Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2025129143
Vol. 12, No. 1, Februari 2025, hlm. 133-144 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

ANALISIS PERBANDINGAN TEKNIK *WORD2VEC* DAN *DOC2VEC* DALAM MENGUKUR KEMIRIPAN DOKUMEN MENGGUNAKAN *COSINE SIMILARITY*

Dede Iskandar[*1], Ana Kurniawati[2]

[1,2]Universitas Gunadarma Depok, Depok
Email: [1]ddiskandar84@gmail.com, [2]ana@staff.gunadarma.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 31 Agustus 2024, diterima untuk diterbitkan: 10 Februari 2025)

Abstrak

Tempatkan Era digital memudahkan akses dokumen online dalam jumlah besar menjadi lebih mudah dan cepat, namun juga menimbulkan tantangan kompleks dalam pengelolaan dan analisis informasi. Salah satu tantangan utama adalah mengukur kemiripan antar dokumen, yang penting untuk berbagai aplikasi seperti deteksi plagiarisme. Menanggapi tantangan ini, banyak teknik yang dapat digunakan dalam merepresentasikan dokumen menjadi vektor untuk mengukur kemiripan dokumen. Dalam penelitian ini teknik *Word2vec* dan *Doc2vec* digunakan untuk merepresentasikan dokumen menjadi vektor, dan dalam mengukur kemiripan dokumen menggunakan metode *Cosine Similarity*. Objek penelitian dilakukan pada paragraf abstrak dari 20 jurnal ilmiah dengan tema data *mining* yang diterbitkan antara tahun 2020 hingga 2024 dari E-Journal Universitas Gunadarma. Metodologi penelitian meliputi pengumpulan data, *text mining*, pra-pemrosesan teks, implementasi teknik *Word2vec* dan *Doc2vec*, serta pengukuran *Cosine Similarity*. Hasil penelitian menunjukkan bahwa teknik *Word2vec* menghasilkan nilai *Cosine Similarity* yang lebih tinggi dibandingkan dengan *Doc2vec* untuk pasangan jurnal yang sama, dapat dilihat pada pasangan jurnal J02 dengan J14 memiliki nilai *Cosine Similarity* 0.892 pada teknik *Word2vec*, sedangkan pada *Doc2vec* nilainya 0.434. Hal ini menandakan bahwa hasil teknik *Word2vec* terbukti lebih efektif dalam menangkap kemiripan semantik antara jurnal-jurnal dibandingkan dengan teknik *Doc2vec*.

Kata kunci: *Word2vec*, *Doc2vec*, *Cosine Similarity*, Kemiripan Dokumen

A COMPARATIVE ANALYSIS OF WORD2VEC AND DOC2VEC TECHNIQUES IN DOCUMENT SIMILARITY USING COSINE SIMILARITY

Abstract

The digital era has made access to many online documents easier and faster, but it has also created complex challenges in information management and analysis. One of the main challenges is measuring the similarity between documents, which is crucial for various applications such as plagiarism detection. In response to this challenge, many techniques can be used to represent documents as vectors to measure document similarity. In this research, Word2vec and Doc2vec techniques are used to represent documents as vectors, and Cosine Similarity is used to measure document similarity. The research objects are abstract paragraphs from 20 scientific journals on data mining published between 2020 and 2024 from Gunadarma University's E-Journal. The research methodology includes data collection, text mining, text pre-processing, Word2vec and Doc2vec techniques implementations, and Cosine Similarity measurement. The results show that the Word2vec technique produces higher Cosine Similarity values compared to Doc2vec for the same journal pairs, as seen in the journal pair J02 and J14 having a Cosine Similarity value of 0.892 using the Word2vec technique, while with Doc2vec the value is 0.434. This indicates that the Word2vec technique proves to be more effective in capturing semantic similarities between journals compared to the Doc2vec technique.

