

PERANCANGAN APLIKASI DETEKSI PLAGIARISME DENGAN TF-IDF DAN *COSINE SIMILARITY*

Shandy Arkan¹, Heru Sulistiono², Irawan Setiadi³

Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer
Universitas Indraprasta PGRI^{1,2,3}

Jalan Raya Tengah No 80, Kelurahan Gedong, Pasar Rebo, Jakarta Timur
dalamshandy.sa@gmail.com¹, mildlaser3@gmail.com², irawan.setiadi91@gmail.com³

Abstrak

Plagiarisme merupakan tantangan serius dalam dunia akademik yang mengancam integritas ilmiah. Akses terhadap alat deteksi plagiarisme komersial seperti Turnitin sering terbatas karena biaya tinggi, sementara alat gratis memiliki batasan jumlah kata dan masalah privasi data. Penelitian ini bertujuan mengembangkan aplikasi deteksi plagiarisme *offline* berbasis *Natural Language Processing* (NLP) dengan algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity*. Aplikasi ini dirancang untuk beroperasi tanpa koneksi internet, tanpa batasan *word count*, dan menjamin keamanan data. Metode pengembangan menggunakan Python 3.13 dengan *framework* PySide6 untuk antarmuka pengguna dan SQLite sebagai database. Hasil pengujian terhadap 1 dokumen uji dan 9 dokumen pembanding menunjukkan akurasi deteksi dengan *overall score* 9,38% (metode *SkLearn*). Waktu pemrosesan masing-masing adalah 8,55 detik. Aplikasi ini diharapkan dapat menjadi solusi alternatif bagi mahasiswa dan institusi pendidikan dalam mendeteksi plagiarisme secara mandiri dan aman.

Kata Kunci : Deteksi Plagiarisme, *Natural Language Processing*, TF-IDF, *Cosine Similarity*, Python

Abstract

Plagiarism is a serious challenge in the academic world that threatens scientific integrity. Access to commercial plagiarism detection tools like Turnitin is often limited due to high costs, while free tools have word count limitations and data privacy issues. This research aims to develop an offline plagiarism detection application based on natural language processing using the term frequency-inverse document frequency (TF-IDF) and cosine similarity algorithms. This application is designed to operate without an internet connection and without word count limitations and guarantees data security. The development method uses Python 3.13 with the PySide6 framework for the user interface and SQLite as the database. The test results on 1 test document and 9 comparison documents show a detection accuracy with an overall score of 9.38% (SkLearn method). The processing time for each is 8.55 seconds. This application is expected to be an alternative solution for students and educational institutions in independently and safely detecting plagiarism.

Keywords: *Plagiarism Detection, Natural Language Processing, TF-IDF, Cosine Similarity, Python*

PENDAHULUAN

Integritas akademik merupakan fondasi utama dalam dunia pendidikan yang merepresentasikan kredibilitas dan kualitas ilmu pengetahuan (Gregory & Leeman, 2021). Namun, fenomena plagiarisme menjadi tantangan serius yang mengancam nilai-nilai kejujuran akademik. Plagiarisme didefinisikan sebagai tindakan mengambil bahasa, pemikiran, ide, atau ekspresi dari penulis lain dan menganggapnya sebagai karya asli sendiri. Fenomena ini telah berkembang menjadi masalah internasional yang kompleks seiring kemudahan akses informasi digital (Curtis & Tremayne, 2021). Di Indonesia, penelitian (Karim & Zakiyah, 2023) menunjukkan bahwa 58,5% mahasiswa di UIN Syarif Hidayatullah Jakarta pernah melakukan tindakan plagiarisme, dengan faktor utama berupa tekanan akademik, keterbatasan akses alat deteksi premium, dan kurangnya pemahaman tata cara sitasi. *Tools* deteksi plagiarisme komersial seperti Turnitin memiliki keterbatasan akses karena biaya tinggi. Sementara itu, *tools* gratis memiliki batasan jumlah kata (1.000–5.000 kata) dan akurasi yang tidak konsisten (Foltýnek dkk., 2019). Selain itu, aspek privasi dan keamanan data menjadi kekhawatiran utama karena sistem deteksi *online* sering mengharuskan pengunggahan

