PENDETEKSI KEMIRIPAN TEKS PARAGRAF DALAM DOKUMEN MENGGUNAKAN ALGORITMA *LEACOCK CHODOROW* DAN *COSINE SIMILARITY*

SKRIPSI

ALDRICH WILLIAM CHOALES

181402074

PROGRAM STUDI S1 TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2024

PENDETEKSI KEMIRIPAN TEKS PARAGRAF DALAM DOKUMEN MENGGUNAKAN ALGORITMA *LEACOCK CHODOROW* DAN *COSINE SIMILARITY*

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana Teknologi Informasi

ALDRICH WILLIAM CHOALES

181402074

PROGRAM STUDI S1 TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2024

PERSETUJUAN

Judul	: PENDETEKSI KEMIRIPAN TEKS PARAGRAF DALAM DOKUMEN MENGGUNAKAN ALGORITMA LEACOCK CHODOROW DAN COSINE SIMILARITY
Kategori	: Skripsi
Nama Mahasiswa	: Aldrich William Choales
Nomor Induk Mahasiswa	: 181402074
Program Studi	: Sarjana (S-1) Teknologi Informasi
Fakultas	: Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara

Medan, 10 Januari 2024

Komisi Pembimbing:

Pembimbing 2,

Dr. Erna Budhiarti Nababan, M.IT.

NIP. 196210262017042001

Pembimbing 1,

Prof. Dr. Drs. Opim Salim Sitompul, M.Sc

NIP. 196108171987011001

Diketahui/disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,

Dedy Arisandi S.T., M.Kom.

NIP. 197908312009121002

PERNYATAAN

PENDETEKSI KEMIRIPAN TEKS PARAGRAF DALAM DOKUMEN MENGGUNAKAN ALGORITMA *LEACOCK CHODOROW* DAN *COSINE SIMILARITY*

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 11 Januari 2024

Aldrich William Choales

181402074

UCAPAN TERIMAKASIH

Dengan penuh rasa syukur yang tak terhingga, penulis ingin menyampaikan penghargaan serta terima kasih kepada Tuhan Yang Maha Esa atas anugerah-Nya yang memungkinkan penulis melakukan penyelesaian penulisan skripsi ini. Skripsi ini ditulis untuk memenuhi syarat memperoleh gelar Sarjana pada Program Studi S1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara. Penulis juga ingin menyampaikan rasa terima kasih yang besar kepada semua pihak yang terlibat dan memberikan bantuan selama perjalanan perkuliahan, yang pada akhirnya memungkinkan penulis menyelesaikan penyusunan skripsi ini. Penulis tidak dapat menyelesaikan penyusunan skripsi ini tanpa bimbingan, doa, dan dukungan dari berbagai pihak yang terlibat dalam masa perkuliahan hingga penyusunan skripsi ini selesai. Adapun dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Keluarga penulis, Ayah Susanto, Ibu Lilywaty, bersama saudari serta saudara penulis, Angeline Candice Choales dan Andrew Ryan Choales yang terus menerus memberikan kasih sayang, doa, dukungan, serta semangat kepada penulis sehingga dapat menyelesaikan skripsi ini
2. Bapak Prof. Dr. Mulyanto Amin, M.Si., selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc., selaku Dekan Fasilkom-TI USU.
4. Bapak Dedy Arisandi, S.TI., M.Kom., selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Prof. Dr. Drs. Opim Salim Sitompul, M.Sc selaku Dosen Pembimbing Pertama serta Ibu Dr. Erna Budhiarti Nababan, M.IT., selaku Dosen Pembimbing Kedua yang telah meluangkan waktu untuk membimbing, memberikan masukan, kritik serta masukan kepada penulis selama proses pengerjaan skripsi dari awal hingga akhir.
6. Bapak Ivan Jaya, S.Si., M.Kom. sebagai Dosen Pembanding Pertama dan Ibu Dr. Marischa Elveny S.TI., M.Kom. sebagai Dosen Pembanding Kedua yang banyak memberikan kritik serta masukan terhadap hasil penelitian penulis dan penulisan skripsi penulis.

7. Penulis mengucapkan terimakasih kepada seluruh Dosen, Staff, dan Pegawai dari Program Studi S1 Teknologi Informasi yang telah memberikan pengajaran ilmu yang sangat berharga sepanjang masa perkuliahan dan bantuan dalam berbagai urusan administrasi selama masa studi dan dalam proses penyelesaian penulisan skripsi.
8. Kepada penulis sendiri yang terus berusaha dengan tekun tanpa putus asa dalam pengerjaan tugas akhir, walaupun harus melewati banyak rintangan hidup selama proses penyelesaian tugas akhir ini.
9. Sahabat seperjuangan penulis semasa kuliah, Willi Nardo, Naldo Yohardi, Endity Wasita Angkasa, Jimmy Widianto, Xixilia Sunaryo, Kevin Patrick Lee, serta William Yuhandinata yang telah menemani dan memberikan semangat juga dukungan kepada penulis.
10. Teman teman Angkatan 2018 Teknologi Informasi USU yang telah bersama-sama berjuang bersama penulis dari semasa perkuliahan hingga selesai penyusunan skripsi
11. Kepada senior, junior, dan teman-teman lainnya yang tidak dapat penulis sebut satu persatu yang telah memberikan semangat dan kenangan indah semasa kuliah hingga selesai penyusunan skripsi.
12. Semua pihak yang terlibat langsung maupun tidak langsung yang tidak dapat penulis tuliskan satu persatu yang telah membantu dalam penyelesaian skripsi ini.

Medan, 11 Januari 2024

Penulis

ABSTRAK

Plagiarisme adalah tindakan penyalinan atau peniruan secara dekat, pengambilan karya dari penulis ataupun pencipta lainnya tanpa terlebih dahulu meminta izin dengan maksud mengambil karya menjadi milik ataupun ciptaan asli sendiri. Plagiarisme menjadi masalah yang sering terjadi di kalangan umum maupun kalangan akademis. Dalam segi akademis, plagiarisme menjadi masalah yang besar karena verifikasi keaslian suatu dokumen memakan waktu yang lama dengan tingkat presisi yang beragam. Maka dari itu, perlu dikembangkan suatu pendekatan dengan tujuan melakukan deteksi kemiripan paragraf dalam dokumen. Penelitian ini memiliki tujuan untuk menanggulangi masalah dengan melakukan deteksi kemiripan isi dokumen karya ilmiah berdasarkan paragraf secara otomatis dengan menggunakan algoritma *Leacock Chodorow* dan *cosine similarity*. Dalam pengujian sistem digunakan data uji dan data pembanding sebanyak 28 dokumen dengan dokumen uji sebanyak 7 dokumen dengan 146 total paragraf uji serta dokumen pembanding sebanyak 21 dokumen dengan 438 total paragraf pembanding. Hasil evaluasi dari sistem pendeteksi kemiripan paragraf mendapatkan nilai akurasi sebesar 0.923 atau 92.3%, nilai presisi sebesar 0.908 atau 90.8%, nilai *recall* sebesar 0.953 atau 95.3%, serta nilai *F-measure* sebesar 0.930 atau 93%.

Kata Kunci: Kemiripan Teks, Kemiripan *Leacock Chodorow*, Kemiripan Kosinus, Plagiarisme

DETECTION OF TEXT PARAGRAPH SIMILARITY IN DOCUMENTS USING LEACOCK CHODOROW ALGORITHM AND COSINE SIMILARITY

ABSTRACT

Plagiarism is the act of closely copying or imitating, taking works from authors or creators without prior permission, with the intention of claiming the work as one's own original creation. Plagiarism is a common issue in both general and academic settings. In the academic context, plagiarism poses a significant problem because verifying the authenticity of a document is time-consuming and varies in terms of precision. Therefore, an approach need to be to developed aimed at detecting paragraph similarity in documents. This research aims to address the issue by automatically detecting similarity in the content of scholarly documents based on paragraphs, using the Leacock Chodorow algorithm and cosine similarity. In the system testing, 28 test data and reference data are used, with 7 test documents consisting of a total of 177 test paragraphs and 21 reference documents consisting of a total of 438 reference paragraphs. The evaluation results of the paragraph similarity detection system obtained an accuracy of 0.923 or 92.3%, precision of 0.908 or 90.8%, recall of 0.953 or 95.3%, and an F-measure of 0.930 or 93%.

Keyword: *Text Similarity, Leacock Chodorow Similarity, Cosine Similarity, Plagiarism*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi pengumpulan informasi pada masa modern sekarang berkembang serta mengalami kemajuan pesat. Perkembangan dari teknologi ini menjadi suatu alat yang dipakai untuk membantu dan memudahkan berbagai kegiatan manusia. Salah satu pengaruh besar yang tampak secara jelas dari perkembangan ini adalah semakin mudahnya seseorang untuk dapat mencari karya ilmiah baik dalam bentuk dokumen maupun teks hanya dengan melakukan *browsing* di *internet*.

Secara umum, mengingkatnya kemampuan dan semakin mudahnya mengumpulkan informasi berdampak positif bagi manusia. Namun, dengan peningkatan kemampuan dan kemudahan tersebut muncul juga masalah yang perlu diperhatikan. Plagiarisme; yaitu tindakan penyalinan atau peniruan secara dekat, pengambilan karya dari penulis ataupun pencipta lainnya tanpa terlebih dahulu meminta izin dengan maksud mengambil karya menjadi milik ataupun ciptaan asli sendiri (Reitz, 2004) menjadi masalah yang sering terjadi di kalangan umum maupun kalangan akademis. Dalam segi akademis, plagiarisme menjadi masalah yang besar (Fatonah, 2020) karena verifikasi keaslian suatu dokumen memakan waktu yang lama dengan tingkat presisi yang beragam. Seorang penulis wajib melakukan parafrase dan pengutipan/sitasi terkait dokumen karya ilmiah orang lain apabila penulis ingin menggunakannya sebagai referensi untuk menghindari tindak plagiarisme.

Alasan kesamaan teks pada dokumen karya ilmiah sering terjadi adalah dikarenakan kemudahan mengumpulkan informasi dari *internet* dan juga kurangnya perhatian dari penulis untuk melakukan parafrase serta pengutipan ataupun dikarenakan kesengajaan oleh penulis yang mengambil karya orang lain tanpa melakukan parafrase maupun pengutipan. Dari penjelasan latar belakang sebelumnya, perlu dikembangkan suatu pendekatan dengan tujuan melakukan deteksi kemiripan paragraf dalam dokumen secara lebih efisien dari segi struktur kata dan makna.

Bedasarkan latar belakang permasalahan yang dipaparkan, dalam penelitian ini penulis menggunakan algoritma *Leacock Chodorow* serta *cosine similarity* untuk melakukan deteksi kemiripan dokumen karya ilmiah berdasarkan paragraf. Deteksi merujuk pada langkah-langkah yang diambil untuk mengenali jenis masalah atau menemukan solusi dalam suatu situasi. Tujuannya adalah untuk mendukung pengambilan keputusan atau pembuatan kesimpulan dengan memahami permasalahan yang dihadapi.

Sebelumnya telah dilakukan beberapa penelitian tentang pendeteksian kemiripan teks pada dokumen diantaranya adalah Firdaus *et al* (2014), Gokul *et al* (2017), Soyusiawaty & Zakaria (2018), Madani *et al* (2018) dan ChIru *et al* (2021).

Firdaus *et al* (2014) melakukan penelitian tentang pendeteksian dokumen teks dengan implementasi metode *cosine similarity* serta algoritma Nazief & Andriani, pada penelitian ini terlebih dahulu kata diubah kedalam bentuk dasar dengan menggunakan algoritma Nazief & Andriani untuk selanjutnya dilakukan perhitungan dengan menggunakan metode *cosine similarity.*

Gokul *et al* (2017) melakukan penelitian tentang implementasi *cosine similarity* untuk mendeteksi kemiripan teks. Penelitian ini melakukan aplikasi algoritma *cosine similarity* untuk melakukan pengukuran terhadap tingkat kemiripan antar kalimat dengan mencari kemungkinan parafrase kalimat dalam bahasa *Malayalam*.

Soyusiawaty & Zakaria (2018) melakukan penelitian tentang implementasi *cosine similarity* dalam pendeteksian kemiripan antar dokumen. Penelitian ini melakukan aplikasi algoritma *cosine similarity* untuk melakukan pengukuran kemiripan isi antar dokumen berdasarkan peringkat nilai tertinggi hingga terendah.

Madani *et al* (2018) melakukan implementasi algoritma *Leacock Chodorow* dalam dalam dua pendekatan untuk mengklasifikasi *tweet*. Penelitian mengusulkan pendekatan klasifikasi *tweet* menurut tiga kelas : negatif, positif atau netral, menurut dua kelas: negatif atau positif. Penelitian menggunakan algoritma *Leacock Chodorow* dikarenakan pendekatan tersebut menghasilkan tingkat kemiripan terbesar dengan waktu eksekusi yang paling cepat.

Chiru *et al* (2021) melakukan terdahulu yang berkaitan dengan penggunaan algoritma *Leacock Chodorow*. Penelitian menggunakan beberapa algoritma untuk menghitung nilai kemiripan teks dalam *WordNet*, yaitu *Leacock Chodorow, Wu Palmer,* dan *path similarity*. Hasil dari perbandingan menunjukkan bahwa *Leacock Chodorow* dapat hasil paling dekat dengan *Word2Vec cosine similarity.*

Cosine similarity dapat didefiniskan sebagai sebuah algoritma yang umumnya digunakan dalam pengukuran kesamaan teks antar vektor. Algoritma ini merupakan hasil perkalian *inner product space* yang diukur dengan fungsi cosinus untuk melihat arah dua vektor dan menentukan kesamaan arahnya (Han, et al. 2012).

Leacock Chodorow merupakan algoritma *semantic similarity* yang dapat digunakan untuk mengukur derajat keterkaitan antar semantik. Algoritma ini merupakan hasil perluasan dari algoritma *path-based similarity* dengan cara mengikut sertakan *maximum depth of taxonomy* (*Leacock* & *Chodorow*, 1998)

Bedasarkan latar belakang permasalahan yang dipaparkan dan juga beberapa penelitian terdahulu sebagai pendukung, penulis melakukan penelitian dengan judul "*PENDETEKSI KEMIRIPAN PARAGRAF DALAM DOKUMEN MENGGUNAKAN ALGORITMA LEACOCK CHODOROW SIMILARITY DAN COSINE SIMILARITY*".

1.2. Rumusan Masalah

Kesamaan teks pada dokumen karya ilmiah sering terjadi karena mudahnya pengumpulan informasi dari *internet* dan juga kurangnya perhatian dari penulis untuk melakukan parafrase serta pengutipan ataupun kesengajaan penulis yang mengambil karya orang lain tanpa melakukan parafrase ataupun pengutipan. Dari penjelasan latar belakang sebelumnya, perlu dikembangkan suatu pendekatan dengan tujuan melakukan deteksi kemiripan paragraf dalam dokumen.

1.3. Batasan Masalah

Terdapat ruang lingkup ataupun batasan masalah dari penelitian yang diteliti. Ruang lingkup atau batasan masalah penelitian yang diteliti adalah:

- Bahasa dalam dokumen yang dapat digunakan berupa bahasa Indonesia dan Inggris.
- Data yang akan diuji berupa data teks.
- Data teks didapatkan dari bagian tinjauan pustaka atau landasan teori dokumen yang digunakan.
- Dokumen yang tidak dapat diakses atau terkunci tidak dapat diproses.
- Data teks yang digunakan sebagai dokumen pembanding didapatkan dengan cara *browsing* secara manual.

1.4. Tujuan Penelitian

Penelitian memiliki tujuan melakukan deteksi kemiripan isi dokumen karya ilmiah berdasarkan paragraf secara otomatis dengan menggunakan algoritma *Leacock Chodorow* dan *cosine similarity*.

1.5. Manfaat Penelitian

Terdapat kegunaan atau keuntungan penelitian yang diharap untuk didapatkan. Manfaat yang diharap bisa dihasilkan dari penelitian adalah:

1. Memudahkan penguji dalam proses pengecekan kemiripan isi dokumen karya ilmiah berdasarkan paragraf secara otomatis.
2. Mengurangi waktu proses pengecekan kemiripan isi dokumen karya ilmiah dibandingkan dengan pengecekan secara manual.
3. Meningkatkan tingkat presisi dalam proses pengecekan kemiripan isi dokumen karya ilmiah.

1.6. Sistematika Penulisan

Sistematika dalam proses mengerjakan dan menulis penelitian ini dapat dibagi kedalam beberapa bagian inti atau utama. Sistematika yang digunakan dalam penulisan penelitian ini adalah:

Bab I: Pendahuluan

Bagian memiliki isi berupa penjelasan tentang latar belakang, penjelasan rumusan masalah, penjelasan batasan masalah, penjelasan tujuan penelitian, penjelasan manfaat penelitian, penjelasan metodologi penelitian, serta penjelasan sistematika penulisan.

Bab II: Landasan Teori

Bagian memiliki isi berupa penjelasan secara menyeluruh tentang teori penelitian, yaitu teori yang dipakai dan diperlukan guna menyelesaikan masalah dalam penelitian. Bab akan menjelaskan secara terperinci tentang *WordNet*, *Leacock Chodorow*, dan *cosine similarity*.

Bab III: Analisis dan Perancangan Sistem

Bagian memiliki isi berbentuk penjelasan hasil Analisa berserta dengan langkah menerapkan *WordNet*, *Leacock Chodorow*, dan *cosine similarity* untuk melakukan proses perhitungan. Nilai *word similarity* yang didapat dengan metode *Leacock Chodorow* akan diproses dengan menggunakan *cosine similarity* sehingga didapatkan nilai kemiripan antar paragraf

Bab IV: Implementasi dan Pengujian

Bagian memiliki isi berupa penjelasan implementasi terhadap rancangan penerapan sistem penelitian dari penjabaran rancangan sebelumnya dalam bab 3. Selain itu, Bab akan melakukan penjelasan terhadap hasil pengujian setelah tahap pengujian sistem selesai dilakukan.

Bab V: Kesimpulan dan Saran

Bagian memilki isi berupa ringkasan penelitian serta konklusi berdasarkan perolehan implementasi sistem serta pemecahan masalah dari penjabaran masalah sebelumnya dalam bab 4. Bab juga akan memberikan saran untuk mengembangkan penelitian dengan harapan bahwa saran dapat dipakai serta dilakukan dalam penelitian selanjutnya

BAB II

LANDASAN TEORI

2.1. Deteksi

Deteksi dapat didefinisikan sebagai langkah-langkah yang diambil untuk mengenali jenis masalah atau menemukan solusi dalam suatu situasi. Tujuan dilakukannya deteksi adalah untuk mendukung dalam pengambilan keputusan atau pembuatan kesimpulan dengan memahami permasalahan yang dihadapi.