Keywords: *Word2vec, Doc2vec, Cosine Similarity, Document Similarity*

1. PENDAHULUAN

Di era digital yang berkembang pesat, mengakses dokumen dalam jumlah besar secara online menjadi lebih mudah dan cepat. Fenomena ini didorong oleh kemajuan teknologi internet dan pertumbuhan platform online yang memungkinkan pengguna mengakses berbagai jenis dokumen dengan cepat dan efisien, seperti jurnal ilmiah

akademik, artikel dan laporan. Namun, seiring dengan meningkatnya ketersediaan informasi, tantangan dalam mengelola, menganalisis, dan mengekstraksi pengetahuan dari dokumen-dokumen tersebut menjadi semakin kompleks.

Salah satu tantangan terbesar dalam pengelolaan informasi adalah menemukan dokumen yang mirip dengan dokumen tertentu. Konsep kemiripan dokumen penting dalam berbagai aplikasi seperti dalam mendeteksi plagiarisme. Plagiarisme merupakan tindakan mengambil karangan, pendapat, atau materi lain dari sumber lain dan mengakuinya sebagai karya atau pendapat pribadi. Plagiarisme dokumen dapat disebut sebagai plagiarisme ringan apabila nilai kemiripan kurang dari 30%, plagiarisme sedang nilai kemiripan antara 30 % - 70 % sementara dikatakan plagiarisme berat atau total nilai kemiripan lebih dari 70 % (Sastroasmoro, 2007).

Untuk mengukur kemiripan antar dokumen, berbagai pendekatan berdasarkan representasi vektor dokumen telah dikembangkan. Dengan menerapkan representasi vektor tersebut maka kata dalam suatu teks ataupun dokumen dapat diubah dan direpresentasi kedalam vektor, sehingga dapat dilakukan perbandingan dan analisis yang lebih efektif. Dua teknik yang umum digunakan dalam konteks ini adalah *Word2vec* dan *Doc2vec*.

Word2vec merupakan teknik yang digunakan untuk merepresentasikan vektor yang berasal dari kata dalam suatu teks. Word2vec diperkenalkan oleh Mikolov et al pada tahun 2013, yang terdiri dari model Continous Bag of Words (CBOW) dan model Skip-gram. Model CBOW memprediksi kata saat ini berdasarkan konteks, sedangkan Skip-gram menggunakan kata saat ini sebagai referensi untuk memprediksi kata di sekitarnya (Mikolov et al., 2013).

Doc2vec dikenal juga *Paragraph Vector*, merupakan teknik yang digunakan untuk merepresentasikan vektor yang berasal dari dokumen. Teknik ini pengembangkan dari teknik *Word2vec* (Le & Mikolov, 2014). *Doc2vec* terdiri dari model *Paragraph Vector Distributed Bag of Words* (PV-DBOW) dan model *Paragraph Vector Distributed Memory* (PV-DM).

Hasil representasi vektor dengan *Word2vec* maupun *Doc2vec* akan dilakukan pengukuran mengunakan *cosine similarity* untuk mendapatkan nilai kemiripan dokumen. Diungkapkan oleh (Riyani et al, 2019) bahwa *cosine similarity* memiliki kelebihan dalam melakukan pengukuran, *cosine similarity* tidak dipengaruhi oleh panjang dokumen dan memiliki tingkat akurasi yang tinggi.

Dalam upaya memahami dan meningkatkan pemahaman mengenai pengukuran tingkat kemiripan dokumen, banyak teknik dalam merepresentasikan teks kedalam vektor yang digunakan oleh Peneliti. Pada penelitian terdahulu yang dilakukan (Chen & Sokolova, 2021), Peneliti melakukan analisis terhadap objek data obesitas dan artikel sains yang berbahasa Inggris dimana teknik yang digunakan *Word2vec* dan *Doc2vec*.