dokumen ke server eksternal dengan metode perlindungan data yang tidak transparan (Ihle dkk., 2020). Berdasarkan permasalahan tersebut, penelitian ini bertujuan mengembangkan aplikasi deteksi plagiarisme *offline* berbasis *Natural Language Processing* (NLP) dengan algoritma Term Frequency-Inverse Document Frequency (TF-IDF) dan *Cosine Similarity*. Aplikasi ini dirancang untuk beroperasi tanpa koneksi internet dan tanpa batasan jumlah kata. Implementasi NLP dipilih karena efektivitasnya dalam mendeteksi kemiripan teks secara leksikal, sebagaimana ditunjukkan dalam penelitian Meidelfi dkk. (2021) yang berhasil menggabungkan TF-IDF dan *Cosine Similarity* untuk mendeteksi kemiripan dokumen dengan akurasi yang baik. Manfaat penelitian ini terdiri dari aspek teoritis dan praktis. Secara teoritis, penelitian dapat menjadi referensi pengembangan metode NLP berbasis TF-IDF dan *Cosine Similarity* untuk teks berbahasa Indonesia. Secara praktis, aplikasi ini diharapkan dapat digunakan oleh mahasiswa sebagai *tool* deteksi plagiarisme gratis dan mandiri, serta oleh institusi pendidikan sebagai alternatif solusi pencegahan plagiarisme.

PENELITIAN RELEVAN

Penelitian mengenai sistem deteksi plagiarisme menggunakan metode *Natural Language Processing* (NLP) dengan algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity* telah banyak dilakukan dalam beberapa tahun terakhir. Berikut ini beberapa penelitian terdahulu telah melakukan percobaan sehingga menyumbang kontribusi untuk pengembangan dan kemajuan sistem deteksi plagiarisme.

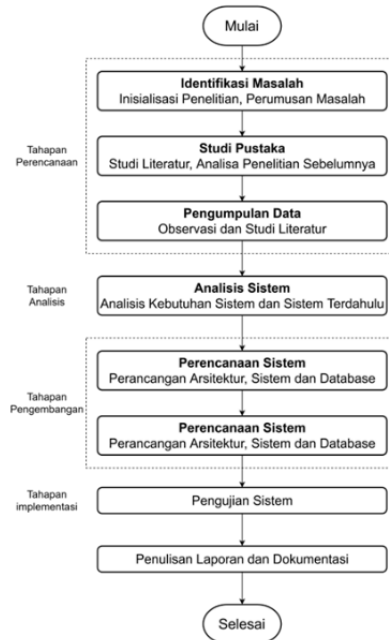
Tabel 1. Penelitian Relevan

No.	Peneliti (Tahun)	Judul Penelitian	Metode Penelitian	Hasil Penelitian	Analisis Relevansi
1	(Meidelfi dkk., 2021)	<i>TF-IDF Implementation for Similarity Checker on The Final Project Title</i>	<i>Preprocessing, TF-IDF, Cosine Similarity</i>	Akurat dalam menghitung kemiripan judul proyek akhir.	Menunjukkan efektivitas dasar kombinasi TF-IDF & <i>Cosine Similarity</i> , menjadi fondasi penelitian ini.
2	(Riyani dkk., 2019)	Penerapan <i>Cosine Similarity</i> dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen	<i>Preprocessing</i> (termasuk <i>stemming</i> dengan Sastrawi), TF-IDF, <i>Cosine Similarity</i>	<i>Stemming</i> meningkatkan nilai kemiripan rata-rata 10%; efektif untuk abstrak skripsi.	Penelitian menggunakan bahasa Indonesia dan membuktikan pentingnya <i>stemming</i> .
3	(Wibowo dkk., 2022)	<i>Similarity Identification of Large-scale Biomedical Documents using Cosine Similarity and Parallel Computing</i>	<i>Preprocessing, TF-IDF dengan parallel computing, Cosine Similarity</i>	Efektif dan mampu digunakan untuk <i>dataset</i> besar, menunjukkan efisiensi komputasi paralel.	Menunjukkan kemampuan algoritma, memberikan optimasi performa jika dikembangkan untuk <i>dataset</i> besar.
4	(Halim & Lasut, 2024)	<i>Document Plagiarism Detection Application Using Web-Based TF-IDF and Cosine Similarity Methods</i>	<i>Preprocessing, TF-IDF, Cosine Similarity</i> , berbasis web	Algoritma tersebut akurat dalam mendeteksi kesamaan dokumen	Penelitian serupa memiliki perbedaan platform memberikan nilai pembeda.
5	(Singh dkk., 2023)	<i>Plagiarism Checker using TF-IDF, Cosine Similarity and Jaccard Similarity</i>	<i>Preprocessing, TF-IDF, Cosine Similarity, Jaccard Similarity</i>	Model berguna untuk deteksi plagiarisme	Menggunakan pendekatan <i>hybrid</i> , memberikan perspektif alternatif dalam pengukuran <i>similarity</i> .
6	(Foltýnek dkk., 2019)	<i>Academic Plagiarism Detection: A Systematic Literature Review</i>	<i>Systematic literature review</i> terhadap berbagai metode deteksi plagiarisme	Mengklasifikasikan metode deteksi berdasarkan lapisan bahasa (karakter, sintaks, semantik, ide).	Memberikan kerangka teoretis komprehensif yang menjadi pijakan penentuan level deteksi (leksikal) pada penelitian ini.