2.2. Kemiripan Teks

Kemiripan teks menentukan derajat kedekatan antara dua bagian teks dari kemiripan leksikal yang merupakan kemiripan segi permukaan atau kemiripan semantik yang merupakan kemiripan makna. Kemiripan leksikal berfokus pada apakah dua bagian teks terdiri dari kata-kata yang sama, sedangkan kemiripan semantik berfokus pada kedekatan makna antara kata-kata yang membentuk dua bagian teks teks (Ganesan, 2015).

2.3. *Leacock Chodorow (LCH)*

Leacock Chodorow merupakan algoritma *semantic similarity* yang dapat digunakan untuk mengukur derajat keterkaitan antar dokumen. Algoritma ini merupakan hasil perluasan dari algoritma *Path-Based Similarity* dengan cara mengikut sertakan *maximum depth of taxonomy* (Leacock & Chodorow, 1998). Oleh karena itu, algoritma ini mengambil log negatif dari *shortest path* (spath) antara dua konsep (*synset_1* dan *synset_2*), dibagi dua kali total *depth of taxonomy* (*Depth*). *Leacock Chodorow* dirumuskan pada persamaan 2.1.

$$LCH\ Similarity = -log \frac{spath\ (synset1,synset2)}{2*Depth} \quad (2.1)$$

Dengan keterangan:

spath (synset1,synset2) = jarak terpendek antara *synset1* dan *synset2*

Depth = kedalaman maksimun taksonomi

2.4. *Cosine Similarity*

Cosine Similarity dapat dijelaskan sebagai sebuah algoritma dengan fungsi melakukan pengukuran nilai kemiripan teks antar vektor. Algoritma ini merupakan hasil perkalian *inner product space* yang diukur dengan fungsi cosinus untuk melihat apakah dua vektor memiliki arah yang sama atau tidak (Han, et al. 2012). Proses *Cosine Similarity* dirumuskan pada persamaan 2.2.

$$CosSim(M,N) = \frac{M*N}{||M|| * ||N||} \quad (2.2)$$

Dengan keterangan:

M = *Vector* M

N = *Vector* N

||M|| = Eculedian Norm *Vector* M = $\sqrt{M_1^2 + M_2^2 + \cdots + M_n^2}$

||N|| = Eculedian Norm *Vector* N = $\sqrt{V_1^2 + V_2^2 + \cdots + V_n^2}$

2.5. *WordNet*

WordNet adalah jaringan semantik *online* yang desainnya terinspirasi oleh teori psikolinguistik terkini dari memori leksikal manusia (Miller et al. 1990). Kamus *WordNet* berisi kumpulan jenis kata seperti *noun* (kata benda), *verb* (kata kerja), *adjective* (kata sifat), serta *adverb* (kata keterangan) dalam bahasa Inggris yang diatur ke dalam kumpulan *synsets* (*synonim sets*) yang masing-masing mewakili satu konsep leksikal yang mendasarinya. Hubungan yang berbeda menghubungkan kumpulan sinonim. *WordNet* berisi sekitar 155.327 kata, 175.979 *synsets* dan 207.016 pasangan kata. Keunggulan yang dimiliki *WordNet* berkat konsep *synset* nya adalah kemampuan nya dalam melihat kedekatan kata dari segi holonim, meronim, hipernim, hiponim, sinonim, dan antonim.

2.6. *Natural Language Processing (NLP)*

NLP dapat dijelaskan sebagai sebuah cabang ilmu dengan asal dari ilmu kecerdasan buatan (Artifi*cial Intelligence*) dimana fokus utamanya adalah penerjemahan bahasa manusia. *NLP* memberikan pemahaman pada komputer terhadap teks dan kata seperti yang dilakukan oleh manusia. *NLP* adalah teknik komputasi yang digunakan untuk pemrosesan teks kedalam sebuah sistem komputer pada tingkat analisis ilmu bahasa satu atau lebih dengan maksud melakukan pemrosesan bahasa yang mirip dengan bahasa manusia (Liddy, 2001).

2.7. *Text Mining*

Text mining adalah praktik ekstraksi informasi dari data dokumen berupa teks dengan tujuan menemukan kata-kata yang mencerminkan isi dokumen secara representatif. Proses ini bertujuan untuk memperoleh informasi berharga dari kumpulan dokumen melalui analisis teks, memungkinkan analisis keterhubungan antar dokumen untuk tujuan tertentu. Untuk melakukan analisis lebih lanjut, data teks dalam *text mining* perlu mengalami proses pengubahan menjadi bentuk data numerik. Oleh karena itu, istilah preprocessing data digunakan, mengacu pada langkah-langkah awal yang diterapkan pada data teks untuk menghasilkan representasi numerik (Wahyuni, et al., 2017). Definisi *text mining* merujuk pada proses penemuan pola tersembunyi, berguna, dan menarik dari dokumen yang tidak terstruktur. Tahap ini diperlukan karena kompleksitas serta tidak terstrukturnya teks dalam sebuah dokumen (Syaifudin et al., 2018).

2.8. *Pre-processing*

Tahap *preprocessing* didefinisikan sebagai suatu tahap dimana dilakukan persiapan terhadap data yang akan digunakan dalam penelitian. Data yang akan dilakukan persiapan termasuk teks ataupun dokumen dimana konten ataupun informasi perlu digunakan. (Syaifudin et al, 2018).

2.8.1 *Case Folding*

Case folding merupakan suatu proses dimana dilakukan perubahan terhadap format data dengan tujuan mengurangi redundansi pada proses klasifikasi, menghasilkan perhitungan yang lebih optimal. Proses ini melibatkan pengubahan format data menjadi huruf kecil (*lowercase*) atau huruf besar (*uppercase*) sesuai kebutuhan (Muttaqin & Bachtiar, 2019). Penggunaan huruf kecil serta huruf besar dapat menentukan interpretasi semantik, sehingga berpengaruh dalam *text mining*. Huruf besar digunakan dalam teks atas berbagai alasan, seperti awal kalimat, sebagai bagian dari judul, atau saat menggunakan nama khusus (Petrović & Stanković, 2019). Dalam penelitian, *case folding* dilakukan terhadap data untuk meningkatkan akurasi serta menghindari masalah interpretasi oleh sistem.

2.8.2 *Tokenizing*

Token dijelaskan sebagai urutan karakter yang kontinu dengan makna semantik, dengan tambahan bahwa *token* dapat diulang tanpa pemrosesan tambahan (seperti *stemming*). *Tokenizing* merupakan sebuah proses di mana teks dipecah menjadi per bagian atau *token*, sebagian besar sesuai dengan kata dalam bahasa yang teksnya ditulis (Petrović & Stanković, 2019). Dalam penelitian, *tokenizing* dilakukan terhadap data dalam upaya memudahkan proses sistem karena *tokenizing* memecah kalimat pada paragraf data kedalam kata sehingga proses perhitungan nilai kemiripan menggunakan algoritma *Leacock Chodorow* serta *cosine similarity* menjadi lebih optimal, meningkatkan akurasi sistem.

2.8.3 *Stemming*

Stemming merupakan sebuah proses dimana dilakukan perubahan kata dalam teks kedalam bentuk dasar (Syaifudin et al, 2018). *Stemming* memerlukan perhatian khusus dikarenakan proses ini memiliki aplikasi yang sangat tergantung pada bahasa teks yang dilakukan analisis, dan dengan demikian, pengaruh proses pada sistem tergantung terhadap bahasa yang dipakai (Petrović & Stanković, 2019). Dalam penelitian, *stemming* dilakukan terhadap data agar kata dengan bahasa Indonesia dapat dideteksi dengan baik. *Stemming* tidak dilakukan terhadap teks bahasa Inggris atas dasar proses melakukan penghapusan akhiran terhadap banyak kata bahasa Inggris yang seharusnya tidak perlu dihapus, menyebabkan hilangnya makna dari kata secara menyeluruh

2.8.4 *Filtering*

Filtering merupakan sebuah proses dimana dilakukan pengambilan kata-kata penting yang dihasilkan dari proses tokenisasi. Pada tahap ini, kata yang dianggap tidak begitu penting oleh sistem akan dibuang dengan melakukan penghapusan *stopwords*, sementara kata-kata yang dianggap penting disimpan (Syaifudin et al, 2018). *Stopword* merupakan daftar kata umum yang ditemukan dalam setiap bahasa, namun tidak memberikan kontribusi pada keragaman konten (Petrović & Stanković, 2019). Dalam penelitian, *filtering* dilakukan terhadap data untuk mempercepat proses perhitungan nilai kemiripan.

2.9. Penelitian Terdahulu

Penelitian sebelumnya yang memiliki kaitan dengan kemiripan dokumen teks dilakukan oleh Firdaus *et al* (2014) dengan menggunakan metode *Cosine Similarity* dan algoritma Nazief & Andriani. Pada penelitian ini dokumen - dokumen yang akan diuji terlebih dahulu melalui proses *preprocessing.* Khusus pada bagian *stemming* akan diterapkan algoritma Nazief & Andriani. Algoritma ini dipilih karena pengembangan algoritma ini didasarkan pada morfologi dari Bahasa Indonesia dimana imbuhan telah dikelompokkan menjadi bagian awalan, sisipan, dan akhiran. Setelah proses *stemming* selesai maka dilakukan perhitungan kemiripan dengan menghitung *term frequency* kata. Penelitian dengan bantuan algoritma Nazief & Andriani memperoleh hasil 5,98% lebih tinggi.

Penelitian terdahulu yang berkaitan dengan kemiripan teks lainya pernah juga dilakukan Imbar *et al* (2014). Penelitian mengimplementasi Algoritma *Smith-Waterman* dan *Cosine Similarity* dimana algoritma tersebut berfungsi untuk mendeteksi kemiripan teks. Penelitian ini menggunakan *Cosine Similarity* untuk mengukur kesamaan teks berdasarkan struktur teks, serta algoritma *Smith – Waterman* dipakai dengan fungsi melakukan perhitungan kemiripan teks yang didasari susunan kata dan *ouput* yang diberikan adalah hasil kemiripan dari struktur teks dan hasil kemiripan dari urutan kata.

Penelitian terdahulu yang berkaitan dengan penggunaan metode *Cosine Similarity* pernah dilakukan oleh Ariantini *et al* (2016). Penelitian menghitung tingkat kemiripan dengan cara melakukan perhitungan dari nilai *TF* (*Term Frequency*) guna menghitung kemunculan dari kata dan kemudian melakukan perhitungan kemiripan dokumen teks dengan metode *Cosine Similarity* dan menyimpannya ke *database*.

Penelitian terdahulu yang berkaitan dengan kemiripan teks pernah dilakukan oleh Gokul *et al* (2017). Penelitian ini melakukan aplikasi algoritma *cosine similarity* untuk melakukan pengukuran terhadap tingkat kemiripan antar kalimat dengan mencari kemungkinan parafrase kalimat dalam Bahasa *Malayalam*. Penelitian menggunakan data uji berupa 900 dan 1400 pasang kalimat dari corpus *FIRE 2016 Malayalam* yang dari dua iterasi menghasilkan output dengan tingkat akurasi 0.8 dan 0.59.

Penelitian terdahulu yang berkaitan dengan kemiripan teks dan penggunaan metode *Cosine Similarity* pernah diterapkan oleh Soyusiawaty & Zakaria (2018). Penelitian menggunakan *Cosine Similarity* dalam pengukuran kemiripan isi antara dua dokumen berdasarkan peringkat nilai tertinggi hingga terendah. Penelitian dengan penerapan *Cosine Similarity* menghasilkan output yang diinginkan dengan tingkat akurasi 82.14%.

Penelitian terdahulu yang menjelaskan keunggulan algoritma *Leacock Chodorow* dilakukan oleh Madani *et al* (2018). Penelitian ini mengusulkan dua pendekatan untuk mengklasifikasikan *tweet*, yang pertama menurut *class* positif, netral atau negatif (tiga *class*), dan yang kedua menurut *class* positif atau negatif (dua *class*). Penelitian menggunakan algoritma *Leacock Chodorow* dikarenakan pendekatan tersebut menghasilkan tingkat kemiripan terbesar dengan waktu eksekusi yang paling cepat. Pendekatan yang diusulkan memiliki akurasi 93 dan 91%, presisi sebesar 80 dan 81%, *recall* sebesar 85 dan 84% dan skor F1 sebesar 82 dan 83%.

Penelitian terdahulu yang berkaitan dengan kemiripan teks dan penggunaan metode *Cosine Similarity* yang lainnya pernah dilakukan oleh Hartono *et al* (2021). Penelitian ini menggunakan dan menerapkan algoritma *Rabin-Karp* serta metode *Cosine Similarity* sebagai *Distance-Based Similarity Measure* dan menggunakan teknik *stemming*. Penelitian mencari kata dasar dalam abstrak dengan menggunakan *stemming* dan setelah membentuk *gram* dan mencari nilai *hash*, nilai *hash* tersebut dipakai dalam perbandingan nilai *hash* tugas akhir yang terdapat didalam *database*.

Penelitian terdahulu yang berkaitan dengan penggunaan algoritma *Leacock Chodorow* juga dilakukan oleh Chiru *et al* (2021). Penelitian menggunakan beberapa algoritma dalam menghitung nilai kemiripan teks dalam *WordNet*, yaitu *Path Similarity*, *Wu Palmer*, dan *Leacock Chodorow*. Hasil dari perbandingan menunjukkan bahwa *Leacock Chodorow* mendapat hasil paling dekat dengan *Word2Vec Cosine Similarity*. Rangkuman seluruh penelitian sebelumnya terkait penelitian ini seperti pada Tabel 2.1

Tabel 2.1 Penelitian Terdahulu

No.	Peneliti	Metode	Keterangan
1	Firdaus *et al.* (2014)	*Cosine Similarity* Nazief & Andriani	Penelitian menerapkan *stemming* dengan algoritma Nazief & Andriani serta perhitungan kemiripan dengan algoritma *Cosine Similarity* dan didapatkan jika pengujian dengan *stemming* Nazief & Andriani menghasilkan nilai lebih tinggi 5.98%.
2	Imbar *et al.* (2014)	*Cosine Similarity* Smith-Waterman	Penelitian mendeteksi kemiripan teks dengan menggunakan dua cara yaitu dari struktur kata dengan menggunakan algoritma *Cosine Similarity* dan dari segi urutan kata dengan algoritma *Smith-Waterman.*
3	Ariantini *et al.* (2016)	*Cosine Similarity*	Penelitian menghitung tingkat kemiripan dengan melakukan perhitungan nilai *TF (Term Frekuensi)* guna menghitung kemunculan dari kata lalu menghitung kemiripan dokumen teks dengan metode *Cosine Similarity* dan menyimpannya ke *database.* Selisih antara nilai aktual dan prediksi 9 - 15%
4	Gokul *et al* (2017)	*Cosine Similarity*	Penelitian menggunakan *Cosine Similarity* untuk menghitung tingkat kemiripan antar kalimat. Penelitian menghasilkan output dengan tingkat akurasi 0.8 dan 0.59

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

No.	Peneliti	Metode	Keterangan
5	Soyusiawaty & Zakaria (2018)	*Cosine Similarity*	Penelitian menggunakan *Cosine Similarity* dalam upaya pengukuran kemiripan isi antara dua dokumen. Penelitian menghasilkan output yang diinginkan dengan tingkat akurasi 82.14%.
6	Madani *et al* (2018)	*Leacock Chodorow*	Penelitian dilakukan menggunakan algoritma *Leacock Chodorow* dalam dua pendekatan untuk melakukan klasifikasi *tweet*. Pendekatan yang diusulkan memiliki akurasi 93 dan 91%, presisi sebesar 80 dan 81%, *recall* sebesar 85 dan 84% dan skor F1 sebesar 82 dan 83%.
7	Hartono *et al* (2021)	*Rabin-Karb Cosine Similarity*	Penelitian menggunakan algoritma *Rabin-Karp* serta *Cosine Similarity* dalam upaya menghitung tingkat kemiripan antar kalimat.
8	Chiru *et al* (2021)	*Path Similarity Wu Palmer Leacock Chodorow*	Penelitian menggunakan algoritma *Leacock Chodorow*, *Wu Palmer*, dan *Path Similarity*. Penelitian menunjukkan *Leacock Chodorow* menghasilkan hasil paling dekat dengan *Word2Vec Cosine Similarity*

2.10. Perbedaan Penelitian Terdahulu

Terdapat beberapa perbedaan yang dapat dilihat dalam penelitian ini dibanding dengen penelitian terdahulu. Adapun perbedaan yang terdapat pada penelitian ini adalah metode yang digunakan dalam penelitian, dimana penelitian ini melakukan penggunaan *Leacock Chodorow* untuk mengukur derajat keterkaitan antar dokumen serta penggunaan *cosine similarity* untuk melakukan pengukuran nilai kemiripan teks antar vektor secara bersamaan. Pada penelitian terdahulu, metode yang digunakan untuk menghitung tingkat kemiripan antar kalimat berbeda dengan penelitian ini ataupun terbatas kepada penggunaan salah satu dari metode yang digunakan dalam penelitian ini.