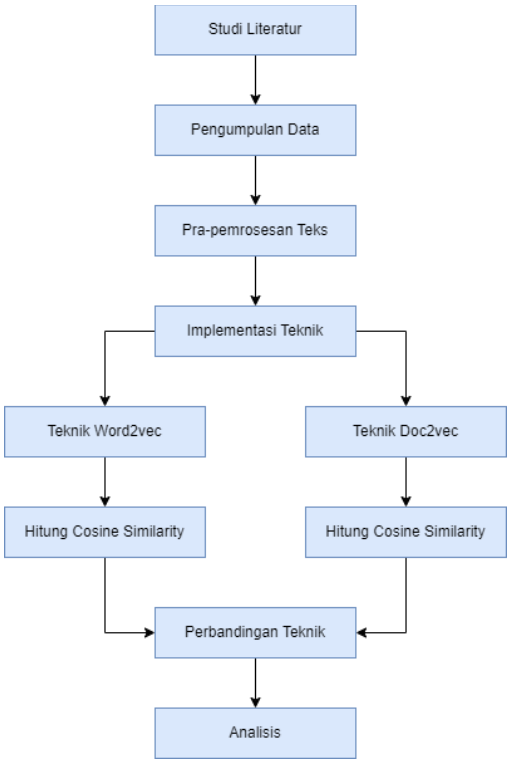
Dalam penelitian yang dilakukan oleh Alshammeri et al (2021), fokus utama peneliti adalah melakukan penelitian dengan objek data ayat-ayat dalam Alquran berbahasa Arab untuk mendeteksi kemiripan ayat-ayat yang ada dalam Alquran. Teknik *Doc2vec* digunakan untuk merepresentasikan ayat Alquran menjadi vektor dan dalam pengukuran kemiripan mengunakan *cosine similarity*.

Sedangkan Nurdin et al (2020), melakukan penelitian dengan fokus utama adalah untuk melakukan perbandingan kinerja dari teknik *Word2vec*, *Glove* dan *Fasttext* pada klasifikasi teks. Objek data yang digunakan sebagai sumber penelitian berupa 20 *newsgroup* dan *reuters newswire topic classification* dan menggunakan bahasa Inggris.

Teknik *Word2vec*, *Doc2vec*, *Glove*, dan *Fasttext* telah menjadi inti dari berbagai penelitian di berbagai bidang, termasuk *Natural Language Procissing* (NLP) dan analisis teks. Dalam penelitian ini, peneliti akan fokus dalam menganalisis perbandingan teknik *Word2vec* dan *Doc2vec* untuk mengukur kemiripan dokumen jurnal ilmiah berbahasa Indonesia mengunakan *cosine similary*. Sehingga didapatkan hasil dengan nilai efektivitas yang baik antara teknik *Word2vec* dengan *Doc2vec*.

2. METODE PENELITIAN

Pada kegiatan penelitian ini dilakukan dalam beberapa tahap yang disajikan pada Gambar 1.



Gambar 1 Alur Tahap Penelitian

2.1. Studi Literatur

Tahap ini melibatkan pencarian dan pengumpulan referensi teori yang relevan dengan penelitian, terutama terkait teknik *Word2vec* dan *Doc2vec* untuk mengukur kemiripan dokumen menggunakan *cosine similarity*. Sumber referensi mencakup buku teks, artikel jurnal ilmiah, dan sumber terpercaya dari internet, yang memberikan landasan teori, wawasan aplikasi, dan perkembangan terbaru dalam bidang ini. Google Colab, dengan kemudahan penggunaan dan sumber daya komputasi yang kuat, memungkinkan peneliti untuk fokus pada pengembangan dan eksperimen model tanpa khawatir tentang konfigurasi lingkungan (Carneiro et al., 2018). Python, bahasa pemrograman tingkat tinggi dengan sintaksis sederhana dan perpustakaan yang luas, banyak digunakan dalam analisis data dan kecerdasan buatan, menjadi pilihan utama untuk implementasi teknik-teknik ini (Rahman et al., 2023). Dalam penelitian ini, beberapa library Python yang digunakan meliputi *Natural Language Toolkit* (NLTK) untuk pemrosesan bahasa alami, Gensim untuk implementasi *Word2vec* dan *Doc2vec*, *NumPy* dan *Pandas* untuk manipulasi data, *scikit-learn* untuk implementasi *cosine similarity*, *pdfplumber* untuk ekstraksi teks dari file *Portable Document Format* (PDF), dan *Sastrawi* untuk *stemming* bahasa Indonesia. Pengumpulan referensi yang beragam dan komprehensif, serta penggunaan *library* ini, bertujuan untuk membangun dasar teori yang kuat dan mendukung analisis mendalam dalam perbandingan *Word2vec* dan *Doc2vec* untuk mengukur kemiripan dokumen.