METODE PENELITIAN

Tahapan Penelitian

Dalam penelitian ini, terdapat lima tahapan yang meliputi perencanaan, analisis, pengembangan, implementasi dan simpulan penelitian. Berikut ini gambar *flowchart* dari tahapan penelitian :



Gambar 1. *Flowchart* Tahapan Penelitian

1. Perencanaan

a. Identifikasi Masalah

Pada tahapan awalan perencanaan dilakukan identifikasi masalah, bertujuan untuk menemukan masalah yang sedang terjadi. Terdapat masalahnya adalah keterbatasan akses terhadap alat deteksi dikarenakan berbayar, batasan jumlah kata pada alat gratis, masalah privasi data, inkonsistensi hasil dan kurang banyak tersedia sistem *offline*.

b. Studi Kepustakaan

Studi kepustakaan merupakan tahapan yang mencakup penelusuran literatur ilmiah seperti jurnal internasional maupun nasional. Fokus utama dalam studi utama ini adalah memahami teori dan konsep-konsep dasar terkait plagiarisme, *Natural Language Processing*, algoritma *Term Frequency-Inverse Document Frequency* dan *Cosine Similarity*.

c. Pengumpulan Data

Data primer dikumpulkan melalui observasi langsung terhadap kebutuhan pengguna dan analisis karakteristik terhadap karya ilmiah, lalu data sekunder dikumpulkan melalui studi literatur dan pengumpulan dokumen karya ilmiah untuk keperluan pengujian sistem nantinya.

2. Analisis

Pada tahapan analisis sistem, peneliti melakukan identifikasi terhadap kebutuhan yang diperlukan dalam pengembangan aplikasi sistem deteksi plagiarisme, sebelum lebih lanjut beranjak ke tahapan perancangan.

3. Pengembangan

Tahap pengembangan sistem merupakan implementasi dari desain dasar atau *blueprint* menjadi kode program. Pengembangan ini dilakukan menggunakan bahasa pemrograman python versi 3.13 dengan penggunaan *library* yang dibutuhkan sistem dan juga SQLite 3 untuk database

4. Implementasi

Dalam Tahapan ini dilakukan pengujian sistem yang bertujuan untuk melakukan pemeriksaan dan memastikan bahwa aplikasi telah memenuhi terhadap semua kebutuhan atau standar yang

telah ditetapkan. Pengujian ini juga meliputi tes fungsional aplikasi, performa aplikasi dan desain antarmuka yang mudah dimengerti dan digunakan.

5. Simpulan

Sistem ini mengimplementasikan algoritma *Term Frequency-Inverse Document Frequency* untuk ekstraksi fitur dan pembobotan kata, serta *Cosine Similarity* untuk mengukur kemiripan antar dokumen.

Algoritma

1. Text Processing

Text preprocessing adalah tahapan awal berfungsi mengubah *raw text* menjadi format yang terstruktur dan bisa dianalisis oleh sistem. Proses *text preprocessing* ini juga hal yang krusial dalam sistem deteksi plagiarisme dikarenakan dapat meningkatkan nilai akurasi dan efisiensi dalam proses deteksi. Meliputi *case folding*, *tokenizing*, *punctuation removal*, *stopword removal* dan *stemming*.

2. Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency adalah metode pembobotan statistik yang digunakan untuk mengevaluasi tingkat frekuensi kemunculan suatu kata pada dokumen relatif terhadap dokumen secara keseluruhan. Algoritma ini bekerja dengan langkah pertama hitung nilai TF, kedua nilai IDF dan terakhir melakukan pembobotan nilai TF dan IDF. Implementasi algoritma TF-IDF dalam aplikasi ini terdapat dua rumus yaitu rumus teoritis klasik dan implementasi praktis dalam pustaka python yaitu *scikit-learn*.

a. Rumus classic

Pertama dilakukan perhitungan TF sebagai berikut.

$$TF_{ij} = \frac{F_{ij}}{D}$$

Di mana setiap variabel dijelaskan sebagai berikut :