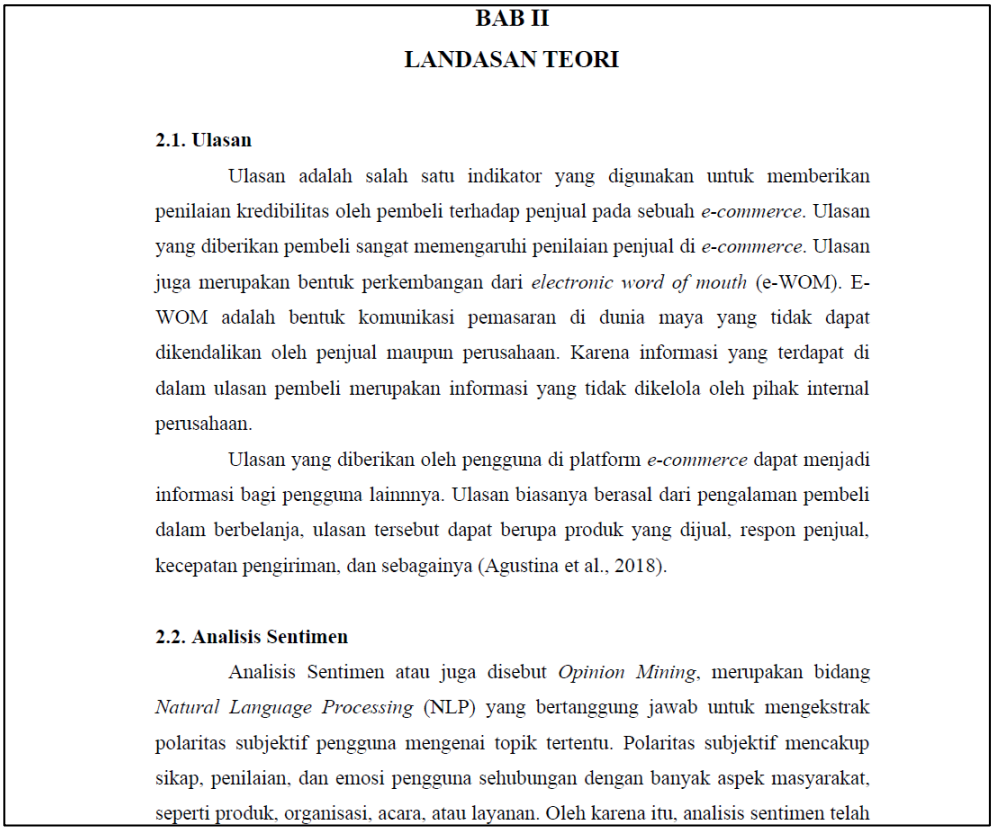
Penelitian ini menggunakan dokumen uji serta dokumen banding berupa tinjauan pustaka dan landasan teori dari skripsi mahasiswa sebagai fokusnya. Selain itu Bahasa yang dapat diproses oleh sistem dalam penelitian ini juga berbeda dibandingkan dengan bahasa yang digunakan pada penelitian terdahulu.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Data

Untuk data, penelitian akan memakai dokumen uji yang diambil melalui *website* repositori USU dengan ketentuan pemilihan yang acak dari beberapa Angkatan yang berbeda (2018, 2017, 2016) dengan tujuan menambah variasi data, sedangkan untuk dokumen pembanding akan diperoleh melalui pencarian dokumen referensi yang didapatkan dengan memperhatikan sitasi pada paragraf yang terdapat dalam data uji serta dokumen yang digunakan sebagai referensi penelitian dari. daftar isi yang terdapat dalam data uji. Contoh dokumen uji seperti pada Gambar 3.1 serta dokumen pembanding seperti pada Gambar 3.2



BAB II

LANDASAN TEORI

2.1. Ulasan

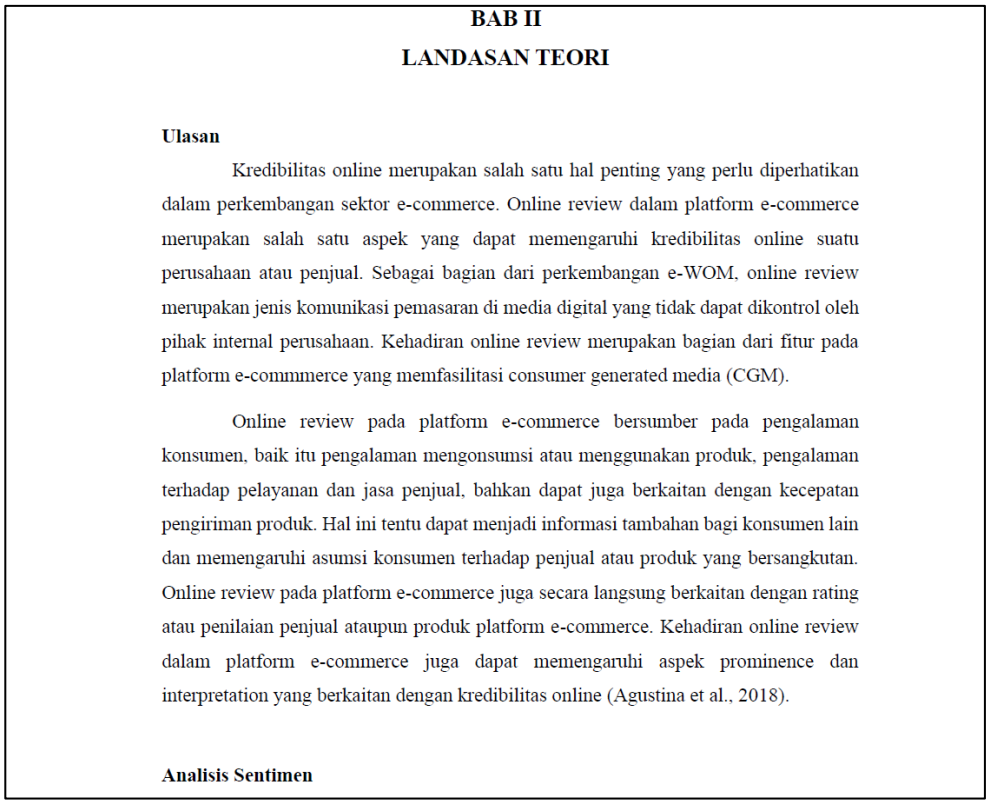
Ulasan adalah salah satu indikator yang digunakan untuk memberikan penilaian kredibilitas oleh pembeli terhadap penjual pada sebuah *e-commerce*. Ulasan yang diberikan pembeli sangat memengaruhi penilaian penjual di *e-commerce*. Ulasan juga merupakan bentuk perkembangan dari *electronic word of mouth* (e-WOM). E-WOM adalah bentuk komunikasi pemasaran di dunia maya yang tidak dapat dikendalikan oleh penjual maupun perusahaan. Karena informasi yang terdapat di dalam ulasan pembeli merupakan informasi yang tidak dikelola oleh pihak internal perusahaan.

Ulasan yang diberikan oleh pengguna di platform *e-commerce* dapat menjadi informasi bagi pengguna lainnnya. Ulasan biasanya berasal dari pengalaman pembeli dalam berbelanja, ulasan tersebut dapat berupa produk yang dijual, respon penjual, kecepatan pengiriman, dan sebagainya (Agustina et al., 2018).

2.2. Analisis Sentimen

Analisis Sentimen atau juga disebut *Opinion Mining*, merupakan bidang *Natural Language Processing* (NLP) yang bertanggung jawab untuk mengekstrak polaritas subjektif pengguna mengenai topik tertentu. Polaritas subjektif mencakup sikap, penilaian, dan emosi pengguna sehubungan dengan banyak aspek masyarakat, seperti produk, organisasi, acara, atau layanan. Oleh karena itu, analisis sentimen telah

Gambar 3.1 Contoh Dokumen Uji



BAB II

LANDASAN TEORI

Ulasan

Kredibilitas online merupakan salah satu hal penting yang perlu diperhatikan dalam perkembangan sektor e-commerce. Online review dalam platform e-commerce merupakan salah satu aspek yang dapat memengaruhi kredibilitas online suatu perusahaan atau penjual. Sebagai bagian dari perkembangan e-WOM, online review merupakan jenis komunikasi pemasaran di media digital yang tidak dapat dikontrol oleh pihak internal perusahaan. Kehadiran online review merupakan bagian dari fitur pada platform e-commmerce yang memfasilitasi consumer generated media (CGM).

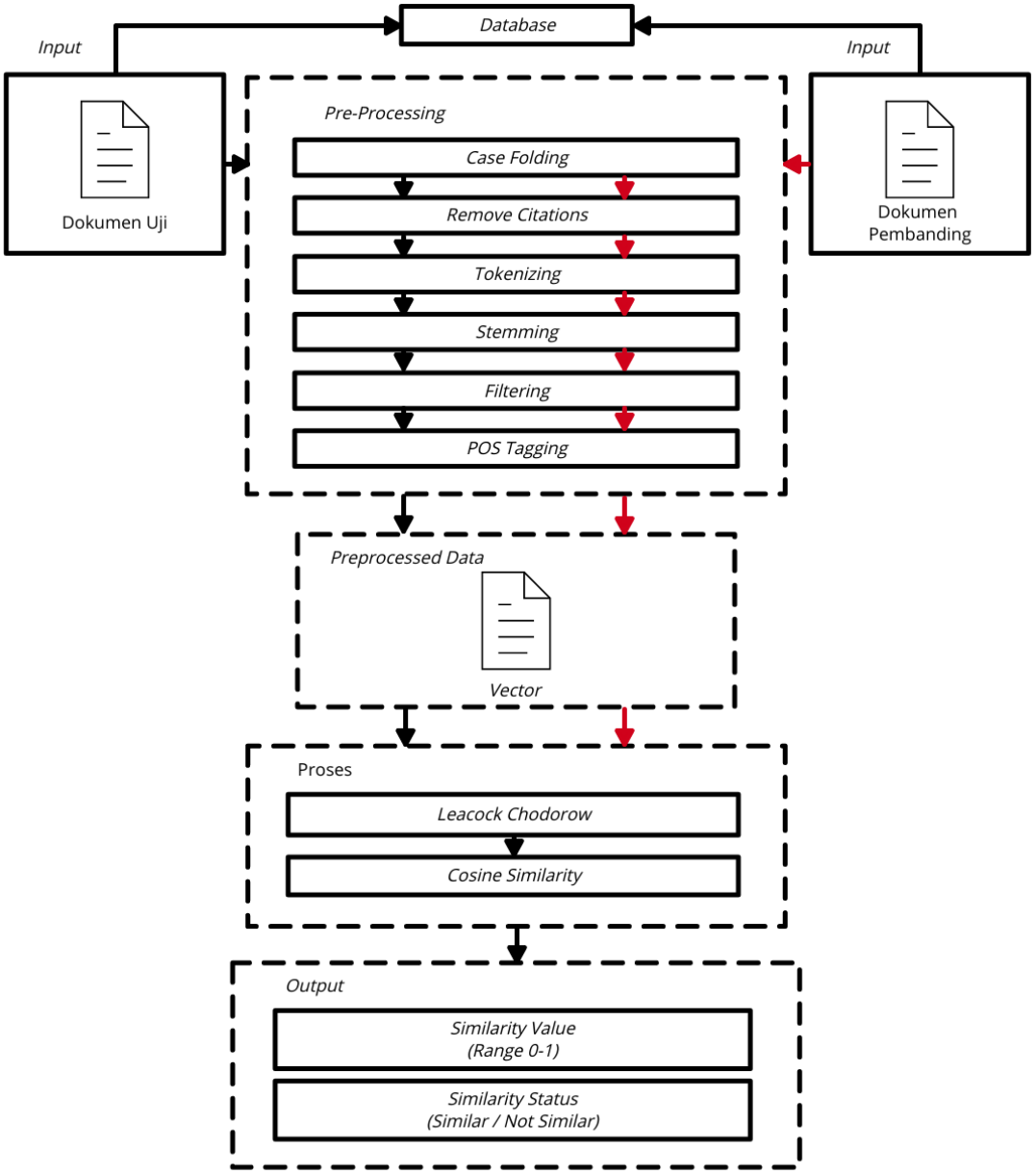
Online review pada platform e-commerce bersumber pada pengalaman konsumen, baik itu pengalaman mengonsumsi atau menggunakan produk, pengalaman terhadap pelayanan dan jasa penjual, bahkan dapat juga berkaitan dengan kecepatan pengiriman produk. Hal ini tentu dapat menjadi informasi tambahan bagi konsumen lain dan memengaruhi asumsi konsumen terhadap penjual atau produk yang bersangkutan. Online review pada platform e-commerce juga secara langsung berkaitan dengan rating atau penilaian penjual ataupun produk platform e-commerce. Kehadiran online review dalam platform e-commerce juga dapat memengaruhi aspek prominence dan interpretation yang berkaitan dengan kredibilitas online (Agustina et al., 2018).

Analisis Sentimen

Gambar 3.2 Contoh Dokumen Pembanding

3.2. Arsitektur Umum

Penelitian ini memerlukan pelatihan dan juga pengujian algoritma agar sistem mampu mengetahui dan mengindikasikan nilai kemiripan teks. Tahapan *Pre-Processing* akan dilakukan terhadap data yang sudah terkumpul, mulai dari *Case Folding*, *Remove Citations*, *Tokenizing*, *Filtering*, *Stemming* untuk dokumen dalam bahasa Indonesia dan *Part-of-Speech Tag*. Setelah itu, data hasil *Pre-Process* akan memasuki tahap proses dimana kemiripan teks akan dihitung dengan algoritma *Leacock Chodorow* dan *Cosine Similarity*. Arsitektur umum dari sistem dalam penelitian yang dilakukan digambarkan ditunjukkan pada Gambar 3.3.



Gambar 3.3 Arsitektur Umum Sistem

3.2.1. *Input*

Input dalam sistem terbagi menjadi dua, yaitu dokumen uji (dokumen yang ingin dilihat tingkat kemiripannya) serta dokumen pembanding (dokumen referensi dari dokumen uji). Data yang digunakan sebagai dokumen uji bersal dari dokumen skripsi milik mahasiswa pelajar Jurusan Teknologi Informasi Fakultas Teknologi Informasi dan Ilmu Komputer USU, dengan kriteria pemilihan dokumen acak dari beberapa angkatan yang berbeda (2018, 2017, 2016) yang diperoleh melalui *website* repositori USU.

Repositori USU sendiri merupakan sebuah *website* yang dikelola oleh Universitas Sumatera Utara dengan tujuan sebagai repositori penyimpanan berkas penelitian dan juga karya ilmiah yang telah diteliti oleh mahasiswa USU, dengan alamat https://repositori.usu.ac.id. Data yang digunakan untuk dokumen pembanding diperoleh melalui *browsing* karya ilmiah maupun jurnal dengan memperhatikan bagian sitasi pada paragraf dokumen uji serta dokumen referensi penelitian dari. daftar isi yang terdapat dalam dokumen uji.

3.2.2. *Pre-Processing*

Preprocessing merupakan serangkaian tahapan untuk menyeleksi data yang akan dipakai agar lebih terstruktur sehingga mempermudah penelitian. Adapun tahapan pada *preprocessing* terdiri atas:

3.2.2.1. *Case Folding*

Pada *Case folding*, kata yang terdapat dalam dokumen akan diubah kedalam bentuk *uppercase* atau *lowercase*. Proses juga melakukan perubahan tanda kurung menjadi tanda kurung siku, hal ini agar memudahkan proses pengambilan sitasi. Contoh *Case folding* diterapkan seperti pada Tabel 3.1

Penggalan Kode *Case Folding*

```
# Ubah teks paragraf ke huruf kecil
paragraf = paragraf.lower()
```

Penggalan kode akan mengubah huruf pada teks dalam variabel ‘*Paragraph*’ menjadi huruf kecil. Variabel memiliki fungsi untuk menampung teks yang sudah diubah ke huruf kecil.

Tabel 3.1 Contoh penerapan *Case Folding*

Sebelum *Case Folding*	**Setelah *Case Folding***
Komputer telah menjadi bagian integral dari kehidupan moderen, memfasilitasi pekerjaan, dan interaksi sosial (Aldrich, *et al*. 2023).	komputer telah menjadi bagian integral dari kehidupan moderen, memfasilitasi pekerjaan, dan interaksi social (aldrich, *et al*. 2023)
*Computer have become an integral part of modern life, facilitating work, and social interacti*on (Aldrich, *et al*. 2023).	*computer have become an integral part of modern life, facilitating work, and social interaction* (aldrich, *et al*. 2023).

Tabel 3.1 menunjukkan kata yang terdapat dalam dokumen diubah kedalam bentuk *uppercase* atau *lowercase*. Setelah proses *case folding* selesai, teks yang telah dibuah bentuknya akan masuk pada tahap selanjutnya yaitu *remove citation*.

3.2.2.2. *Remove Citation*

Pada *remove citation*, setelah sebelumnya dokumen uji dan dokumen pemanding telah melalui tahap *Case folding*, dokumen uji akan melewati tahapan penghapusan sitasi sehingga dokumen menjadi lebih sederhana untuk diproses. Sitasi yang akan dihapus adalah sitasi terdaftar berdasarkan buku pedoman penulisan skripsi oleh Sitompul *et al* (2014) yaitu Penulis (Tahun), (Penulis, Tahun), Penulis & Penulis (Tahun), Penulis *et al*. (Tahun) serta (Penulis, *et al*. Tahun). Contoh *remove citation* diterapkan seperti pada Tabel 3.2.

Penggalan Kode *Remove Citation*

```
# Import module
import re

# Hapus berbagai format kutipan: Author (Year), author (Year), (author, Year), author & author (Year), author et al. (Year), (author, et al. Year)
paragraf = re.sub(r'\b[A-Za-z]+(?:\s*(?:&|et\s*al\.)\s*[A-Za-z]+)?\s*(?:\(\d{4}\)|\(\d{4},?\s*et\s*al\.\))', '', paragraf)

# Hapus tanda kurung dan segala yang berada di dalamnya
paragraf = re.sub(r'\([^)]\)|\[[^\]]\]', '', paragraf)
```

Penggalan kode akan melakukan penghapusan terhadap sitasi dengan kriteria: Penulis (Tahun), (Penulis, Tahun), Penulis & Penulis (Tahun), Penulis *et al*. (Tahun) serta (Penulis, *et al*. Tahun), serta seluruh sitasi yang terdapat didalam kurung siku.

Tabel 3.2 Contoh penerapan Remove Citation

Sebelum *Remove Citation*	**Setelah *Remove Citation***
komputer telah menjadi bagian integral dari kehidupan moderen, memfasilitasi pekerjaan, dan interaksi social (aldrich, *et al*. 2023)	komputer telah menjadi bagian integral dari kehidupan moderen, memfasilitasi pekerjaan, dan interaksi sosial
computer have become an integral part of modern life, facilitating work, and social interaction (aldrich, *et al*. 2023)	*computer have become an integral part of modern life, facilitating work, and social interaction*

Tabel 3.2 menunjukkan pembuangan sitasi “(aldrich, *et al*. 2013)” dari dokumen uji serta dokumen pembanding. Setelah proses *remove citation* selesai, teks yang telah dibuang sitasinya akan masuk pada tahap selanjutnya yaitu *tokenization*.

3.2.2.3. *Tokenization*

Pada proses Tokenisasi, dilakukan proses pemetaan kalimat menjadi kata (Guo, 1997). Hal ini membantu dalam proses penelitian dikarenakan sebuah karya ilmiah memiliki lebih dari satu paragraf ataupun kalimat. Tahapan *Tokenizing* akan memudahkan proses karena kalimat yang terdapat pada paragraf akan dipisahkan menjadi kata. Contoh *Tokenizing* diterapkan seperti pada Tabel 3.3.