2.2. Pengumpulan Data

Tahap ini dilakukan untuk memperoleh data yang akan digunakan dalam penelitian analisis perbandingan teknik *Word2vec* dan *Doc2vec* dengan menggunakan metode *cosine similarity*. Langkah awal untuk mendapatkan data dimulai dengan mengakses serta mengunduh jurnal-jurnal ilmiah yang relevan dari portal E-Journal Universitas Gunadarma. Fokus utama adalah jurnal ilmiah dengan tema data *mining*, yang merupakan bagian dari bidang informatika dan komputer, yang diterbitkan antara tahun 2020 hingga tahun 2024. Data yang sudah terkumpul dalam format PDF akan melalui proses *text mining* untuk mendapatkan paragraf abstrak, yang selanjutnya akan diolah pada tahap berikutnya.

Berikut tampilan website e-journal Universitas Gunadarma yang diakses untuk pengumpulan data jurnal, dapat dilihat pada Gambar 2.

Setelah pengumpulan data selesai, dilakukan pendataan jurnal yang diperoleh. Berikut daftar jurnal dapat dilihat pada Tabel 1.

Gambar 2 Tampilan Website E-jurnal Universitas Gunadarma
Sumber :
https://ejournal.gunadarma.ac.id/index.php/infokom/index

Tabel 1 Daftar Jurnal

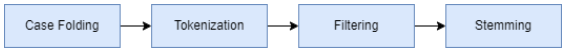
Id	Judul	Tahun
J01	Analisis Sentimen Dan Klasifikasi Tweets Berbahasa Indonesia Terhadap Transportasi Umum MRT Jakarta Menggunakan Naïve Bayes Classifier	2020
J02	Analisis Sentimen Pengguna Twitter Terhadap Dompet Elektronik Dengan Metode Lexicon Based Dan K-Nearest Neighbor	2020
J03	Analisis Sentimen Terhadap Pelayanan KRL Commuterline Berdasarkan Data Twitter Menggunakan Algortima Bernoulli Naive Bayes	2020
J04	Metode Decision Tree Untuk Klasifikasi Hasil Seleksi Kompetensi Dasar Pada CPNS 2019 Di Arsip Nasional Republik Indonesia	2020
J05	Analisis Peramalan Tingkat Kemiskinan Di Indonesia Dengan Model Arima	2021
J06	Implementasi Algoritma C4.5 Untuk Klasifikasi Penyakit Infeksi Saluran Kemih Berbasis Web	2021
J07	Implementasi Algoritma Klasifikasi Support Vector Machine Untuk Analisa Sentimen Pengguna Twitter Terhadap Kebijakan PSBB	2021
J08	Implementasi Metode Lexicon Base Untuk Analisis Sentimen Kebijakan Pemerintah Dalam Pencegahan Penyebaran Virus Corona Covid-19 Pada Twitter	2021
J09	Analisis Kredit Calon Debitur Menggunakan Metode Fuzzy Tsukamoto	2022
J10	Klasifikasi Topik Tweet Mengenai Covid Menggunakan Metode Multinomial Naïve Bayes Dengan Pembobotan	2022
J11	Identifikasi Topik Artikel Berita Menggunakan Topic Modelling Dengan Latent Dirichlet Allocation	2022
J12	Clustering Relationship Berdasarkan Bobot Pembentuk Social Trust Network Untuk Sistem Rekomendasi Pada Media Sosial Instagram	2022
J13	Analisis Sentimen Review Pengguna Aplikasi Depok Single Window Di Google Play Menggunakan Algoritma Support Vector Machine	2023
J14	Analisis Sentimen Terhadap Twit Maxim Pada Twitter Menggunakan R Programming Dan K Nearest Neighbors	2023
J15	Analisis Sentimen Warga Twitter Terhadap Game Shopee Cocoki Dengan Metode Naive Bayes Classifier	2023
J16	Pengelompokan Wilayah Kasus Balita Stunting Di Indonesia Menggunakan Algoritma K-Means	2023
J17	Analisis Minat Beli Produk Fashion Menggunakan Algoritma Fp-Growth (Frequent Patten Growth)	2024