F_{ij} = jumlah kemunculan term j dalam dokumen i

D = total jumlah term dalam dokumen i

Kedua dilakukan perhitungan IDF dengan persamaan sebagai berikut.

$$IDF_{ij} = \log\left(\frac{N}{D_{fj}}\right)$$

Di mana setiap variabel dijelaskan sebagai berikut :

N = jumlah total dokumen dalam korpus

D_{fj} = jumlah dokumen yang mengandung term j

Terakhir dilakukan perhitungan pembobotan TF dan IDF dengan persamaan sebagai berikut.

$$W_{ij} = TF_{if} \times IDF_j$$

Bobot ini merepresentasikan tingkat kepentingan suatu term dalam dokumen tertentu relatif terhadap keseluruhan korpus.

b. Implementasi dalam *scikit-learn*

Perbedaan utama dengan rumus klasik terletak pada penambahan *smoothing* pada perhitungan IDF dan skema normalisasi vektor secara otomatis. *Scikit-learn* menambahkan konstanta *smoothing* pada IDF untuk menghindari pembagian dengan nol dan memberikan bobot yang lebih baik untuk istilah yang tidak muncul di beberapa dokumen. Rumusnya menjadi:

$$IDF_{ij} = \log\left(\frac{1 + N}{1 + D_{fj}}\right) + 1$$

3. Cosine Similarity

Cosine similarity merupakan suatu metode untuk pengukuran kesamaan yang digunakan untuk membandingkan dua buah vektor nol-nol dalam ruang produk (*inner product space*).

Implementasi algoritma *cosine similarity* dalam aplikasi ini terdapat dua rumus yaitu rumus teoritis klasik dan implementasi praktis dalam pustaka python yaitu *scikit-learn*.

a. Rumus klasik *cosine similarity*

$$Similarity = \cos(\theta) = \frac{(A \times B)}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{(\sum_{i=1}^n A_i^2)^{1/2} \times (\sum_{i=1}^n B_i^2)^{1/2}}$$

Di mana setiap variabel dijelaskan sebagai berikut :

A: Vektor A

B: Vektor B

$\|A\|$: Magnitudo (panjang) dari vektor A

$\|B\|$: Magnitudo (panjang) dari vektor B

b. Implementasi *cosine similarity* dalam *scikit-learn*

Ketika vektor input (seperti vektor TF-IDF yang telah dinormalisasi L2 oleh TF-IDF *Vectorizer* maka secara otomatis nilai *magnitude* menjadi 1, maka penyebut pada rumus menjadi $1 \times 1 = 1$. Dengan demikian, perhitungan *cosine similarity* direduksi menjadi hanya perkalian titik (*dot product*) antar vektor

HASIL DAN PEMBAHASAN

Pembahasan

Dokumen yang akan dipakai penelitian ini adalah 1 dokumen yang diuji dan 11 dokumen pembanding. Dokumen yang diuji diambil dari jurnal yang dipublikasikan oleh Universitas Negeri Jakarta yang terbit pada 05 Januari 2024 dengan judul "Pengaruh Penerapan Teknologi Digital Terhadap Kemandirian Siswa Kelas IV Pada Sekolah Penggerak Di Wilayah Kota Tangerang" dan tulis oleh Arry Patria Surya Azhar, Edwita dan Gusti Yarmi. Pada dokumen pembanding diambil dari referensi yang digunakan pada saat pengujian hanya berjumlah 9 dokumen.

1. *Text Processing*

a. *Case Folding*

Teks input akan dilakukan perubahan pada karakter yang menggunakan huruf kapital menjadi kecil.

Tabel 1. *Case Folding*

Sebelum	Sesudah
Abstrak : Artikel ini bertujuan untuk mendalami penerapan teknologi dalam konteks Program Sekolah Penggerak dengan fokus pada bagaimana teknologi dapat digunakan untuk mendorong kemandirian siswa di era digital.	abstrak : artikel ini bertujuan untuk mendalami penerapan teknologi dalam konteks program sekolah penggerak dengan fokus pada bagaimana teknologi dapat digunakan untuk mendorong kemandirian siswa di era digital.

b. *Punctuation Removal*

Pada tahapan ini akan dilakukan penghapusan tanda baca, simbol dan angka.