Penggalan Kode *Tokenization*

```
# Import library nltk
import nltk
from nltk.tokenize import word_tokenize
# Tokenisasi paragraf menjadi kata-kata
paragraf = "teks paragraf yang ingin di-tokenisasi"
kata_kata = word_tokenize(paragraf)
```

Penggalan kode akan melakukan proses pemisahan teks dalam variabel ‘paragraf’ kedalam bentuk vektor kata yang terpisah kedalam variabel ‘kata_kata’.

Tabel 3.3 Contoh penerapan *Tokenizing*

Sebelum *Tokenizing*	Setelah *Tokenizing*
komputer telah menjadi bagian integral dari kehidupan moderen, memfasilitasi pekerjaan, dan interaksi sosial	['komputer', 'telah', 'menjadi', 'bagian', 'integral', 'dari', 'kehidupan', 'modern', 'memfasilitasi', 'pekerjaan', 'dan', 'interaksi', 'sosial']
computer have become an integral part of modern life, facilitating work, and social interaction	[*'computer', 'have', 'become', 'an', 'integral', 'part', 'of', 'modern', 'life', 'facilitating', 'work', 'and', 'social', 'interaction'*]

Tabel 3.3 menunjukkan mengubah dokumen uji serta dokumen pembanding dengan melakukan pemetaan kalimat menjadi kata. Setelah proses *tokenizing* selesai, teks hasil pemetaan akan masuk pada tahap selanjutnya yaitu *stemming*.

3.2.2.4. *Stemming*

Pada proses *Stemming*, kata yang ada dalam teks akan dilakukan proses perubahan kedalam bentuk kata dasar agar kata dengan bahasa Indonesia dapat dideteksi dengan baik. Contoh *Stemming* diterapkan seperti pada Tabel 3.4.

Penggalan Kode *Stemming*

```
# Impor StemmerFactory dari perpustakaan Sastrawi
import StemmerFactory

# Inisialisasi stemmer
stemmer_factory = StemmerFactory()
stemmer = stemmer_factory.create_stemmer()

# Asumsikan bahwa 'words' adalah daftar kata yang ingin di-stem
words = [daftar kata]

# Lakukan stemming pada setiap kata
stemmed_words = []

untuk setiap kata dalam words:
    kata_stem = stemmer.stem(kata)
    tambahkan kata_stem ke dalam stemmed_words

# Sekarang 'stemmed_words' berisi versi kata yang telah di-stem
```

Penggalan kode akan melakukan proses perubahan kata kedalam bentuk dasar dengan tujuan meningkatkan dideteksi bahasa dari sistem.

Tabel 3.4 Contoh penerapan *Stemming*

Sebelum *Stemming*	Setelah *Stemming*
['komputer', 'telah', 'menjadi', 'bagian', 'integral', 'dari', 'kehidupan', 'modern', 'memfasilitasi', 'pekerjaan', 'dan', 'interaksi', 'sosial']	['komputer', 'telah', 'jadi', 'bagi', 'integral', 'dari', 'hidup', 'modern', 'fasilitas', 'kerja', 'dan', 'interaksi', 'sosial']

Tabel 3.4 menunjukkan penerapan *stemming* dimana kata akan diubah menjadi bentuk kata dasar (*root word*) hal ini dibutuhkan agar *WordNet* dapat mendeteksi kata bahasa indonesia dengan baik. Pada penelitian Fatonah et al (2020) didapat kesimpulan jika tahapan stemming membuat proses program menjadi lebih cepat.

Pada tahap ini penulis memilih menggunakan stemming Sastrawi karena pada penelitian Fatonah et al (2020) algoritma ini menghasilkan proses paling cepat jika dibandingkan algoritma Nazief-Andriani, dan algoritma Sastrawi memiliki tingkat akurasi yang tinggi. Tahapan proses *stemming* pada algoritma sastrawi adalah sebagai berikut :

1. Pengecekan kata per kata dilakukan. jika kata tersedia didalam kamus maka pengecekan berhenti, jika tidak maka pengecekan akan berlanjut.
2. Jika ditemukan kata dengan akhiran (kah, lah, mu, nya, ku) maka akhiran akan dihapus. Jika kata tersebut adalah adalah particles (tah, pun, lah, kah,) langkah akan diulang untuk menentukan jika terdapat kata ganti (mu, ku , nya), maka akan dihapus.
3. Jika ditemukan kata dengan akhiran (i, an, kan), algoritma akan menghentikan proses, jika kata dengan akhiran tersebut tidak ditemukan maka akan masuk ke tahap berikutnya.
4. Jika akhiran (an) pada kata yang dihapus menghasilkan (k) sebagai huruf terakhir dari kata, akhiran (k) akan dilakukan penghapusan, dipastikan kembali di dalam kamus ketersediaan kata, apabila tidak tersedia kata dengan akhiran tersebut akan dikembalikan seperti semula.

5. Jika ditemukan kata yang dihapus pada tahap sebelumnya, maka dilakukan pemeriksaan awalan. Jika ditemukan awalan dalam kata maka awalan akan dihapus, jika kata tidak tidak mengandung awalan dan akhiran yang tidak di ijinkan maka *stemming* dilakukan. Penjelasan lebih lanjut seperti pada Tabel 3.5, 3.6, dan 3.7.

Tabel 3.5 Contoh proses *Stemming* (ku, nya, mu, lah, kah)

Kata	Akhiran	Hasil *Stemming*
temanku	ku	teman
kelihatannya	nya	kelihatan
salahmu	mu	salah
besarkah	kah	besar
sudahlah	lah	sudah

Tabel 3.5 menunjukkan contoh hasil *stemming* terhadap beberapa kata yang memiliki akhiran yang terdaftar dalam *stemmer* Sastrawi.

Tabel 3.6 Contoh proses *Stemming* tanpa penghapusan akhiran (i, kan, an)

Kata	Akhiran	Hasil *Stemming*
Padi	i	padi
Pekan	kan	pekan
Iman	an	iman

Tabel 3.6 menunjukkan contoh hasil *stemming* terhadap beberapa kata yang memiliki akhiran kata yang tidak akan dilakukan penghapusan.

Tabel 3.7 Contoh proses *Stemming* akhiran (kan)

Kata	Akhiran	Hasil *Stemming*
Tahankan	kan	tahan
Pesanan	an	pesan
Sungkan	kan	Sungkan

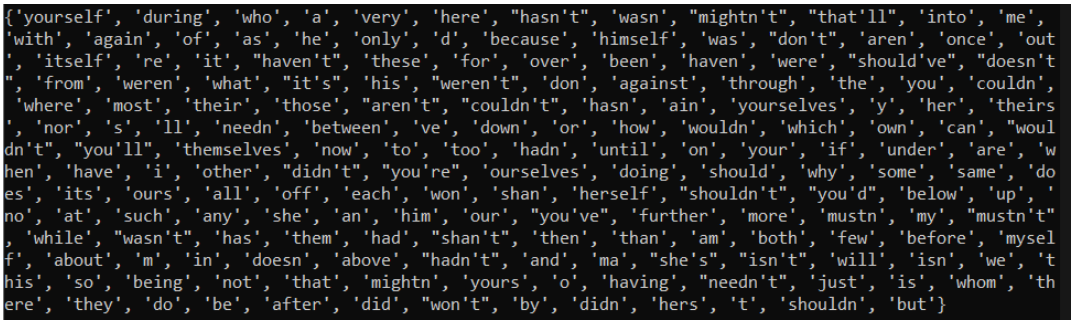
Tabel 3.7 menunjukkan contoh hasil *stemming* terhadap kata sesuai tahap 4 serta tahap 5 proses *stemming* Sastrawi. Setelah *stemming* dilakukan, *preprocessing* akan masuk pada tahap *filtering*.

Tahap *stemming* tidak dilakukan terhadap teks bahasa Inggris atas dasar penghapusan akhiran terhadap banyak kata bahasa Inggris yang seharusnya tidak perlu dihapus karena akhiran tersebut termasuk salah satu akhiran yang dihapus dalam *stemming* bahasa Inggris. Beberapa akhiran kata yang hilang menyebabkan hilangnya makna dari kata secara menyeluruh. Teks yang kehilangan akhiran serta makna akan mengurangi akurasi pengujian, dan dengan demikian *stemming* bahasa Inggris tidak dilakukan demi menjaga akurasi data dalam pengujian.

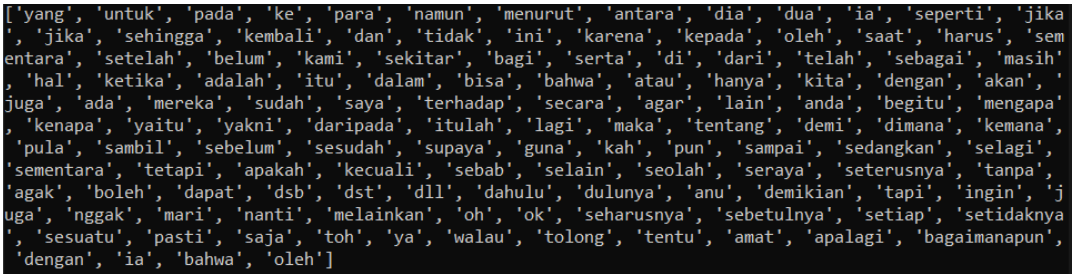
3.2.2.5. *Filtering*

Pada proses *Filtering*, proses penyaringan teks yang dianggap kurang penting, misalnya kata “untuk”, “pada”, “ke”, serta “yang” akan dilakukan. *stop list* atau *word list* merupakan beberapa algoritma yang dapat digunakan dalam proses ini. Proses *Filtering* dibutuhkan untuk mempercepat proses perhitungan. Dalam tahanp penelitian ini penulis akan menggunakan *library python Sastrawi* serta metode *Stoplist* (*stopword removal*) dalam proses *filtering*.

Pada tahap ini akan penulis memanfaatkan library Sastrawi untuk melakukan stopword removal pada teks dengan bahasa Indonesia dan menggunakan library nltk pada teks dengan bahasa Inggris. Daftar stopword pada library Sastrawi seperti pada Gambar 3.4 serta Gambar 3.5.



Gambar 3.4 Daftar *Stopword Library NLTK*



Gambar 3.5 Daftar *Stopword Library Sastrawi*

Gambar 3.4 serta Gambar 3.5 menampilkan daftar *stopword* yang dibuang dalam proses *filtering* sistem. Setelah dokumen melalui tahap *tokenizing*, tahap yang dilakukan dalam penelitian adalah tahap *filtering* dengan ketentuan penggunaan daftar *stopword* sesuai bahasa yang digunakan didalam dokumen. Contoh penerapan *Filtering* seperti pada Tabel 3.8.

Tabel 3.8 Contoh penerapan *Filtering*

Sebelum *Filtering*	**Setelah *Filtering***
['komputer', 'telah', 'jadi', 'bagi', 'integral', 'dari', 'hidup', 'modern', 'fasilitas', 'kerja', 'dan', 'interaksi', 'sosial']	['komputer', 'jadi', 'integral', 'hidup', 'modern', 'fasilitas', 'kerja', 'interaksi', 'sosial']
[*'computer', 'have', 'become', 'an', 'integral', 'part', 'of', 'modern', 'life', 'facilitating', 'work', 'and', 'social', 'interaction'*]	[*'computer', 'become', 'integral', 'part', 'modern', 'life', 'facilitating', 'work', 'social', 'interaction'*]

Tabel 3.8 menunjukkan contoh penerapan *filter* terhadap paragraf uji serta paragraf pembanding yang diberikan. Setelah proses filtering selesai, teks yang telah disaring akan masuk pada tahap selanjutnya yaitu tahap *POS tagging.*

3.2.2.6. *POS Tagging*

Pada tahap *POS tagging*, dilakukan langkah pemberian tanda atau label kata didalam sebuah kalimat dalam bentuk *POS* atau *tag* sesuai kelas kata dengan pembagian berupa kata kerja, kata keterangan, kata sifat dan lainnya. Tahap ini digunakan dikarenakan *WordNet* yang berguna dalam melihat kedekatan kata dari segi holonim, meronim, hipernim, hiponim, sinonim, serta antonym. *WordNet* memiliki *synset* (*synonim set*) yang dikelompokkan ke dalam kelas kata seperti *verb, adverb, adjective*, serta *noun*. *POS tagging* juga mempercepat proses *word similarity* pada algoritma *Leacock Chodorow*. Pada tahap ini, penelitian akan menggunakan dua *library* berbeda, pertama *libray crf tagger* dengan *dataset* yang dikembangkan oleh Dinakaramani *et al* (2010) yang mengandung 10.000 kata untuk *POS Tag* bahasa Indonesia, serta *library NLTK* untuk *POS Tag* berbahasa *Inggris*. Daftar tag pada *POS Tag* bahasa Indonesia seperti pada Tabel 3.9 Dinakaramani et al (2010), dan daftar *tag* pada *POS Tag* bahasa Inggris seperti Tabel 3.10.

Tabel 3.9 *POS Tag* Bahasa Indonesia

Tag	Deskripsi
Z	Tanda baca
X	Kategori tidak diketahui
WH	Pronominal penanya
VB	Kata kerja atau verba
UH	Interjeksi
SYM	Simbol
SC	Konjungtor subordinative
RP	Partikel
RB	Kata keterangan atau adverbial
PRP	Pronomina persona
PR	Pronomina penunjuk

Tabel 3.9 *POS Tag* Bahasa Indonesia (Lanjutan)

Tag	Deskripsi
OD	Numeralia ordinal
NNP	Proper Noun
NND	Penggolongan atau nomina ukuran
NN	Kata benda atau nomina
NEG	Kata ingkar
MD	Verba modal dan verba bantu
JJ	Kata sifat atau adjektiva
IN	Kata depan
FW	Bahasa asing
DT	Artikel
CD	Numeralia cardinal
CC	Konjungtor koordinatif

Tabel 3.9 menunjukkan jenis *POS tag* bahasa Indonesia yang terdapat dalam *dataset* yang dipakai dalam penelitian berupa *dataset* yang dikembangkan oleh Dinakaramani *et al* (2010).

Tabel 3.10 *POS Tag* Bahasa Inggris

Tag	Deskripsi
WRB	Wh-adverb
WP$	Possessive wh-pronoun
WP	Wh-pronoun
WDT	Wh-determiner
VBZ	Verb, 3^{rd} person singular present
VBP	Verb, non- 3^{rd} person singular present
VBN	Verb, past participle
VBG	Verb, gerund or present participle
VBD	Verb, past tense
VB	Verb, base form

Tabel 3.10 *POS Tag* Bahasa Inggris (Lanjutan)

Tag	Deskripsi
UH	Interjection
TO	To
SYM	Symbol
RP	Particle
RBS	Adverb, superlative
RBR	Adverb, comparative
RB	Adverb
PRP$	Possessive pronoun
PRP	Personal pronoun
POS	Possessive ending
PDT	Predeterminer
NNS	Noun, plural
NNPS	Proper noun, plural
NNP	Proper noun, singular
NN	Noun, singular or mass
MD	Modal
LS	List item marker
JJS	Adjective, superlative
JJR	Adjective, comparative
JJ	Adjective
IN	Subordinating conjunction
FW	Foreign word
EX	Existential there
DT	Determiner
CD	Cardinal number
CC	Coordinating conjunction

Tabel 3.10 menunjukkan jenis *POS tag* bahasa Inggris yang terdapat dalam *library NLTK* yang digunakan dalam penelitian

Contoh pengaplikasian *POS Tag* terhadap teks berbahasa Indonesia seperti pada Tabel 3.11 serta pengaplikasian *POS Tag* terhadap teks berbahasa Inggris seperti pada Tabel 3.12.

Tabel 3.11 Contoh *POS Tagging* Bahasa Indonesia

Kata Masukan	*POS Tag*
Computer	NN
Jadi	VB
Integral	NN
Hidup	NN
Modern	JJ
Fasilitas	NN
Kerja	NN
Interaksi	NN
Sosial	JJ

Tabel 3.11 menunjukkan pemberian *POS tag* bahasa Indonesia terhadap seluruh kata yang dimasukkan dalam paragraf uji.

Tabel 3.12 Contoh *POS Tagging* Bahasa Inggris

Kata Masukan	*POS Tag*
Computer	NN
Become	VB
Integral	JJ
Part	NN
Modern	JJ
Life	NN
Facilitating	VBG
Work	NN
Social	JJ
Interaction	NN

Tabel 3.12 menunjukkan pemberian *POS tag* bahasa Inggris terhadap seluruh kata yang dimasukkan dalam paragraf pembanding. Setelah tahap *POS tagging* selesai selanjutnya masuk ke tahap proses, yaitu tahap perhitungan kemiripan paragraf dengan menggunakan *Leacock-Chodorow* dan *cosine similarity*.

3.2.3. *Pre-Processed Document*

Setelah melewati tahapan *preprocessing* tahapan selanjutnya adalah menyimpan dokumen uji dan dokumen pembanding yang telah dilakukan *preprocess* kedalam sebuah direktori. Dokumen ini disimpan untuk melihat hasil dari dokumen uji dan dokumen pembanding yang digunakan sebelum masuk pada tahap proses.

3.2.4. *Proses*

Pada tahap proses akan dilakukan metode perhitungan yaitu menggunakan algoritma *Leacock Chodorow Similarity*, dan *cosine similarity* dengan acuan pada kamus *WordNet* bahasa Indonesia dan bahasa Inggris.

3.2.4.1. *Leacock Chodorow Similarity*

Pada tahap ini akan dilakukan perhitungan dengan menggunakan metode *Leacock Chodorow Similarity* (LCH) untuk menemukan kemiripan pada teks antara dokumen uji dan dokumen pembanding. Pada umumnya, algoritma *Cosine Similarity* menghitung kemiripan dengan pemakaian *term frequency* (tf) kata terhadap data uji dan data pembanding. Algoritma ini melakukan perbandingan kemiripan kata dari keberadaan kata yang diuji pada kata pembanding dan akan mengembalikan nilai 1 apabila terdapat kata yang sama, serta nilai 0 apabila tidak ada. *Cosine similarity* tidak memperhatikan hubungan semantik antar kata yang diuji. Algoritma *Leacock Chodorow* dapat menanggulangi masalah ini karena *Leacock Chodorow* mengukur derajat keterkaitan antar teks. Agar hasil kemiripan tidak mengalami *over value*, *threshold* digunakan terhadap nilai kemiripan kata yang didapat. Apabila nilai *word Similarity* yang dihasilkan lebih kecil dibanding nilai *threshold* yang dipakai, kata akan dianggap memiliki nilai kemiripan 0.