Id	Judul	Tahun
J18	Implementasi Data Mining Algoritma Decision Tree Untuk Klasifikasi Status Gizi Balita Di Kecamatan Ciledug	2024
J19	Prediksi Tingkat Kualitas Udara Dengan Pendekatan Algoritma K-Nearest Neighbor	2024
J20	Prediksi Kepuasan Pelanggan Hotel Studi Perbandingan Algoritma Decision Tree Dan Knearest Neighbor	2024

2.3.Pra-pemrosesan Teks

Pra-pemrosesan Teks merupakan langkah penting dalam penelitian ini, dengan Pra-pemrosesan Teks dapat meningkatkan nilai akurasi dalam menghitung nilai *cosine similarity* (Budiman & Widjaja, 2020). Tahap ini bertujuan untuk memastikan data teks siap untuk direpresentasikan menjadi vektor mengunakan teknik *Word2vec* dan *Doc2vec* selanjutnya mengukur cosine similarity. Tahap Pra-pemrosesan Data memiliki langkah-langkah yang beraneka ragam, hal ini bergantung pada dataset yang digunakan pada kegiatan penelitian (HaCohen-Kerner et al, 2020). Disampaikan oleh Cahyono (2019), langkah-langkah pra-pemrosesan data terdiri atas *tokenizing*, *filtering* dan *stemming* dengan dataset dokumen tesis dalam bahasa Indonesia. Sedangkan pada penelitian yang dilakukan oleh (Cahyani & Patasik, 2021) memiliki langkah *case folding*, *filtering*, normalisasi, *stopwords* dan *stemming* dengan dataset berasal *tweets* pengguna Transjakarta dan *Commuterline*.

Penelitian ini mencakup beberapa langkah dalam pra-pemrosesan teks, yang alurnya dapat dilihat pada Gambar 3.



Gambar 3 Alur Pra-pemrosesan Teks

Alur langkah dapat dijelaskan sebagai berikut:

1. *Case Folding*

Case Folding merupakan langkah pertama pra-pemrosesan teks. Pada langkah ini mentransformasikan semua huruf kapital dari mulai ‘a’ sampai dengan ‘z’ menjadi huruf kecil (Fataruba, 2018). Konversi kedalam huruf kecil dalam case folding dapat mengurangi dimensi data sambil mempertahankan informasi semantik serta meningkatkan kekuatan statistik (Hickman et al., 2022).

2. *Tokenization*

Tokenization merupakan salah satu teknik yang sangat penting dalam tahap pra-pemrosesan teks pada aplikasi NLP. Teknik ini berfungsi untuk memisahkan teks menjadi unit-unit terkecil yang memiliki makna, yang disebut sebagai token (Song et al., 2020). Selain itu, melalui tahap ini, angka dan tanda baca yang terdapat pada teks abstrak akan dihapus.

3. *Filtering*

Filtering adalah proses pemilihan kata-kata kunci dari hasil *tokenization* dengan menghilangkan *stopwords*. Penghapusan *stopwords* penting karena meskipun kata-kata ini meningkatkan volume teks, mereka tidak berkontribusi signifikan terhadap makna dokumen. Manfaat dari penghapusan stopwords meliputi pengurangan ukuran indeks, percepatan waktu pemrosesan, dan penurunan noise dalam analisis (Suyanto et al, 2023). Identifikasi *stopwords* umumnya menggunakan *stop list*, yaitu daftar kata-kata yang dianggap sebagai *stopwords* (Parwita, 2020). Contoh *stopwords* dalam bahasa Indonesia termasuk "yang", "dan", "itu", "tidak", "dengan", "dari", "untuk", "dalam", "ini", "akan", dan lainnya (Tala, 2003).