Tabel 2. *Punctuation Removal*

Sebelum	Sesudah
Abstrak : artikel ini bertujuan untuk mendalami penerapan teknologi dalam konteks program sekolah penggerak dengan fokus pada bagaimana teknologi dapat digunakan untuk mendorong kemandirian siswa di era digital.	abstrak artikel ini bertujuan untuk mendalami penerapan teknologi dalam konteks program sekolah penggerak dengan fokus pada bagaimana teknologi dapat digunakan untuk mendorong kemandirian siswa di era digital

c. *Tokenization*

Pada tahap ini kalimat akan di rubah bentuk menjadi sebuah bentuk token.

Tabel 3. *Tokenization*

Sebelum	Sesudah
Abstrak artikel ini bertujuan untuk mendalami penerapan teknologi dalam konteks program sekolah penggerak dengan	"abstrak", "artikel", "ini", "bertujuan", "untuk", "mendalami", "penerapan", "teknologi", "dalam", "konteks", "program", "sekolah", "penggerak", "dengan",

fokus pada bagaimana teknologi dapat digunakan untuk mendorong kemandirian siswa di era digital	"fokus", "pada", "bagaimana", "teknologi", "dapat", "digunakan", "untuk", "mendorong", "kemandirian", "siswa", "di", "era", "digital",
---	--

d. *Stopword Removal*

Pada tahap ini akan dilakukan penghapusan kata-kata yang memiliki nilai semantik rendah.

Tabel 4. *Stopword Removal*

Sebelum	Sesudah
"abstrak", "artikel", "ini", "bertujuan", "untuk", "mendalami", "penerapan", "teknologi", "dalam", "konteks", "program", "sekolah", "penggerak", "dengan", "fokus", "pada", "bagaimana", "teknologi", "dapat", "digunakan", "untuk", "mendorong", "kemandirian", "siswa", "di", "era", "digital",	"abstrak", "artikel", "bertujuan", "mendalami", "penerapan", "teknologi", "konteks", "program", "sekolah", "penggerak", "fokus", "teknologi", "mendorong", "kemandirian", "siswa", "era", "digital",

e. *Stemming*

Pada Tahap ini akan dilakukan pengubahan bentukan kata kembali menjadi ke bentuk dasarnya.

Tabel 5. *Stemming*

Sebelum	Sesudah
"abstrak", "artikel", "bertujuan", "mendalami", "penerapan", "teknologi", "konteks", "program", "sekolah", "penggerak", "fokus", "teknologi", "mendorong", "kemandirian", "siswa", "era", "digital",	"abstrak", "artikel", "tujuan", "dalam", "terap", "teknologi", "konteks", "program", "sekolah", "gerak", "fokus", "teknologi", "dorong", "mandiri", "siswa", "era", "digital",

2. Pembobotan TF-IDF

Setelah dilakukan tahap *text processing* data akan dilakukan pembobotan TF-IDF. Pada tahap ini dilakukan pembobotan statistik yang digunakan untuk mengevaluasi tingkat frekuensi kemunculan suatu kata pada dokumen relatif terhadap dokumen secara keseluruhan.

```

TF-IDF MATRIX SUMMARY
=====
Matrix Shape: 10 documents x 4723 terms
Total Elements: 47,230
Non-Zero Elements: 8,258
Zero Elements: 38,972
Sparsity: 82.52% (percentage of zero elements)

Per-Document Statistics:
-----
Doc# Document Name Non-Zero Max TF-IDF Avg TF-IDF
-----
0 Input Document 388 0.0078 0.0006
1 Proceeding of Biology Educa... 701 0.0120 0.0005
2 Implementasi Pembelajaran O... 507 0.0157 0.0006
3 KECANDUAN GAWAI SEBAGAI IMP... 606 0.0280 0.0005
4 Inovasi Pemanfaatan Teknolo... 795 0.0045 0.0004
5 Literasi Digital Peserta Di... 632 0.0200 0.0005
6 Kontribusi Orang Tua Dalam ... 706 0.0045 0.0004
7 MODEL KOMUNIKASI ORANG TUA ... 2564 0.0108 0.0002
8 Melek Digital_ Tantangan Gu... 729 0.0085 0.0004
9 Problematika Pembelajaran D... 630 0.0106 0.0005
    
```

Gambar 2. Hasil Perhitungan Pembobotan TF-IDF

3. Perhitungan Nilai *Cosine Similarity*

Berikut hasil *cosine similiarity* yang didapatkan dari nilai pengecekan satu dokumen uji dengan masing-masing dokumen pembanding yang berjumlah sembilan.