Pada penelitian ini, nilai threshold yang dipakai adalah 0.7, yang dijelaskan sebagai nilai *threshold* yang digunakan dalam penelitian yang dilakukan oleh Wicaksana & Hakim (2006). Contoh *word similarity* dengan menggunakan *cosine similarity* konvensional seperti pada Tabel 3.13, sedangkan *Leacock Chodorow similarity* pada Tabel 3.14 dan 3.15.

Tabel 3.13 Contoh *Word Similarity* dengan *Cosine Similarity*

Kata Uji	Kata Pembanding	*Word Similarity*
komputer	mesin	0
jadi	jadi	1.00
integral	integral	1.00
hidup	nyawa	0
fasilitas	akomodasi	0
kerja	tugas	0
interaksi	tali	0
sosial	masyarakat	0

Tabel 3.13 menunjukkan perhitungan *word similarity* menggunakan *cosine similarity* dan mendapatkan hasil yang kurang baik dalam perhitungan kemiripan kata karena *cosine similarity* melakukan perhitungan kemiripan kata dari keberadaan kata dan akan mengembalikan nilai 1 apabila kata pada dokumen uji ada dalam dokumen pembanding, serta nilai 0 apabila kata tidak ada

Perhitungan *word similarity* menggunakan *Leacock Chodorow similarity* dapat dilakukan menggunakan persamaan 2.1. Contoh perhitungan *word similarity* antara kata ‘komputer’ dan ‘mesin’ menggunakan *Leacock Chodorow similarity* dapat dilakukan dengan ketentuan:

Shortest path pada *WordNet* untuk kata ‘komputer’ dan kata ‘mesin’ = 8

Max Depth pada *WordNet* untuk kata ‘komputer’ = 8

Max Depth pada *WordNet* untuk kata ‘mesin’ = 11

Didapatkan:

spath (‘komputer’,‘mesin’) = 8

Depth (diambil dari kata ‘mesin’) = 11

$$Word\ Similarity = -\log \frac{8}{(2 * 11)}$$

$$Word\ Similarity = -\log \frac{8}{22}$$

$$Word\ Similarity = -\log\ 0.36363636363636365$$

$$Word\ Similarity = \mathbf{1.0116009116784799}$$

Nilai *word similarity* akan dinormalisasi terlebih dahulu dengan tujuan memudahkan pembandingan terhadap nilai *threshold*. Normalisasi dilakukan dengan membagikan nilai *similarity* terhadap nilai *similarity* maksimum dalam *WordNet*. Nilai *similarity* maksimum dalam *WordNet* didapatkan dengan menggunakan persamaan 2.1 dengan ketentuan:

Shortest path pada *WordNet* = 1

Max Depth pada *WordNet* = 19

$$Max\ WordNet\ Similarity = -\log \frac{1}{(2 * 19)}$$

$$Max\ WordNet\ Similarity = -\log \frac{1}{38}$$

$$Max\ WordNet\ Similarity = \mathbf{3.6375861597263857}$$

Setelah didapatkan nilai *similarity* maksimum *WordNet*, akan dilakukan pembagian nilai *similarity* antara kata ‘komputer’ dan ‘mesin’ terhadap nilai *similarity* maksimum *WordNet*, dan didapatkan persamaan 3.1:

$$Normalized\ Similarity = \frac{Word\ Similarity}{Max\ WordNet\ Similarity} \quad (3.1)$$

$$Normalized\ Similarity = \frac{1.0116009116784799}{3.6375861597263857}$$

$$Normalized\ Similarity = \mathbf{0.27809675627}$$

Dengan demikian perhitungan *similarity* antara kata ‘komputer’ dan ‘mesin’ menggunakan *Leacock Chodorow* menghasilkan nilai **0.2781**.

Tabel 3.14 Contoh *Word Similarity* dengan *Leacock-Chodorow Similarity*

Kata Uji	Kata Pembanding	*POS Tag*	*Word Similarity*
komputer	mesin	*Noun, Noun*	1.0116
jadi	jadi	*Adjective, Adjective*	3.6376
integral	integral	*Adjective, Adjective*	3.6376
hidup	nyawa	*Verb, Verb*	1.2527
fasilitas	akomodasi	*Noun, Noun*	1.1632
interaksi	tali	*Noun, Noun*	0.9808
sosial	masyarakat	*Adjective, Noun*	*None*

Tabel 3.14 menunjukkan bahwa melakukan perhitungan *word similarity* dengan menggunakan *Leacock-Chodorow* menghasilkan bentuk nilai berbeda dari *cosine similarity* dan perlu dinormalisasi sebelum dapat masuk pada tahap selanjutnya. Perhitungan *Leacock Chodorow* akan menghasilkan nilai *None* jika *POS Tag* kata yang dibandingkan berbeda, hal ini terjadi karena kata yang dibandingkan tidak memiliki *shortest path* pada taksonomi *WordNet*

Tabel 3.15 Contoh *Word Similarity* setelah normalisasi

Kata Uji	Kata Pembanding	*POS Tag*	*Normalized Similarity*
komputer	mesin	*Noun, Noun*	0.2781
jadi	jadi	*Adjective, Adjective*	1.0
integral	integral	*Adjective, Adjective*	1.0
hidup	nyawa	*Verb, Verb*	0.3444
fasilitas	akomodasi	*Noun, Noun*	0.3198
interaksi	tali	*Noun, Noun*	0.2696
sosial	masyarakat	*Adjective, Noun*	*None*

Tabel 3.15 menunjukkan bahwa setelah normalisasi *word similarity* dilakukan, didapatkan bahwa *Leacock Chodorow* memberikan hasil yang lebih baik dari *cosine similarity* karena *Leacock Chodorow* mampu mengukur tingkat semantik kata. Setelah nilai kemiripan menggunakan *Leacock Chodorow* didapatkan, nilai kemiripan akan masuk tahap perhitungan menggunakan *cosine similarity* untuk mendapatkan nilai kemiripan akhir.

3.2.4.2. *Cosine Similarity*

Pada tahap ini akan perhitungan dengan menggunakan metode *cosine similarity* dilakukan dengan tujian menemukan kemiripan pada paragraf antara dokumen uji dan dokumen pembanding. Algoritma *cosine similarity* hanya dapat menghitung kemiripan paragraph apabila dimensi vektor paragraf uji dan dimensi vektor paragraf pembanding sama, dan dengan demikian paragraf uji dan paragraf pembanding perlu digabung serta dihilangkan kata duplikat didalamnya (metode *cosine* konvensional). Contoh perhitungan kemiripan paragraf antara dokumen uji dan dokumen pembanding seperti:

Paragraf Uji: Komputer telah menjadi bagian integral dari kehidupan moderen, memfasilitasi pekerjaan, dan interaksi sosial (aldrich, *et al.* 2023)

Paragraf Banding: *Computer have become an integral part of modern life, facilitating work, and social interaction* (aldrich, *et al.* 2023)

Paragraf Uji PreProcessing: [('komputer', 'NN'), ('jadi', 'VB'), ('integral', 'NN'), ('hidup', 'NN'), ('modern', 'JJ'), ('fasilitas', 'NN'), ('kerja', 'NN'), ('interaksi', 'NN'), ('sosial', 'JJ')]

Paragraf Banding PreProcessing: [*('computer', 'NN'), ('become', 'VB'), ('integral', 'JJ'), ('part', 'NN'), ('modern', 'JJ'), ('life', 'NN'), ('facilitating', 'VBG'), ('work', 'NN'), ('social', 'JJ'), ('interaction', 'NN')*]

Gabungan Paragraf Uji dan Paragraf Banding: [('komputer', 'NN'), ('jadi', 'VB'), ('integral', 'NN'), ('hidup', 'NN'), ('modern', 'JJ'), ('fasilitas', 'NN'), ('kerja', 'NN'), ('interaksi', 'NN'), ('sosial', 'JJ'), *('computer', 'NN'), ('become', 'VB'), ('integral', 'JJ'), ('part', 'NN'), ('modern', 'JJ'), ('life', 'NN'), ('facilitating', 'VBG'), ('work', 'NN'), ('social', 'JJ'), ('interaction', 'NN')*]

Selanjutnya dilakukan pencarian kemiripan kata - perkata dengan menggunakan algoritma Leacock Chodorow dengan menggunakan persamaan 2.1. Untuk membentuk vektor paragraf uji ambil nilai maksimal dari setiap cell dan di dapatkan vektor paragraf uji. Contoh dari penerapan langkah seperti pada Gambar 3.6 dan 3.7.

	komputer	jadi	integral	hidup	modern	fasilitas	kerja	interaksi	sosial
komputer	3,6376	0,9295	0,9295	2,2513	-	1,8458	1,5582	1,1530	1,0726
jadi	0,9295	3,6376	0,8044	3,2580	-	0,9295	2,5650	2,1595	0,9295
integral	0,9295	0,8044	3,6376	1,3350	-	0,9295	1,1530	1,1530	0,9295
hidup	2,2513	3,2580	1,3350	3,6376	-	2,0282	3,2580	2,1595	1,3350
modern	-	-	-	-	0,6932	-	-	-	-
fasilitas	1,8458	0,9295	0,9295	2,0282	-	3,6376	2,0282	1,1527	1,0726
kerja	1,5582	2,5650	1,1527	3,2580	-	2,0282	3,6376	2,0282	1,3350
interaksi	1,1527	2,1595	1,1527	2,1595	-	1,1527	2,0282	3,6376	1,1527
sosial	1,0726	0,9295	0,9295	1,3350	-	1,0726	1,3350	1,1527	3,6376
TOTAL	**3,6376**	**3,6376**	**3,6376**	**3,6376**	**0,6932**	**3,6376**	**3,6376**	**3,6376**	**3,6376**

computers	become	integral	part	modern	life	facilitating	work	social	interaction
3,6376	-	0,9295	1,6917	2,2513	2,2513	-	1,5582	1,0726	1,3350
0,9295	3,2580	0,8044	2,1595	0,9985	1,1527	1,6487	2,5650	0,9295	0,9985
0,9295	-	3,6376	1,2397	0,9985	1,3350	-	1,4404	0,9295	1,1527
2,2513	3,2580	1,3350	2,1595	2,5390	3,6376	1,6487	3,2580	1,3350	2,0282
-	-	-	-	0,6932	-	-	-	-	-
1,8458	-	0,9295	2,0282	1,6917	2,0282	-	2,0282	1,0726	1,3350
1,5582	2,1595	1,1527	2,9444	1,4404	2,2513	2,5650	3,6376	1,3350	2,2513
1,1527	1,8718	1,1527	1,8718	1,2397	1,6917	1,4663	2,0282	1,1527	3,6376
1,0726	-	0,9295	1,4404	1,1527	1,3350	-	1,1527	3,6376	1,1527
3,6376	**3,2580**	**3,6376**	**2,9444**	**2,5390**	**3,6376**	**2,5650**	**3,6376**	**3,6376**	**3,6376**

Gambar 3.6 Hasil Vector Paragraf Uji

Vektor paragraf uji dibentuk dengan cara mengambil nilai maksimal dari setiap cell. Dari Gambar 3.6 dapat dilihat nilai setiap cell untuk paragraph uji = [3.6376, 3.6376, 3.6376, 3.6376, 0.6932, 3.6376, 3.6376, 3.6376, 3.6376, 3.6376, 3.2580, 3.6376, 2.9444, 2.5390, 3.6376, 2.5650, 3.6376, 3.6376, 3.6376]

Sebelum dimasukkan kedalam vektor paragraf uji, nilai maksimal terlebih dahulu dinormalisasi terhadap nilai kemiripan maksimum dari *Leacock Chodorow* dan setelah itu akan didapatkan vektor paragraf uji = [1.0, 1.0, 1.0, 1.0, 0.1906, 1.0, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 0.8095, 0.6980, 1.0, 0.7051, 1.0, 1.0, 1.0]

	komputer	jadi	integral	hidup	modern	fasilitas	kerja	interaksi	sosial
computers	3,6376	0,9295	0,9295	2,2513	-	1,8458	1,5582	1,1527	1,0726
become	-	3,2580	-	3,2580	-	-	2,1595	1,8718	-
integral	0,9295	0,8044	3,6376	1,3350	-	0,9295	1,1527	1,1527	0,9295
part	1,6917	2,1595	1,2397	2,1595	-	2,0282	2,9444	1,8718	1,4404
modern	2,2513	0,9985	0,9985	2,5390	0,6932	1,6917	1,4404	1,2397	1,1527
life	2,2513	1,1527	1,3350	3,6376	-	2,0282	2,2513	1,6917	1,3350
facilitating	-	1,6487	-	1,6487	-	-	2,5650	1,4663	-
work	1,5582	2,5650	1,4404	3,2580	-	2,0282	3,6376	2,0282	1,1527
social	1,0726	0,9295	0,9295	1,3350	-	1,0726	1,3350	1,1527	3,6376
interaction	1,3350	0,9985	1,1527	2,0282	-	1,3350	2,2513	3,6376	1,1527
TOTAL	**3,6376**	**3,2580**	**3,6376**	**3,6376**	**0,6932**	**2,0282**	**3,6376**	**3,6376**	**3,6376**

computers	become	integral	part	modern	life	facilitating	work	social	interaction
3,6376	-	0,9295	1,6917	2,2513	2,2513	-	1,5582	1,0726	1,3350
-	3,2580	-	2,1595	-	-	1,6487	2,5650	-	-
0,9295	-	3,6376	1,2397	0,9985	1,3350	-	1,4404	0,9295	1,1527
1,6917	2,1595	1,2397	3,6376	1,8458	2,0282	1,6487	2,5390	1,4404	1,8458
2,2513	-	0,9985	1,8458	3,6376	2,5390	-	1,4404	1,1527	1,4404
2,2513	-	1,3350	2,0282	2,5390	3,6376	-	1,8458	1,3350	2,0282
-	1,6487	-	1,6487	-	-	3,2580	1,8718	-	-
1,5582	2,5650	1,4404	2,5390	1,4404	1,8458	1,8718	3,6376	1,1527	2,2513
1,0726	-	0,9295	1,4404	1,1527	1,3350	-	1,1527	3,6376	1,1527
1,3350	-	1,1527	1,8458	1,4404	2,0282	-	2,2513	1,1527	3,6376
3,6376	**3,2580**	**3,6376**	**3,6376**	**3,6376**	**3,6376**	**3,2580**	**3,6376**	**3,6376**	**3,6376**

Gambar 3.7 Hasil Vector Paragraf Pembanding

Vektor paragraf pembanding juga dibentuk dari nilai maksimal setiap cell. Dari Gambar 3.7 dapat dilihat nilai setiap cell untuk paragraf pembanding = [3.6376, 3.2580, 3.6376, 3.6376, 0.6932, 2.0282, 3.6376, 3.6376, 3.6376, 3.6376, 3.2580, 3.6376, 3.6376, 3.6376, 3.6376, 3.2580, 3.6376, 3.6376, 3.6376]

Sebelum dimasukkan kedalam vektor paragraf pembanding, nilai maksimal terlebih dahulu dinormalisasi terhadap nilai kemiripan maksimum dari *Leacock Chodorow* dan setelah itu akan didapatkan vektor paragraf pembanding = [1.0, 0.8957, 1.0, 1.0, 0.1906, 0.5576, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 1.0, 1.0]

Nilai dari kedua vektor akan kemudian diberikan *threshold* untuk menghindari nilai kemiripan yang *over value*, sehingga menghasilkan vektor akhir berupa

Vektor paragraf uji = [1.0, 1.0, 1.0, 1.0, 0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 0.8095, 0, 1.0, 0.7051, 1.0, 1.0, 1.0]

Vektor paragraf pembanding = [1.0, 0.8957, 1.0, 1.0, 0, 0, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 1.0, 1.0]

Setelah nilai dari kedua vektor didapatkan, akan dilakukan perhitungan nilai kemiripan kedua vektor dengan menggunakan persamaan *cosine similarity*. Perhitungan *similarity* menggunakan *cosine similarity* dapat dilakukan menggunakan persamaan 2.2 dengan ketentuan:

$$CosSim(U,P) = \frac{U * P}{||U|| \ * \ ||P||}$$

Vektor Uji (**U**) = [1.0, 1.0, 1.0, 1.0, 0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 0.8095, 0, 1.0, 0.7051, 1.0, 1.0, 1.0]

Vektor Pembanding (**P**) = [1.0, 0.8957, 1.0, 1.0, 0, 0, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 1.0, 1.0, 1.0, 0.8957, 1.0, 1.0, 1.0]

U * P = (1.0 * 1.0) + (1.0 * 0.8957) + (1.0 * 1.0) + … + (0.7051 * 0.8957) + (1.0 * 1.0) + (1.0 * 1.0)) + (1.0 * 1.0) = **15.136**

||**U**|| = $\sqrt{1^2 + 1^2 + \cdots + 0.7051^2 + 1^2 + 1^2 + 1^2}$ = **3.994061967471211**

||**P**|| = $\sqrt{1^2 + 0.8957^2 + \cdots + 0.8957^2 + 1^2 + 1^2 + 1^2}$ = **4.0500709870322025**

Didapatkan:

$$CosSim(U,P) = \frac{15.136}{3.994061967471211 \ * \ 4.0500709870322025}$$

$$CosSim(U,P) = \mathbf{0.9356936563206849}$$

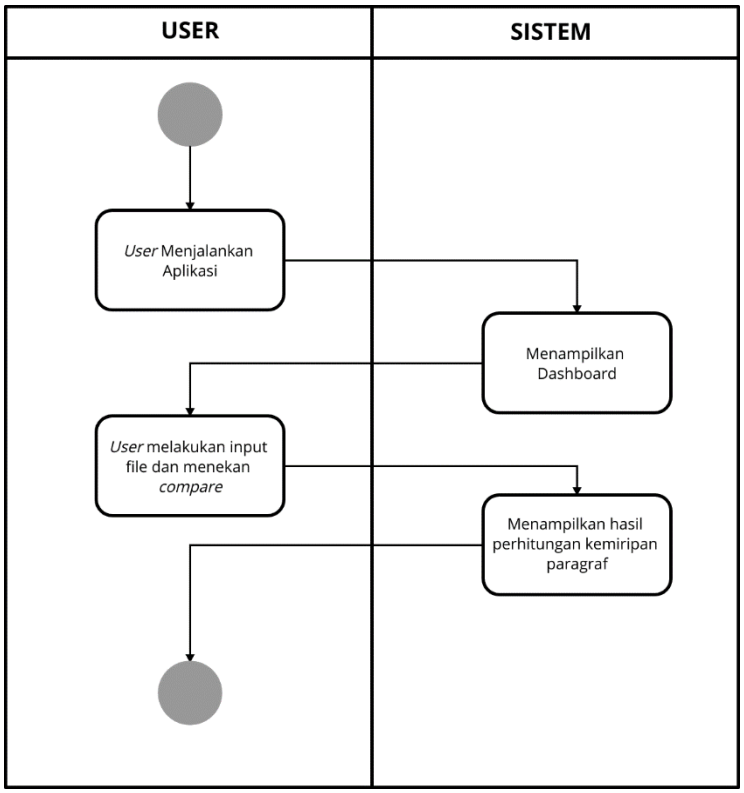
Dengan demikian perhitungan kemiripan teks antara dokumen uji dan dokumen pembanding menggunakan *cosine similarity* menghasilkan nilai akhir **0.9357**.