4. *Stemming*

Stemming adalah langkah terakhir dari pra-pemrosesan teks, langkah ini memiliki bertujuan untuk mereduksi kata menjadi bentuk kata dasar yang prosesnya melibatkan penghapusan imbuhan yang melekat pada setiap kata hal ini bisa berjalan dengan memanfaatkan *library Sastrawi* (Hasanah & Mutiara, 2019). Teknik ini masih dianggap efektif dalam meningkatkan kualitas hasil aplikasi *text mining* (Aggarwal, 2015). Untuk Bahasa Indonesia, dua algoritma *stemming* utama adalah algoritma Nazief dan Adriani yang menggunakan pendekatan confix stripping dengan pemindaian kamus, serta algoritma Tala yang menggunakan pendekatan berbasis aturan untuk memproses awalan, akhiran, dan kombinasinya. Meskipun Bahasa Indonesia memiliki sisipan, penggunaannya jarang sehingga sering diabaikan dalam proses *stemming* (Parwita, 2020).

2.4. Implementasi Teknik

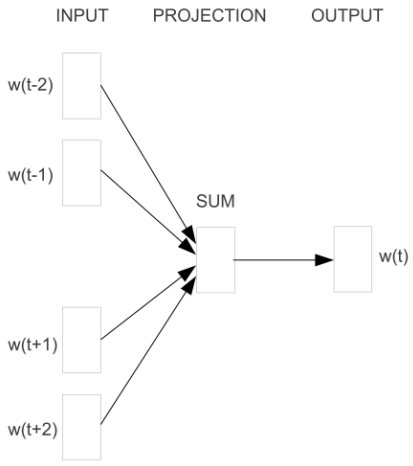
Tahap implementasi teknik dalam penelitian ini dilakukan setelah pra-pemrosesan teks selesai. Pada tahap ini mengunakan dua teknik utama, yaitu *Word2vec* dan *Doc2vec* yang akan diterapkan untuk menghasilkan representasi vektor dari teks yang telah dipra-pemroses. Selain itu, penghitungan *cosine similarity* akan digunakan untuk mengukur kemiripan antar-dokumen berdasarkan representasi vektor tersebut. Adapun penjelasan langkah dari implementasi teknik, sebagai berikut:

Teknik *Word2vec*

Pada langkah teknik *Word2vec*, hasil teks abstrak yang sudah dilakukan pra-pemrosesan akan direprensentasikan menjadi vektor. *Word2vec* menggunakan jaringan saraf tiruan untuk mempelajari representasi vektor kata dari korpus besar. Kata-kata direpresentasikan sebagai *one-hot encoding*, dan jaringan dilatih dengan algoritma *backpropagation* (Rong, 2014).

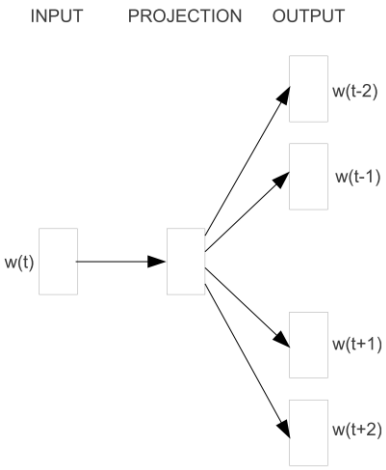
Terdapat dua arsitektur utama dalam *Word2vec*, yaitu:

1. *Continuous Bag-of-Words* (CBOW); Pada arsitektur CBOW, model memprediksi kata target berdasarkan konteks kata-kata di sekitarnya. Berikut arsitektur CBOW dapat dilihat pada Gambar 4.



Gambar 4 Arsitektur CBOW
Sumber : Mikolov et al (2013)

2. *Skip-gram*; Pada arsitektur *Skip-Gram*, model memprediksi kata-kata konteks berdasarkan kata target. Berikut arsitektur *Skip-Gram* dapat dilihat pada gambar 5.