Tabel 6. Nilai *Cosine Similarity*

No.	Peneliti (Tahun)	Judul Penelitian	$((A \times B) / (A \times B))$	Hasil
1	Dokumen Pembanding 1	Proceeding of Biology Education	$((0.645828) / (1 \times 1))$	0.0655
2	Dokumen Pembanding 2	Implementasi Pembelajaran Online Berbasis Project Based Learning untuk Meningkatkan	$((0.447567) / (1 \times 1))$	0.0229

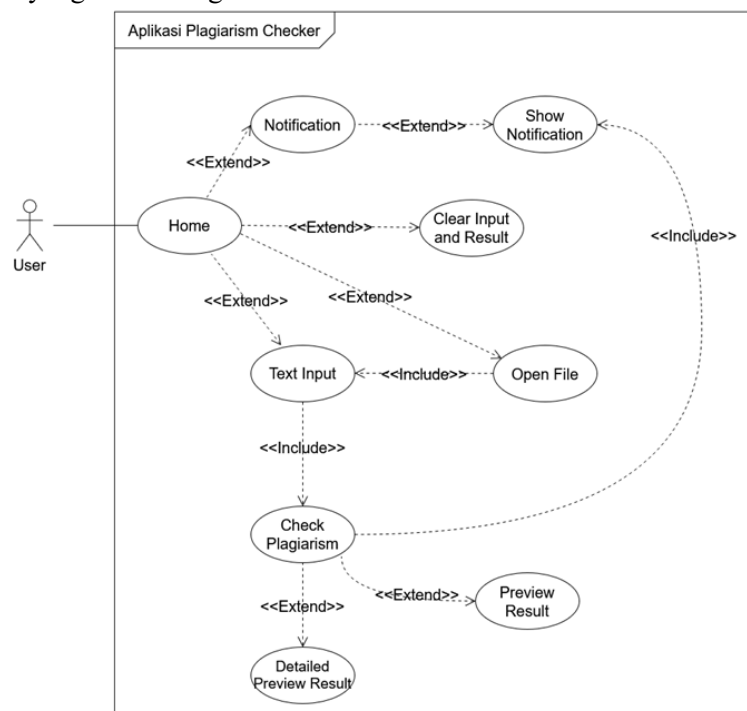
		Kemandirian Belajar Siswa		
3	Dokumen Pembanding 3	Kecanduan Gawai Sebagai Implikasi Pembelajaran Daring Bagi Siswa Sekolah Dasar	$((0.777437) / (1 \times 1))$	0.0128
4	Dokumen Pembanding 4	Inovasi Pemanfaatan Teknologi Sebagai Media Pembelajaran di Era Revolusi 4.0	$((0.611159) / (1 \times 1))$	0.0070
5	Dokumen Pembanding 5	Literasi Digital Peserta Didik Sekolah Dasar: Sebuah Studi Kepustakaan	$((0.477644) / (1 \times 1))$	0.0341
6	Dokumen Pembanding 6	Kontribusi Orang Tua Dalam Pendampingan Belajar Siswa Selama Masa Pandemi	$((0.428126) / (1 \times 1))$	0.0145
7	Dokumen Pembanding 7	Model Komunikasi Orang Tua Dengan Anak	$((0.583030) / (1 \times 1))$	0.0680
8	Dokumen Pembanding 8	Melek Digital Tantangan Guru Saat Pandemi Covid-19	$((0.587601) / (1 \times 1))$	0.0938
9	Dokumen Pembanding 9	Problematika Pembelajaran Daring dalam Penerapan Kurikulum 2013 Tingkat Sekolah Dasar	$((0.664964) / (1 \times 1))$	0.0306

4. Penarikan Kesimpulan

Berdasarkan pengujian sistem dengan cara dilakukan pengecekan plagiarisme menggunakan metode *sklearn* pada dokumen uji dengan dokumen pembanding berjumlah 9 yang di *include* di pengecekan. Berdasarkan pengecekan didapatkan hasil sebesar 9.38% di mana nilai ini tidak termasuk dalam kategori indikasi plagiarisme.