3.2.5. *Output*

Output akan menampilkan tingkat kemiripan dari nilai akhir yang didapat dari tahap sebelumnya dalam rentang nilai 0 sampai 1 dari dokumen uji dan dokumen pembanding, serta menentukan kemiripan dari dokumen dengan pembagian mirip dan tidak mirip .

3.3. Diagram Alur Sistem

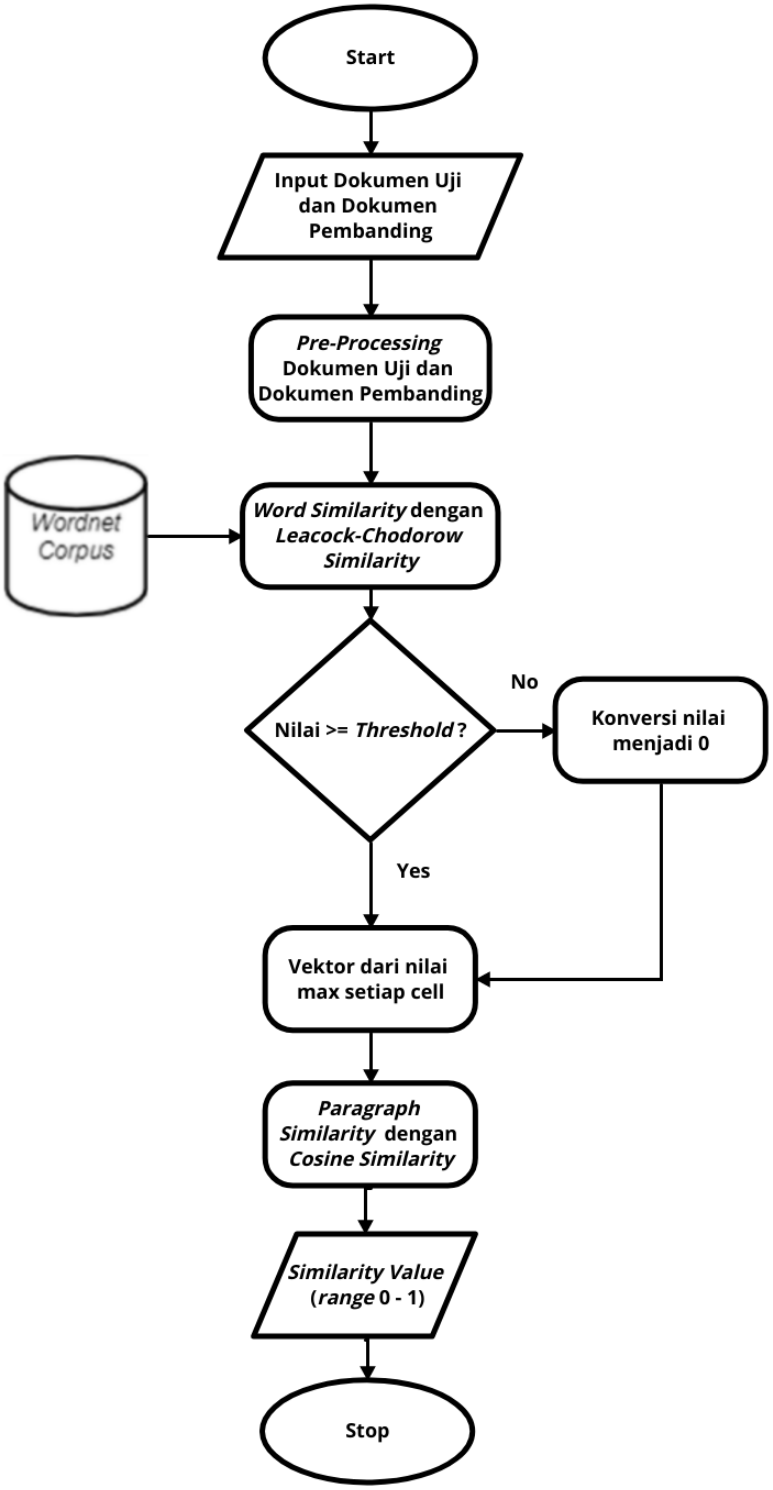
Tahap ini berisikan alur dan penjelasan dari aktivitas yang dapat dilakukan oleh system. Pengguna akan melakukan input paragraf uji dan paragraf pembanding dari sebuah dokumen atau dalam bentuk teks yang kemudian tingkat kemiripan dari kedua input akan dihitung sehingga menghasilkan nilai kemiripan dari kedua input yang akan ditampilkan pada layer beserta rincian dari hasil yang telah didapat. Diagram alur sistem seperti pada Gambar 3.8.



Gambar 3.8 Diagram Alur Sistem

3.4 *Flowchart* Kemiripan Paragraf

Flowchart memberikan penjelasan dalam bentuk gambaran alur serta proses perhitungan kemiripan antar teks yang diinput. *Flowchart* Kemiripan Paragraf serperti pada Gambar 3.9.



Gambar 3.9 *Flowchart* Kemiripan Paragraf

Flowchart sistem perhitungan kemiripan paragraf teks dokumen memiliki penjelasan sebagai berikut:

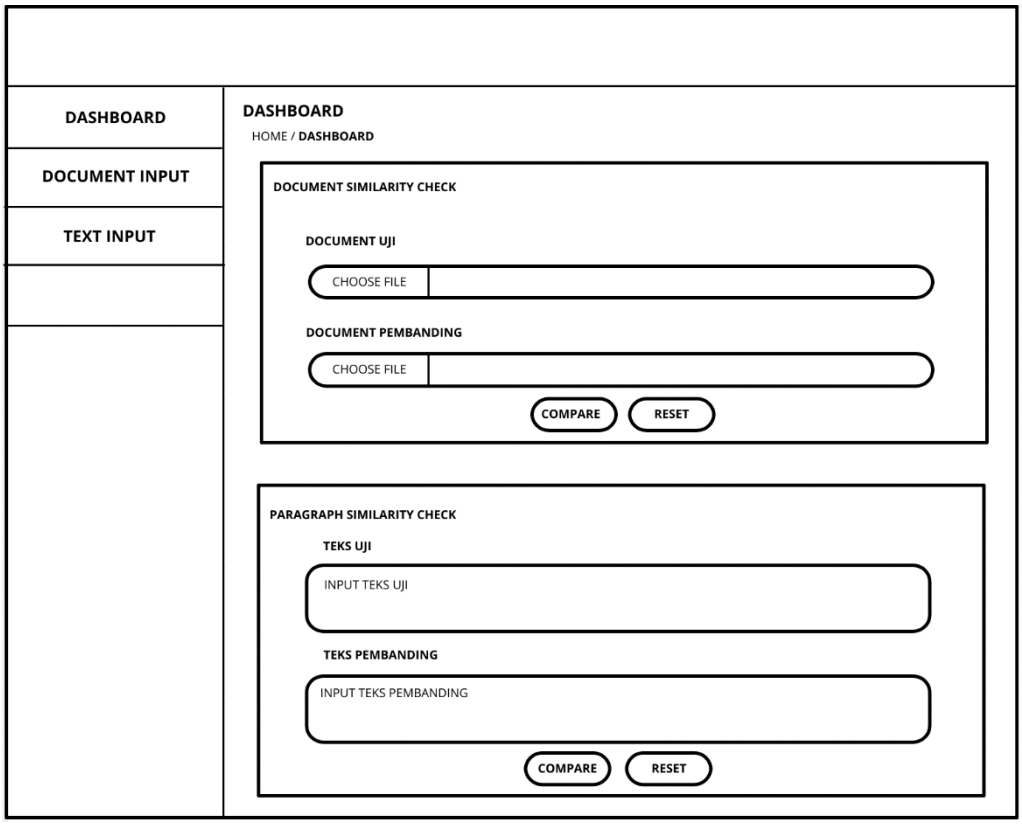
1. Mulai menjalankan sistem.
2. Melakukan *input* dokumen uji serta dokumen pembanding. format yang dapat diterima oleh sistem adalah .doc, docx, .txt, .pdf serta input langsung.
3. Melakukan *preprocessing* terhadap dokumen uji serta dokumen pembanding yang diinput.
4. Melakukan perhitungan *word similarity* menggunakan algoritma *Leacock Chodorow Similarity* dengan *WordNet Corpus*.
5. Melakukan pengecekan nilai *word similarity* terhadap nilai *threshold* yang didapatkan dengan ketentuan jika hasil nilai kemiripan lebih besar dibanding nilai *threshold*, nilai *word similarity* dikonversi menjadi 0.
6. Melakukan pembentukan vektor paragraf uji serta paragraf pembanding dengan ketentuan vektor berisi nilai maksimum dari setiap cell seperti pada Gambar 3.6 serta Gambar 3.7.
7. Melakukan kalkulasi *paragraph similarity* antara paragraf uji serta paragraf banding dengan menggunakan algoritma *cosine similarity*.
8. Menghasilkan *output* berupa tingkat kemiripan paragraf uji dan paragraf banding dengan rentang antara nilai 0 dan 1.
9. Sistem selesai dijalankan.

3.5. Perancangan Antarmuka Sistem

Rancangan antarmuka yang digunakan dalam sistem yang dikembangkan untuk penelitian ini akan dipaparkan dalam bagian ini.

3.5.1. Rancangan Halaman *Dashboard*

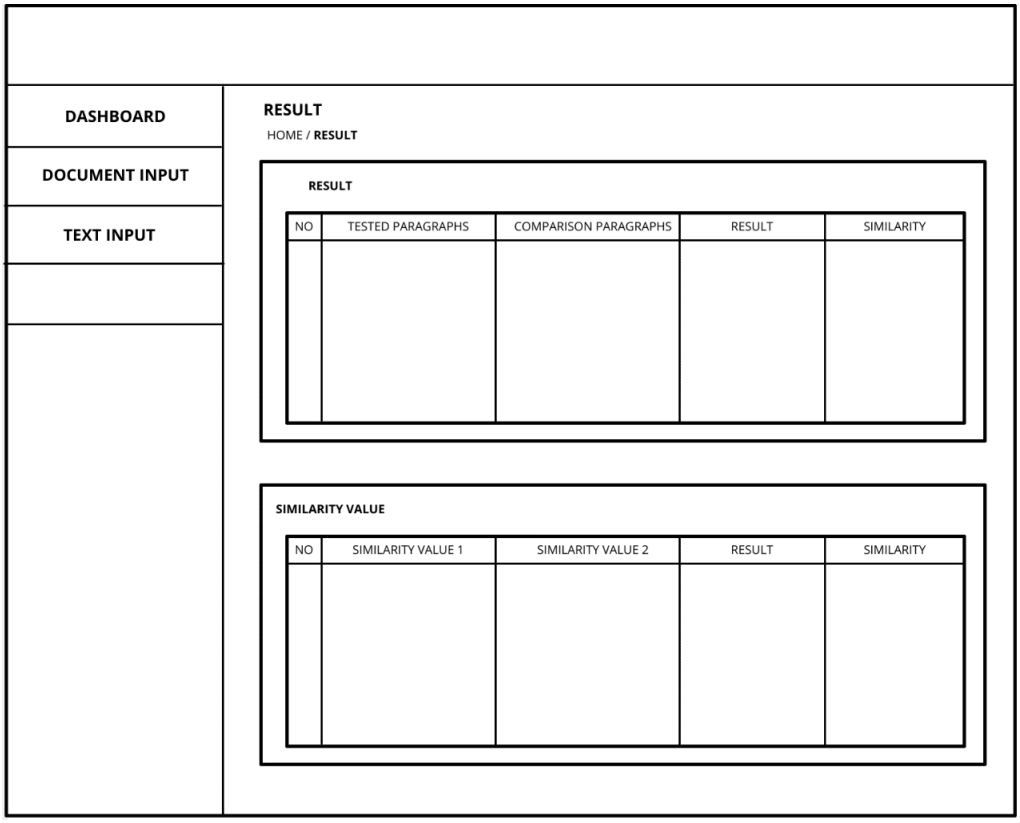
Halaman *Dashboard* adalah tampilan utama dimana sistem dapat digunakan. pada halaman ini terdapat *form upload* untuk dokumen uji serta dokumen pembanding. Selain form upload untuk dokumen juga terdapat *text area* yang berfungsi sebagai input teks secara langsung baik untuk teks uji maupun teks pembanding tanpa menggunakan input *via upload file* dokumen. Kedua jenis input memiliki tombol "*compare"* untuk memulai proses perhitungan kemiripan paragraf serta tombol "*reset"* untuk membersihkan form saat ingin melakukan input ulang. Rancangan Halaman *Dashboard* seperti terlihat pada Gambar 3.10.



Gambar 3.10 Rancangan Halaman *Dashboard*

3.5.2. Rancangan Halaman *Result*

Halaman *Result* adalah tampilan dimana sistem akan menampilkan hasil perhitungan dari dokumen uji serta dokumen pembanding yang diinput pada halaman sebelumnya. Pada halaman ini terdapat *Result* yang berisikan paragraf yang didapatkan dari dokumen uji serta dokumen pembanding, nilai kemiripan dari kedua paragraf, serta tingkat kemiripan dari kedua paragraf yang ditentukan oleh sistem. Selain itu juga terdapat *Details* yang berisikan vektor kata dari kedua paragraf yang dibandingkan dan *POS-Tag* dari tiap kata, serta *Similarity Value* yang berisikan vektor nilai dari kedua paragraf yang dibandingkan. Rancangan Halaman *Result* seperti terlihat pada Gambar 3.11.



Gambar 3.11 Rancangan Halaman *Result*

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

4.1.1. Spesifikasi Perangkat Keras dan Perangkat Lunak

Dalam proses pelaksanaan penelitian, penulis menggunakan beberapa perangkat keras selama melakukan implementasi sistem, spesifikasi perangkat keras berupa:

1. *Processor: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz (4 CPUs), ~2.7GHz*
2. *GPU: NVIDIA GeForce 940MX*
3. *RAM: 8192MB RAM*

Dalam penelitian ini, penulis juga menggunakan beberapa perangkat lunak selama melakukan implementasi sistem, spesifikasi perangkat lunak berupa:

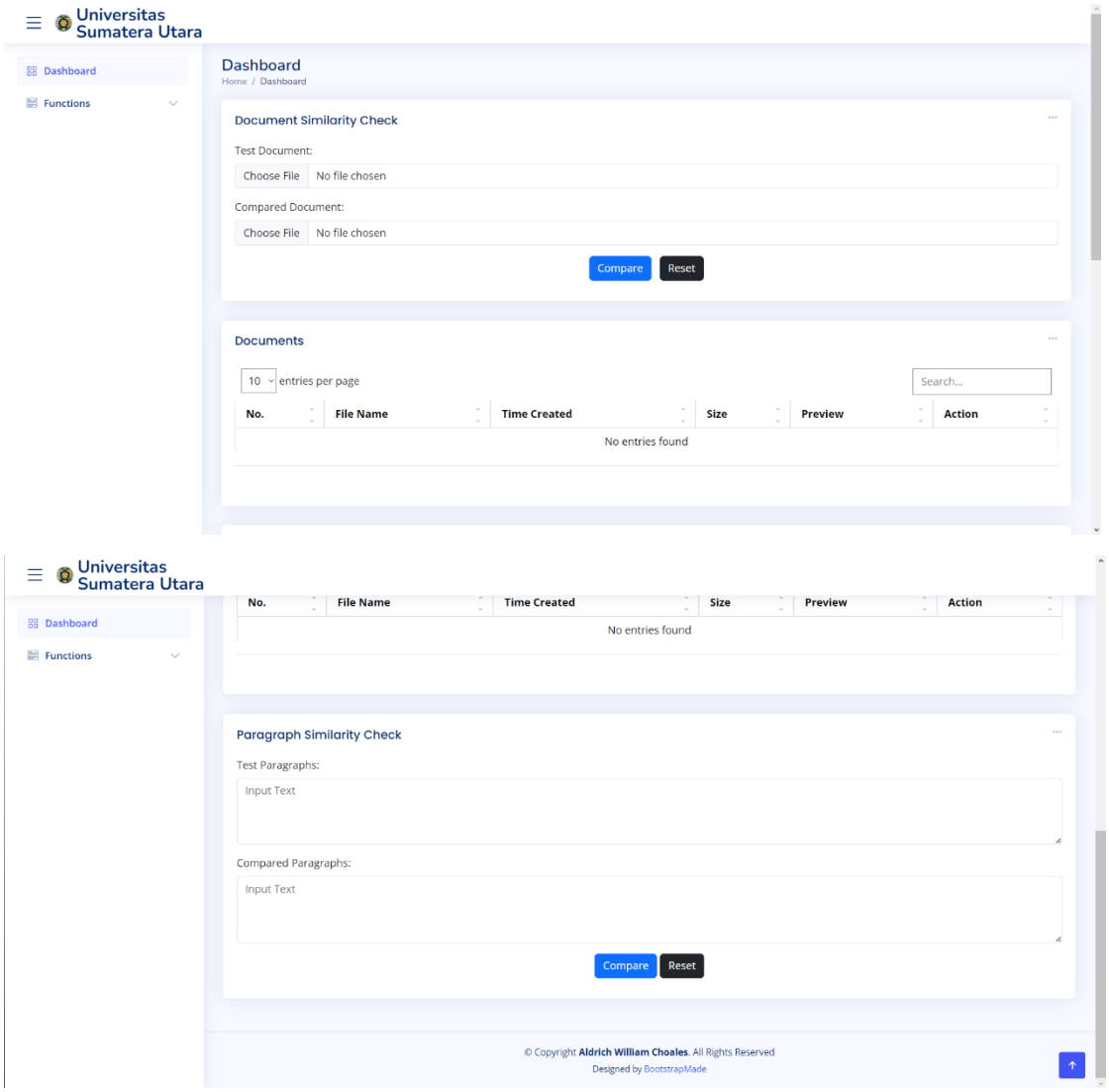
1. *Operating System: Windows 10 Pro 64 Bit*
2. *Source Code Editor: Atom*
3. *Python* (versi 3.10.2)
4. *Library* yang berasal dari bahasa *python*, sebagai berikut:
    a. *Flask (versi 2.2.2)*
    b. *Langdetect (versi 1.0.9)*
    c. *NLTK (versi 3.8.1)*
    d. *Numpy (versi 1.24.2)*
    e. *Pandas (versi 1.5.3)*
    f. *PyPDF2 (versi 3.0.1)*
    g. *Sastrawi (versi 1.0.1)*
    h. *Sklearn (versi 1.2.1)*
5. *Google Collaboratory*

