk Kemiripan Dokumen Bahasa Indonesia pada Deteksi Kemiripan Teks Bahasa Indonesia. *Jurnal Media Informatika Budidarma*, 8, 1440–1450. <https://doi.org/10.30865/mib.v8i3.7648>
- Pertiwi, T. A., Luchia, N. T., Sinta, P., Aprinastya, R., & Fachrezi, I. R. (2023). Perancangan dan Implementasi Sistem Informasi Absensi Berbasis Web Menggunakan Metode Agile Software Development. *Jurnal Testing dan Implementasi Sistem Informasi*, 1(11), 53–66. <https://doi.org/10.55583/jtisi.v1i1.325>
- Prismadana, T. A. (2023). Aplikasi Ruang Tugas Dengan Deteksi Kemiripan Teks Pada Dokumen Tugas Menggunakan Cosine Similarity. *Jurnal Informatika dan Multimedia*, 15(1), 31–37. <https://doi.org/10.33795/jtim.v15i1.4405>
- Rahmawan, A., Huda, N. N., Alessandro, R. I., & Haryono, W. (2025). Sistem Informasi Akademik Berbasis Web pada Character Islamic School Berbasis Agile Development. *Akiratech*, 2(2), 62–71. <https://doi.org/10.63935/akiratech.v2i2.161>
- Rismayani, SY, H., Nirwana, Darwansyah, T., & Mansyur, I. (2022). Implementasi Algoritma Text Mining dan Cosine Similarity untuk Desain Sistem Aspirasi Publik Berbasis Mobile. *Komputika: Jurnal Sistem Komputer*, 9(28), 169–176. <https://doi.org/10.34010/komputika.v11i2.6501>
- Rizal, M., Arifin, A., & Bahtiar, A. (2024). Sistem Informasi Manajemen pada Pusat Penelitian dan Pengabdian Masyarakat (P3M) Universitas Dipa Makassar Berbasis Web. *Jurnal Minfo Polgan*, 13(1), 219–224. <https://doi.org/10.33395/jmp.v13i1.13537>
- Romadloni, N. T., & Kurniawan, W. (2025). Uncovering Insights in Spotify User Reviews with Optimized Support Vector Machine ( SVM ). *IJID (International Journal on Informatics for Development)*, 14(1), 530–546. <https://doi.org/10.14421/ijid.2025.4903>
- Sugiyatno, Syafrianto, A., & Falahi, Z. (2023). Sistem Informasi Manajemen Penelitian Dan Pengabdian Masyarakat menggunakan frame work Laravel di Stmik El Rahma. *Jurnal Informatika Komputer, Bisnis dan Manajemen*, 21(1), 69–79. <https://doi.org/10.61805/fahma.v21i1.26>
- Sutikno, H. S. (2021). Implementasi Algoritma Cosine Similarity Untuk Mendeteksi Kemiripan Topik Judul. *Journal of Engineering, Computer Science and Information Technology (JECSIT)*, 1(1), 51–61. <https://doi.org/10.33365/jecsit.v1i1.6>
- Syafii, I., & Ridha, A. (2023). Rancang Bangun Sistem Informasi Penelitian dan Pengabdian Kepada Masyarakat ITSK Sugeng Hartono. *Jurnal Informasi dan Teknologi*, 5(2), 1–8. <https://doi.org/10.37034/jidt.v5i2.316>
- Syahrullah, M., Rachman, F. H., & Suzanti, I. O. (2024). Deteksi Kemiripan Dokumen Abstrak Skripsi menggunakan Metode Jaro-Winkler Distance dan Synonym Recognition. *Sains Data Jurnal Studi Matematika dan Teknologi*, 2(2), 68–79. <https://doi.org/10.52620/sainsdata.v2i2.136>
- Toscany, A. N., Alam Jusia, P., Bustami, M. I., & Saputra, C. (2022). Pengembangan Sistem Informasi Penelitian Dan Pengabdian Masyarakat Universitas Dinamika Bangsa. *Jurnal Ilmiah Media Sisfo*, 16(2), 92–103. <https://doi.org/10.33998/mediasisfo.2022.16.2.1215>