Use Case Diagram

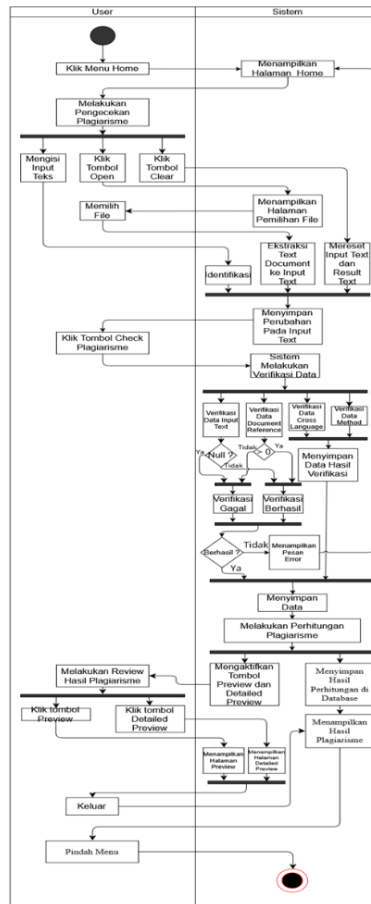
Use case diagram adalah merepresentasikan tugas diskrit yang melibatkan interaksi eksternal atau aktor dengan sistem, sehingga dapat memberikan gambaran yang jelas mengenai konteks dan batasan dari sistem yang dikembangkan



Gambar 3. Use Case Diagram

Activity Diagram

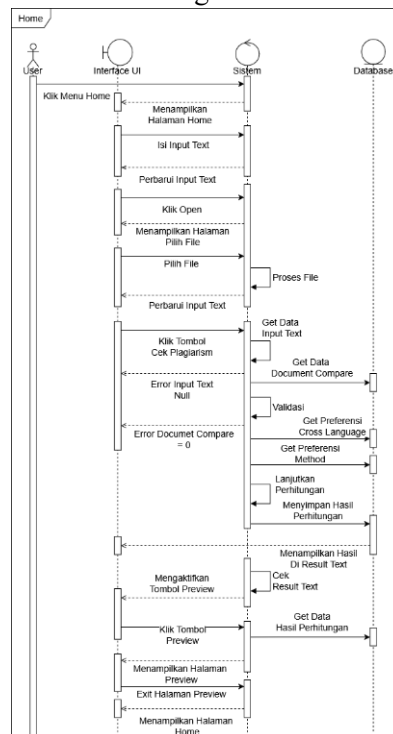
Activity diagram merupakan suatu diagram *Unified Model Language* (UML) digunakan untuk memodelkan bagaimana proses atau alur dari satu aktivitas ke aktivitas lainnya.



Gambar 4. Activity Diagram

Sequence Diagram

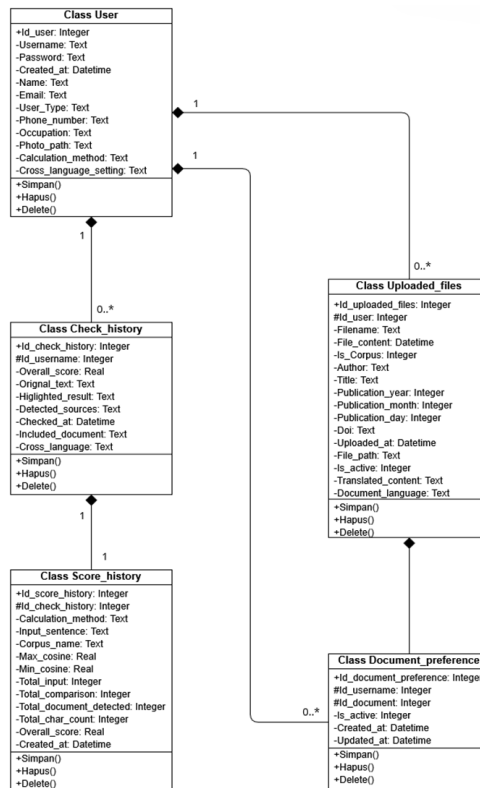
Sequence diagram digunakan untuk menampilkan gambaran bagaimana aktor dan objek dalam sistem berinteraksi dalam menjalankan suatu fungsi atau skenario tertentu.