4.1.2. Implementasi Perancangan Antarmuka

Implementasi perancangan antarmuka berupa hasil dari antarmuka yang sebelumnya telah dirancang dalam pembahasan pada Bab 3. Implementasi dari rancangan antarmuka dalam penelitian yaitu:

1. Tampilan Halaman *Dashboard*

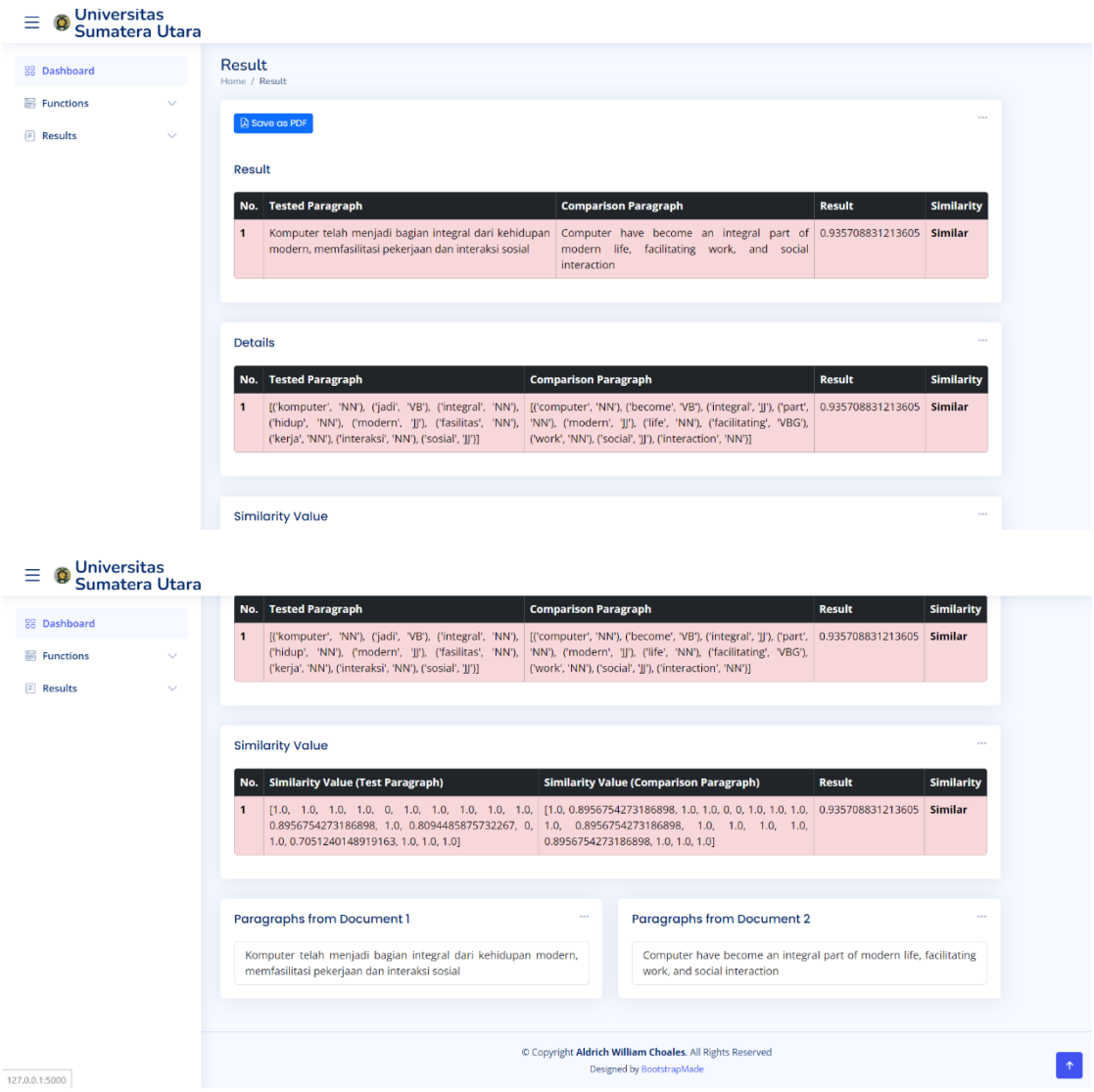
Tampilan halaman *dashboard* dijelaskan sebagai sebuah halaman yang berisi *form upload* dokumen uji serta dokumen pembanding dan *form text input* secara langsung. Implementasi dari tampilan *dashboard* seperti pada Gambar 4.1.



Gambar 4.1 Tampilan Halaman *Dashboard*

2. Tampilan Halaman *Result*

Tampilan halaman *result* dijelaskan sebagai sebuah halaman dimana hasil perhitungan dari dokumen uji serta dokumen pembanding ditampilkan. *Result, Details* serta *Similarity Value* merupakan output dari sistem. Implementasi dari tampilan *result* seperti pada Gambar 4.2.



Gambar 4.2 Tampilan Halaman *Result*

4.2. Hasil Pengujian Sistem

Sistem dalam penelitian ini diuji dengan tujuan mengetahui seberapa baik program dalam upaya menemukan kemiripan antar teks. Dalam pengujian sistem digunakan data uji dan data pembanding sebanyak 28 dokumen dengan dokumen uji sebanyak 7 dokumen dengan 146 total paragraf uji serta dokumen pembanding sebanyak 21 dokumen dengan 438 total paragraf pembanding. Setelah dilakukan pengujian, hasil pengujian yang didapatkan oleh sistem akan dicocokkan dengan pengujian oleh manusia secara manual.

Sistem perlu memiliki kemampuan untuk mengenal apakah paragraf mirip atau tidak mirip. Untuk itu, nilai kemiripan dari *Leacock Chodorow* dan *cosine similarity* memiliki rentang dari nilai 0 sampai nilai 1 dengan ketentuan apabila nilai kemiripan paragraf diatas 0.5 maka sistem akan menghasilkan status kemiripan mirip, sedangkan nilai kemiripan paragraf dibawah nilai tersebut menghasilkan status kemiripan tidak mirip. Pengukuran nilai kemiripan terhadap nilai *threshold* juga perlu dilakukan agar sistem mampu melakukan prediksi dengan lebih baik. Terdapat beberapa percobaan nilai *threshold* yang dipakai terhadap data yang digunakan, yaitu nilai *threshold* 0.5, 0.6, serta 0.7. Hasil percobaan *threshold* mendapatkan hasil yang memuaskan pada nilai *threshold* 0.7. Pengujian kemiripan dengan *threshold* seperti pada Tabel 4.1, Tabel 4.2 serta Tabel 4.3.

Tabel 4.1 Hasil Pengujian Dengan Nilai Threshold 0.5

No.	Tested Paragraph	Comparison Paragraph	Result	System	Manual
1	Ulasan adalah salah satu indikator yang digunakan untuk memberikan penilaian kredibilitas oleh pembeli terhadap penjual pada sebuah e-commerce. Ulasan yang diberikan pembeli sangat memengaruhi penilaian penjual di e-commerce. Ulasan juga merupakan bentuk perkembangan dari electronic word of mouth (e-WOM). E- WOM adalah bentuk komunikasi pemasaran di dunia maya yang tidak dapat dikendalikan oleh penjual maupun perusahaan. Karena informasi yang terdapat di dalam ulasan pembeli merupakan informasi yang tidak dikelola oleh pihak internal perusahaan.	Kredibilitas online merupakan salah satu hal penting yang perlu diperhatikan dalam perkembangan sektor e-commerce. Online review dalam platform e-commerce merupakan salah satu aspek yang dapat memengaruhi kredibilitas online suatu perusahaan atau penjual. Sebagai bagian dari perkembangan e-WOM, online review merupakan jenis komunikasi pemasaran di media digital yang tidak dapat dikontrol oleh pihak internal perusahaan. Kehadiran online review merupakan bagian dari fitur pada platform e-commmerce yang memfasilitasi consumer generated media (CGM).	0.9591	**Similar**	**Similar**

Tabel 4.1 Hasil Pengujian Dengan Nilai Threshold 0.5 (Lanjutan)

No.	Tested Paragraph	Comparison Paragraph	Result	System	Manual
2	Ulasan adalah salah satu indikator yang digunakan untuk memberikan penilaian kredibilitas oleh pembeli terhadap penjual pada sebuah e-commerce. Ulasan yang diberikan pembeli sangat memengaruhi penilaian penjual di e-commerce. Ulasan juga merupakan bentuk perkembangan dari electronic word of mouth (e-WOM). E- WOM adalah bentuk komunikasi pemasaran di dunia maya yang tidak dapat dikendalikan oleh penjual maupun perusahaan. Karena informasi yang terdapat di dalam ulasan pembeli merupakan informasi yang tidak dikelola oleh pihak internal perusahaan.	Online review pada platform e-commerce bersumber pada pengalaman konsumen, baik itu pengalaman mengonsumsi atau menggunakan produk, pengalaman terhadap pelayanan dan jasa penjual, bahkan dapat juga berkaitan dengan kecepatan pengiriman produk. Hal ini tentu dapat menjadi informasi tambahan bagi konsumen lain dan memengaruhi asumsi konsumen terhadap penjual atau produk yang bersangkutan. Kehadiran online review dalam platform e-commerce juga dapat memengaruhi aspek prominence dan interpretation yang berkaitan dengan kredibilitas online (Agustina et al., 2018).	0.9187	**Similar**	**Not Similar**
3	Ulasan adalah salah satu indikator yang digunakan untuk memberikan penilaian kredibilitas oleh pembeli terhadap penjual pada sebuah e-commerce. Ulasan yang diberikan pembeli sangat memengaruhi penilaian penjual di e-commerce. Ulasan juga merupakan bentuk perkembangan dari electronic word of mouth (e-WOM). E- WOM adalah bentuk komunikasi pemasaran di dunia maya yang tidak dapat dikendalikan oleh penjual maupun perusahaan. Karena informasi yang terdapat di dalam ulasan pembeli merupakan informasi yang tidak dikelola oleh pihak internal perusahaan.	Sentiment Analysis (SA), also referred to as Opinion Mining (OM), is the field of Natural Language Processing (NLP) responsible for extracting users subjective polarity which includes users attitudes, appraisals and emotions as regards many aspects of society, such as products, organisations, events or services. SA has, therefore, attracted attention owning to its potential for application in marketing, customer service, infodemiology, hate-speech identification, spam-filters or fake-news detection among other domains. (García-Díaz et al., 2020).	0.9148	**Similar**	**Not Similar**

Tabel 4.1 menunjukkan bahwa threshold 0.5 menghasilkan nilai kemiripan dengan akurasi yang tidak memuaskan karena banyaknya ketidak cocokan dari hasil sistem dengan penilaian kemiripan secara manual.

Tabel 4.2 Hasil Pengujian Dengan Nilai Threshold 0.6

No.	Tested Paragraph	Comparison Paragraph	Result	System	Manual
1	Ulasan adalah salah satu indikator yang digunakan untuk memberikan penilaian kredibilitas oleh pembeli terhadap penjual pada sebuah e-commerce. Ulasan yang diberikan pembeli sangat memengaruhi penilaian penjual di e-commerce. Ulasan juga merupakan bentuk perkembangan dari electronic word of mouth (e-WOM). E- WOM adalah bentuk komunikasi pemasaran di dunia maya yang tidak dapat dikendalikan oleh penjual maupun perusahaan. Karena informasi yang terdapat di dalam ulasan pembeli merupakan informasi yang tidak dikelola oleh pihak internal perusahaan.	Kredibilitas online merupakan salah satu hal penting yang perlu diperhatikan dalam perkembangan sektor e-commerce. Online review dalam platform e-commerce merupakan salah satu aspek yang dapat memengaruhi kredibilitas online suatu perusahaan atau penjual. Sebagai bagian dari perkembangan e-WOM, online review merupakan jenis komunikasi pemasaran di media digital yang tidak dapat dikontrol oleh pihak internal perusahaan. Kehadiran online review merupakan bagian dari fitur pada platform e-commmerce yang memfasilitasi consumer generated media (CGM).	0.8943	**Similar**	**Similar**

Tabel 4.2 Hasil Pengujian Dengan Nilai Threshold 0.6 (Lanjutan)

No.	Tested Paragraph	Comparison Paragraph	Result	System	Manual
2	Ulasan adalah salah satu indikator yang digunakan untuk memberikan penilaian kredibilitas oleh pembeli terhadap penjual pada sebuah e-commerce. Ulasan yang diberikan pembeli sangat memengaruhi penilaian penjual di e-commerce. Ulasan juga merupakan bentuk perkembangan dari electronic word of mouth (e-WOM). E- WOM adalah bentuk komunikasi pemasaran di dunia maya yang tidak dapat dikendalikan oleh penjual maupun perusahaan. Karena informasi yang terdapat di dalam ulasan pembeli merupakan informasi yang tidak dikelola oleh pihak internal perusahaan.	Online review pada platform e-commerce bersumber pada pengalaman konsumen, baik itu pengalaman mengonsumsi atau menggunakan produk, pengalaman terhadap pelayanan dan jasa penjual, bahkan dapat juga berkaitan dengan kecepatan pengiriman produk. Hal ini tentu dapat menjadi informasi tambahan bagi konsumen lain dan memengaruhi asumsi konsumen terhadap penjual atau produk yang bersangkutan. Kehadiran online review dalam platform e-commerce juga dapat memengaruhi aspek prominence dan interpretation yang berkaitan dengan kredibilitas online (Agustina et al., 2018).	0.8312	**Similar**	**Not Similar**
3	Ulasan adalah salah satu indikator yang digunakan untuk memberikan penilaian kredibilitas oleh pembeli terhadap penjual pada sebuah e-commerce. Ulasan yang diberikan pembeli sangat memengaruhi penilaian penjual di e-commerce. Ulasan juga merupakan bentuk perkembangan dari electronic word of mouth (e-WOM). E- WOM adalah bentuk komunikasi pemasaran di dunia maya yang tidak dapat dikendalikan oleh penjual maupun perusahaan. Karena informasi yang terdapat di dalam ulasan pembeli merupakan informasi yang tidak dikelola oleh pihak internal perusahaan.	Sentiment Analysis (SA), also referred to as Opinion Mining (OM), is the field of Natural Language Processing (NLP) responsible for extracting users subjective polarity which includes users attitudes, appraisals and emotions as regards many aspects of society, such as products, organisations, events or services. SA has, therefore, attracted attention owning to its potential for application in marketing, customer service, infodemiology, hate-speech identification, spam-filters or fake-news detection among other domains. (García-Díaz et al., 2020).	0.7760	**Similar**	**Not Similar**

Tabel 4.2 menunjukkan bahwa threshold 0.6 juga menghasilkan nilai kemiripan dengan akurasi yang tidak memuaskan karena menghasilkan banyak kesalahan.

Tabel 4.3 Hasil Pengujian Dengan Nilai Threshold 0.7

No.	Tested Paragraph	Comparison Paragraph	Result	System	Manual
1	Ulasan adalah salah satu indikator yang digunakan untuk memberikan penilaian kredibilitas oleh pembeli terhadap penjual pada sebuah e-commerce. Ulasan yang diberikan pembeli sangat memengaruhi penilaian penjual di e-commerce. Ulasan juga merupakan bentuk perkembangan dari electronic word of mouth (e-WOM). E- WOM adalah bentuk komunikasi pemasaran di dunia maya yang tidak dapat dikendalikan oleh penjual maupun perusahaan. Karena informasi yang terdapat di dalam ulasan pembeli merupakan informasi yang tidak dikelola oleh pihak internal perusahaan.	Kredibilitas online merupakan salah satu hal penting yang perlu diperhatikan dalam perkembangan sektor e-commerce. Online review dalam platform e-commerce merupakan salah satu aspek yang dapat memengaruhi kredibilitas online suatu perusahaan atau penjual. Sebagai bagian dari perkembangan e-WOM, online review merupakan jenis komunikasi pemasaran di media digital yang tidak dapat dikontrol oleh pihak internal perusahaan. Kehadiran online review merupakan bagian dari fitur pada platform e-commmerce yang memfasilitasi consumer generated media (CGM).	0.7547	**Similar**	**Similar**

Tabel 4.3 Hasil Pengujian Dengan Nilai Threshold 0.7 (Lanjutan)

2	Ulasan yang diberikan oleh pengguna di platform e-commerce dapat menjadi informasi bagi pengguna lainnnya. Ulasan biasanya berasal dari pengalaman pembeli dalam berbelanja, ulasan tersebut dapat berupa produk yang dijual, respon penjual, kecepatan pengiriman, dan sebagainya (Agustina et al., 2018).	Online review pada platform e-commerce bersumber pada pengalaman konsumen, baik itu pengalaman mengonsumsi atau menggunakan produk, pengalaman terhadap pelayanan dan jasa penjual, bahkan dapat juga berkaitan dengan kecepatan pengiriman produk. Hal ini tentu dapat menjadi informasi tambahan bagi konsumen lain dan memengaruhi asumsi konsumen terhadap penjual atau produk yang bersangkutan. Online review pada platform e-commerce juga secara langsung berkaitan dengan rating atau penilaian penjual ataupun produk platform e-commerce. Kehadiran online review dalam platform e-commerce juga dapat memengaruhi aspek prominence dan interpretation yang berkaitan dengan kredibilitas online (Agustina et al., 2018).	0.7157	**Similar**	**Similar**
3	Analisis Sentimen atau juga disebut Opinion Mining, merupakan bidang Natural Language Processing (NLP) yang bertanggung jawab untuk mengekstrak polaritas subjektif pengguna mengenai topik tertentu. Polaritas subjektif mencakup sikap, penilaian, dan emosi pengguna sehubungan dengan banyak aspek masyarakat, seperti produk, organisasi, acara, atau layanan. Oleh karena itu, analisis sentimen telah menarik perhatian karena potensinya untuk aplikasi dalam pemasaran, layanan pelanggan, infodemiologi, identifikasi ujaran kebencian, filter spam, atau deteksi berita palsu di antara domain lainnya (García-Díaz et al., 2020).	Sentiment Analysis (SA), also referred to as Opinion Mining (OM), is the field of Natural Language Processing (NLP) responsible for extracting users' subjective polarity concerning a specific topic. Subjective polarity includes users' attitudes, appraisals and emotions as regards many aspects of society, such as products, organisations, events or services. SA has, therefore, attracted attention owning to its potential for application in marketing, customer service, infodemiology, hate-speech identification, spam-filters or fake-news detection among other domains. (García-Díaz et al., 2020).	0.7249	**Similar**	**Similar**

Tabel 4.3 Hasil Pengujian Dengan Nilai Threshold 0.7 (Lanjutan)

4	Analisis sentimen berbasis aspek melakukan analisis yang lebih halus, yang berfokus pada mengidentifikasi ekspresi sentimen dari aspek target dalam dokumen tertentu. Analisis sentimen berbasis aspek bertujuan untuk mengidentifikasi aspek entitas dalam dokumen, dan untuk setiap aspek yang diidentifikasi, polaritas sentimen diperkirakan berdasarkan pendekatan tertentu (Zainuddin et al., 2018).	Basically, an aspect-based sentiment analysis (ABSA) performs a finer-grained analysis, which is also defined as are search problem that focuses on identifying the sentiment expressions of aspects of the target within a given document. Moreover, an ABSA aims at identifying the aspects of entities in the document, and for each identified aspect, the sentiment polarity is estimated based on a specific approach. Most significantly, sentiment analyses at the document level and the sentence level do not determine exactly what people liked or did not like. (Zainuddin et al., 2018).	0.71276 9234532 0821	**Similar**	**Similar**
…	…	…	…	…	…
438	Penelitian mengenai klasifikasi depresi dari konten Twitter pernah dilakukan oleh Budiman et. al. (2021) yang menggunakan algoritma Multinomial Naïve Bayes (MNB) dan Complement Naïve Bayes (CNB), MNB berhasil mencapai akurasi sebesar 91.30% dan CNB berhasil mencapai akurasi 91.98%	Dalam penelitian ini, data dikumpulkan melalui pencarian kata kunci yang mengindikasikan gangguan depresi di platform Twitter. Berdasarkan dataset tersebut, sebuah model prediktif dikembangkan menggunakan metode Multinomial Naïve Bayes (MNB) dan Complement Naïve Bayes (CNB) sebagai metode klasifikasi, serta metode Term Frequency–Inverse Document Frequency (TF–IDF) sebagai metode ekstraksi fitur. Berdasarkan hasil eksperimen, kombinasi metode TF–IDF dan MNB berhasil mencapai tingkat F-score sebesar 91.30%, sementara kombinasi metode TF–IDF dengan CNB mencapai tingkat performa sebesar 91.98%.	0.5853	**Similar**	**Similar**

4.3. Evaluasi

Evaluasi sistem dilakukan dengan tujuan untuk mengetahui seberapa baik program dalam menemukan kemiripan teks. Pada penelitian ini, dilakukan evaluasi untuk menilai kemampuan sistem dalam mengidentifikasi masalah menggunakan perhitungan *confusion matrix*.

Confusion Matrix didefinisikan sebagai sebuah tabel yang menunjukkan representasi baik atau tidak kerja sistem dalam melakukan klasifikasi. Matriks menunjukkan jumlah hasil sistem yang terklasifikasi dengan benar dan yang terklasifikasi dengan salah, dibandingkan dengan hasil sebenarnya (nilai target) dalam data uji. Didalam *confusion matrix*, nilai akurasi, presisi, *recall*, serta *F-Measure* yang didapatkan oleh algoritma sistem akan dihitung.

Terdapat beberapa nilai yang perlu didapatkan terlebih dahulu sebelum menghitung empat data sebelumya, berupa *True Positive* (TP), *True Negative* (TN), *False Negative* (FN), dan *False Positive* (FP). TP adalah kuantitas data positif yang benar diklasifikasi sebagai positif oleh sistem, seperti paragraf mirip yang ditentukan mirip oleh sistem. TN adalah kuantitas data negatif yang benar diklasifikasi sebagai negatif oleh oleh sistem, seperti paragraf tidak mirip yang ditentukan tidak mirip oleh sistem. FP adalah kuantitas data negatif yang salah diklasifikasi sebagai positif oleh sistem, seperti paragraf tidak mirip yang ditentukan mirip oleh sistem. FN adalah kuantitas data positif yang salah diklasifikasi sebagai negatif oleh sistem, seperti paragraf mirip yang ditentukan tidak mirip oleh sistem. Hasil pengujian menunjukkan

Hasil pengujian sistem dengan nilai *threshold* 0.7 mendapatkan data TP, TN, FP, serta FN seperti pada Tabel 4.4.

Tabel 4.4 Jumlah TP, TN, FP, FN pada *threshold* 0.7

Jumlah Paragraf	TP	TN	FP	FN
438	227	177	23	11

Tabel 4.4 menunjukkan bahwa terdapat beberapa data yang setelah dilakukan pengujian mengembalikan FP serta FN. Setelah paragraf dilakukan pemeriksaan kembali secara manual di dapatkan bahwa data FP serta FN dihasilkan karena sistem tidak dapat mengidentifikasi beberapa kata pada paragraf dikarenakan kurang lengkapnya *wordnet corpus* yang dipakai, sehingga terdapat hasil kemiripan data yang tidak tepat.

Selanjutnya, dengan menggunakan data dari Tabel 4.4, nilai akurasi, presisi, *recall*, serta *F-Measure* dapat dihitung. Penjelasan serta perhitungan nilai akurasi, presisi, *recall*, serta *F- Measure* seperti:

Akurasi mengukur kemampuan keseluruhan model klasifikasi dalam mengenali dengan benar contoh-contoh dari semua kelas, baik positif maupun negatif. Persamaan untuk menghitung nilai akurasi seperti pada persamaan 4.1.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \qquad (4.1)$$

Sehingga didapatkan perhitungan:

$$Akurasi = \frac{227 + 177}{227 + 177 + 23 + 11}$$

$$Akurasi = \frac{404}{438}$$

$$Akurasi = \mathbf{0.923}$$

Presisi memiliki fungsi mengukur tingkat ketepatan model dalam mengklasifikasikan contoh sebagai positif. Persamaan untuk menghitung nilai presisi seperti pada persamaan 4.2.

$$Presisi = \frac{TP}{TP+FP} \qquad (4.2)$$

Sehingga didapatkan perhitungan:

$$Presisi = \frac{227}{227 + 23}$$

$$Presisi = \frac{227}{250}$$

$$Presisi = \mathbf{0.908}$$

Recall memiliki fungsi mengukur kemampuan model untuk mengidentifikasi semua contoh positif yang sebenarnya. Persamaan untuk menghitung nilai *recall* seperti pada persamaan 4.3:

$$Recall = \frac{TP}{TP+FN} \qquad (4.3)$$

Sehingga didapatkan perhitungan:

$$Recall = \frac{227}{227 + 11}$$

$$Recall = \frac{227}{238}$$

$$Recall = \mathbf{0.953}$$

F-Measure adalah nilai rata-rata harmonik antara presisi dan *recall*, memberikan keseimbangan antara kedua metrik tersebut. Persamaan untuk menghitung nilai recall seperti pada persamaan 4.4:

$$F1 = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \quad (4.3)$$

Sehingga didapatkan perhitungan:

$$F1 = 2 \times \left(\frac{0.908 \times 0.953}{0.908 + 0.953}\right)$$

$$F1 = 2 \times \left(\frac{0.867}{1.861}\right)$$

$$F1 = \mathbf{0.930}$$

Berdasarkan hasil perhitungan, pengujian sistem dengan nilai threshold 0.7 mendapatkan akurasi dengan nilai 0.923 atau 92.3%, presisi dengan nilai 0.908 atau 98.8%, *recall* dengan nilai 0.953 atau 95.3%, serta *F- Measure* dengan nilai 0.93 atau 93%. Dari nilai akurasi yang didapatkan, didapat kesimpulan bahwa sistem dapat melakukan deteksi kemiripan teks paragraf dalam dokumen dengan algoritma *Leacock Chodorow* dan *cosine similariy* dengan baik.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari sistem pendeteksi kemiripan teks paragraf dokumen yang digunakan dalam penelitian ini, didapatkan beberapa kesimpulan sebagai berikut:

1. Sistem mampu melakukan deteksi kemiripan isi dokumen karya ilmiah berdasarkan paragraf secara otomatis dengan menggunakan algoritma *Leacock Chodorow* dan *cosine similarity* dalam dua bahasa, berupa Bahasa Indonesia serta Bahasa Inggris.
2. Metode *Leacock Chodorow* dan *cosine similarity* dapat digunakan dalam melakukan deteksi kemiripan pada teks paragraf dalam dokumen dari segi leksikal maupun semantic.
3. Hasil pengujian didapatkan dengan menggunakan *threshold* 0.7 untuk memisahkan paragraf tidak mirip dan mirip dengan nilai Akurasi sebesar 0.923 atau 92.3%, nilai Presisi sebesar 0.908 atau 90.8%, nilai *Recall* sebesar 0.953 atau 95.3%, serta nilai *F-Measure* sebesar 0.930 atau 93%.

5.2. Saran

Saran yang dapat diberikan dari hasil penelitian oleh penulis dalam usaha pengembangan penelitian berikutnya yaitu sebagai berikut:

1. Proses perolehan data yang digunakan dalam penelitian bersifat manual dan dengan demikian memakan waktu yang lama. Saran yang dapat diberikan adalah untuk mencari altrenatif dalam proses perolehan data agar dapat meningkatkan kinerja sistem secara menyeluruh.

2. Dikarenakan sistem melakukan perhitungan nilai kemiripan paragraf antar dokumen secara *brute force*, semakin besar dokumen yang dimasukkan sebagai data uji serta data pembanding, waktu yang dibutuhkan sistem untuk menyelesaikan perhitungan juga semakin besar. Banyak langkah proses dalam komputasi serta rumitnya data menyebabkan sistem untuk memakan waktu yang lama dalam melakukan perhitungan nilai kemiripan. Saran yang dapat diberikan adalah untuk mencari alternatif dalam metode perhitungan nilai kemiripan paragraf, baik berdasarkan kata kunci yang sering muncul dari masing-masing paragraf, ataupun menggunakan taksonomi yang telah disediakan oleh *wordnet corpus* sebagai dasar penentuan paragraf mana saja yang perlu untuk dihitung nilai kemiripannya, agar dapat meringankan serta mempercepat kinerja sistem.

3. Sistem penelitian mendapatkan beberapa data yang tidak tepat, mengurangi tingkat akurasi sistem. Data yang tidak tepat ini didapatkan karena *wordnet corpus* yang digunakan dalam penelitian ini memiliki beberapa kata yang tidak terdaftar secara lengkap dalam kedua bahasa (beberapa kata terdaftar dalam *corpus* namun hanya dalam satu bahasa) yang digunakan dalam penelitian, sehingga sistem tidak dapat melakukan proses dengan benar. Saran yang dapat diberikan adalah untuk meningkatkan kualitas dari *corpus* yang dipakai kedepannya, baik dengan cara melengkapi ataupun mengubah *corpus* yang dipakai agar dapat mengurangi tingkat kesalahan sistem.

DAFTAR PUSTAKA

Anonymous, 2017. ECMA-404 The JSON data interchange syntax. 2^{nd} edition. (*Online*) https://www.ecma-international.org/publications-and-standards/standards/ecma-404/ (15 Januari 2023).

Ariantini, D.A.R., Lumenta, A.S.M. & Jacobus, A. 2016. Pengukuruan Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode *Cosine Similarity. E- Journal Teknik Informatika,* 9(1): pp. 1-8. (Online) https://ejournal.unsrat.ac.id/index.php/informatika/article/view/13752 (27 Desember 2022).

Burnette, E. 2008. Hello, Android: Introducing Google's Mobile Development Platform. *Pragmatic Bookshelf.*

Chiru, C.-G., Truică, C.-O., Apostol, E.-S. & Ionescu, A., Improving WordNet using Word Embeddings. *2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 121-128, doi: 10.1109/SYNASC54541.2021.00030.

Fatonah, S., Hadinegoro, A. & Hartanto, A.D. 2020. Deteksi Kemiripan Abstraksi Tugas Akhir Diploma Informatika Universitas AMIKOM Yogyakarta dengan Algoritma Rabin Karp. *JURIKOM (Jurnal Riset Komputer)* 7(1): pp. 1-6, doi: 10.30865/jurikom.v7i1.1927.

Firdaus, A., Ernawati. & Firdaus, A.V. 2014. Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma Nazief & Andriani Dan Metode *Cosine Similarity. Jurnal Teknologi Informasi* 10(1): pp. 96-109.

Ganesan, K. 2015. What is Text Similarity. (*Online*) https://kavita-ganesan.com/what- is-text-Similarity/#.X_XE4FMxdkw (27 Desember 2022).

Gokul, P.P., Akhil, B.K. & Shiva, K.K.M. 2017. Sentence similarity detection in Malayalam language using cosine similarity. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 221-225, doi: 10.1109/RTEICT.2017.8256590.

Guo, J. 1997. Critical Tokenization and its Properties. *Computational Linguistics* 20(4): pp. 569-596.

Han, J., Kamber, M. & Pei, J. 2012. *Data Mining: Concepts and Techniques*. 3rd Edition. Elsevier: Amsterdam.

Hartanto, A.D., Pristyanto, Y. & Saputra, A. 2021. Document Similarity Detection using Rabin-Karp and Cosine Similarity Algorithms. *2021 International Conference on Computer Science and Engineering (IC2SE)*, pp. 1-6, doi: 10.1109/IC2SE52832.2021.9791999.

Imbar, R.V., Adelia., Ayub, M. & Rehatta, A. 2014. Implementasi *Cosine Similarity* dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks. *Jurnal Informatika* 10(1): pp. 31 - 42.

Leacock, C. & Chodorow, M. 1998. "Combining Local Context and WordNet Similarity for Word Sense Identification, WordNet: An Electronic Lexical Database". *MIT Press*, pp. 265-283, doi: 10.7551/mitpress/7287.003.0018.

Liddy, E.D. 2001. *Encyclopedia of Library And Information Science*. 2^{nd} Edition. Marcel Decker: New York.

Madani, Y., Erritali, M. & Jamaa, B. 2019. Sentiment analysis using semantic similarity and Hadoop MapReduce. *Knowledge and Information Systems*, doi: 59. 10.1007/s10115-018-1212-z.

Millah, A. & Nurazizah, S. 2017. Perbandingan Penggunaan Algoritma Cosinus dan Wu Palmer untuk Mencari Kemiripan Kata dalam Plagiarism Checker. *Jurnal Ilmu Komputer dan Desain Komunikasi Visual (JIKDISKOMVIS)* 2(1): pp. 15-25.

Miller, G.A., Beckwith, R., Fellbaum, C., Derek, G. & Miller, K.J. 1990. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4): pp. 235-244, doi: 10.1093/ijl/3.4.235.

Muttaqin, F.A. & Bachtiar A.M. 2019. Implementasi Teks Mining Pada Aplikasi Pengawasan Penggunaan Internet Anak "Dodo Kids Browser. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.

Petrović, Đ. & Stanković, M. 2019. The Influence of Text Preprocessing Methods and Tools on Calculating Text Similarity. *UNIV NIS*, pp. 973-994. doi: 10.22190/FUMI1905973D

Reitz, J.M. 2004. *Dictionary for library and information science / Joan M. Reitz*. London: Libraries Multimed.

Soyusiawaty, D. & Zakaria, Y. 2018. Book Data Content Similarity Detector With Cosine Similarity (Case study on digilib.uad.ac.id). *2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, pp. 1-6, doi: 10.1109/TSSA.2018.8708758.

Syaifudin, Y., Saputra, Pramana, Y. & Puspitasari, Dwi. 2018. The implementation of web service based text preprocessing to measure Indonesian student thesis similarity level. *MATEC Web of Conferences*, pp. 197. doi: 10.1051/matecconf/201819703019.

Prasetya, D.D., Wibawa, A.P. & Hirashima, T. 2018. The performance of text *Similarity* algorithms. *International Journal of Advances in Intelligent Informatics* 4(1): pp. 63-69.

Wahyuni, R.T., Prastiyanto, D., & Supraptono, E. 2017. Penerapan Algoritma *Cosine Similarity* dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro*, 9(1), pp. 18-23. doi: 10.15294/jte.v9i1.10955

Wicaksana, I. W. S. & Hakim, R. H. 2006. Pendekatan *Schema Matching* dalam Bahasa Indonesia. *Universitas Gunadarma*. (*Online*) https://www.semanticscholar.org/paper/Pendekatan-Schema-Matching-Dalam-bahasa-Indonesia-Wicaksana-Hakim/fa36edefba62f0427ef42d3bc1efe6d65cc6fd5e (15 Desember 2023)

Wu, Z. & Palmer, M. 1994. Verb Semantics And Lexical Selection. *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, pp. 133-138.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007
Laman: http://Fasilkomti.usu.ac.id

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
NOMOR :235/UN5.2.1.14/SK/SPB/2024

DEKAN FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 8 Januari 2024 perihal permohonan ujian skripsi:

Nama	: ALDRICH WILLIAM CHOALES
NIM	: 181402074
Program Studi	: Sarjana (S-1) Teknologi Informasi
Judul Skripsi	: Pendeteksi Kemiripan Teks Paragraf Dalam Dokumen Menggunakan Algoritma Leacock Chodorow dan Cosine Similarity

Memperhatikan : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.

Menimbang : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan

Mengingat :
1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026

MEMUTUSKAN

Menetapkan :

Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:

Ketua	: Ivan Jaya S.Si., M.Kom. NIP: 198407072015041001
Sekretaris	: Dr. Maristha Elveny S.TI, M.Kom NIP: 199003272017062001
Anggota Penguji	: Prof. Dr. Drs. Opim Salim Sitompul M.Sc NIP: 196108171987011001
Anggota Penguji	: Dr. Erna Budhiarti Nababan M.IT NIP: 196210262017042001
Moderator	: -
Panitera	: -

Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.

Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

Ditetapkan di : Medan
Pada Tanggal : 11 Januari 2024
Dekan,

MAYA SILVIA LYDIA
NIP 197401272002122001

Tembusan :
1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
2. Yang bersangkutan
3. Arsip