Gambar 5. Sequence Diagram

Class Diagram

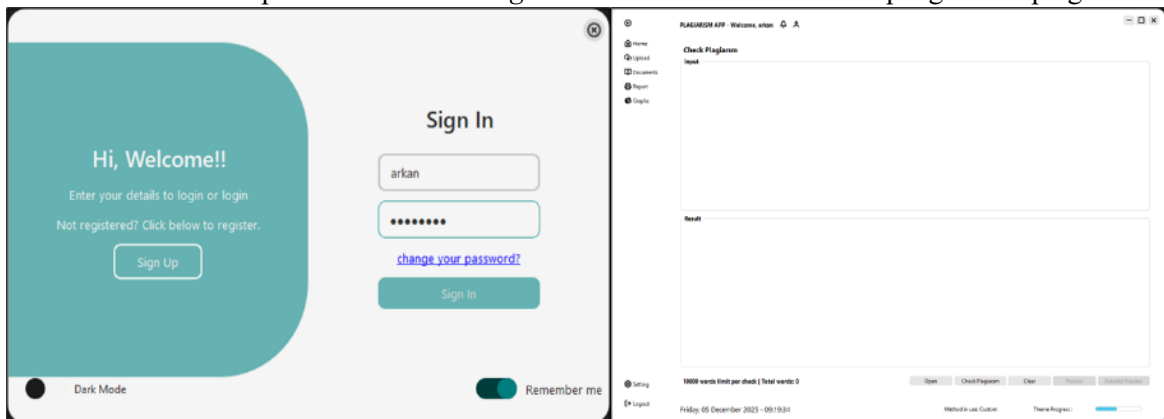
Pada *class* diagram juga memberikan informasi tentang kelas yang dibagi menjadi tiga bagian: nama kelas di bagian atas, atribut kelas di bagian tengah dan operasi yang terkait dengan kelas objek di bagian bawah.



Gambar 6. Class Diagram

Tampilan

Berikut ini adalah tampilan dari halaman *login* dan *home* untuk melakukan pengecekan plagiarisme



Gambar 7. Halaman *login* dan *home*

SIMPULAN

Penelitian ini berhasil menguji dan implementasi aplikasi deteksi plagiarisme *offline* berbasis NLP menggunakan algoritma TF-IDF dan *Cosine Similarity*. Aplikasi yang dibangun dengan Python, PySide6, dan SQLite ini dapat beroperasi tanpa internet, tanpa batasan kata, dan menjamin privasi data. Aplikasi ini menjadi solusi praktis dan gratis bagi mahasiswa serta institusi pendidikan untuk mendeteksi plagiarisme secara mandiri, meskipun masih terbatas pada level leksikal.

DAFTAR PUSTAKA

- Curtis, G. J., & Tremayne, K. (2021). Is Plagiarism Really On The Rise? Results From Four 5-yearly Surveys. *Studies in Higher Education*, 46(9), 1816–1826. <https://doi.org/10.1080/03075079.2019.1707792>.
- Foltýnek, T., Meuschke, N., & Gipp, B. (2019). Academic Plagiarism Detection: A Systematic Literature Review. Dalam *ACM Computing Surveys* (Vol. 52, Nomor 6). Association for Computing Machinery. <https://doi.org/10.1145/3345317>.
- Gregory, A., & Leeman, J. (2021). *On the Perception of Plagiarism in Academia: Context and Intent*. <http://arxiv.org/abs/2104.00574>.
- Halim, J., & Lasut, D. (2024). Document Plagiarism Detection Application Using Web-Based TF-IDF and Cosine Similarity Methods. *bit-Tech*, 7(2), 202–213. <https://doi.org/10.32877/bt.v7i2.1697>.
- Ihle, C., Schubotz, M., Meuschke, N., & Gipp, B. (2020). A First Step Towards Content Protecting Plagiarism Detection. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, 341–344. <https://doi.org/10.1145/3383583.3398620>.
- Karim, A. M., & Zakiyah, E. (2023). *Students' Perception of Plagiarism as a Form of Academic Students' Perception of Plagiarism as a Form of Academic Dishonesty (Case Study on Students of UIN Syarif Jakarta)*. <https://digitalcommons.unl.edu/libphilprac/8001>.
- Meidelfi, D., Rahmayuni, I., Hidayat, T., & Chandra, D. (2021). TF-IDF Implementation for Similarity Checker on The Final Project Title. Dalam *International Journal of Advanced Science Computing and Engineering* (Vol. 3, Nomor 1). <https://doi.org/https://doi.org/10.62527/ijasce.3.1.3>.
- Riyani, A., Zidny Naf'an, M., & Burhanuddin, A. (2019). Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen. Dalam *JLK* (Vol. 2, Nomor 1). <https://www.academia.edu/download/81084477/19.pdf>.
- Singh, S. K., Singh, A., Tiwari, A., Kumar, M., & Chauhan, C. (2023). *Plagiarism Checker Using TF-IDF, Cosine Similarity and Jaccard Similarity* (Vol. 8, Nomor 5). <https://ijnrd.org/papers/IJNRD2305788.pdf>.
- Wibowo, M., Quix, C., Hussien, N. S., Yuliansyah, H., & Adhinata, F. D. (2022). Similarity Identification of Large-Scale Biomedical Documents Using Cosine Similarity and Parallel Computing. *Knowledge Engineering and Data Science*, 4(2), 105. <https://doi.org/10.17977/um018v4i22021p105-116>.