Gambar 5 Arsitektur Skip-grarm
Sumber : Mikolov et al (2013)

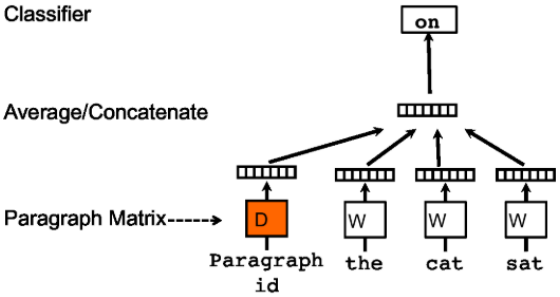
Pada penelitian ini, Skip-gram dipilih karena model ini lebih baik dalam merepresentasikan teks menjadi vektor atas kata yang jarang muncul dalam teks dibandingkan dengan CBOW (Cahyani & Patasik, 2021).

Teknik *Doc2vec*

Teknik *Doc2vec* merupakan pengambangan dari teknik *Word2vec*, pada teknik ini mengubah dokumen menjadi vektor yang kompak. *Doc2vec* telah terbukti efektif dalam beragam aplikasi, seperti analisis sentimen, identifikasi topik, dan sistem rekomendasi (Lau & Baldwin, 2016). Dengan kemampuannya untuk merepresentasikan dokumen sebagai vektor yang padat, *Doc2vec* menjadi sebuah alat yang berharga dalam bidang NLP dan pembelajaran mesin.

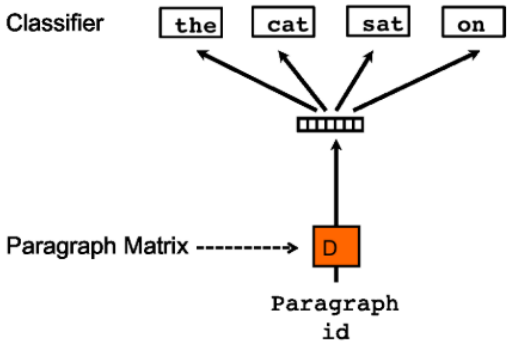
Pada *Doc2vec* terdapat 2 arsitektur utama, yaitu:

1. *Paragraph Vector Distributed Memory* (PV-DM); Pada arsitektur PV-DM, vektor dokumen dipelajari bersama dengan vektor kata-kata, dan vektor dokumen digunakan sebagai memori tambahan untuk memprediksi kata-kata dalam dokumen. Arsitektur ini merupakan pengembangan dari arsitektur CBOW. Berikut arsitektur PV-DM dapat dilihat pada Gambar 6.



Gambar 6 Arsitektur PV-DM
Sumber : Mikolov & Le (2014)

2. *Paragraph Vector Distributed Bag of Words* (PV-DBOW); Pada arsitektur PV-DBOW, vektor dokumen dipelajari terpisah dari vektor kata-kata dan digunakan untuk memprediksi kata-kata dalam dokumen secara langsung. Arsitektur ini merupakan pengambangan dari arsitektur Skip-Gram. Berikut alur proses DBOW dapat dilihat pada Gambar 7.



Gambar 7 Arsitektur PV-DBOW
Sumber : Mikolov & Le (2014)

Dalam Penelitian ini PV-DBOW dipilih sebagai model untuk merepresentasikan teks menjadi vektor, dalam merepresentasikan vektor tidak mempertimbangkan urutan kata dalam teks. Dalam penelitian Amalia et al (2020), model PV-DBOW memiliki kinerja lebih baik dibandingkan model PV-DM.

Cosine Similarity

Pada langkah ini, vektor yang dihasilkan oleh kedua model akan dilakukan penghitungan dengan

cosine similarity. Tujuannya adalah untuk mendapatkan nilai kemiripan dari setiap dokumen yang dibandingkan. Pengukurannya dilakukan atas vektor dengan jenis model yang sama. Cara kerjanya adalah dengan menghitung nilai kosinus dari sudut yang terbentuk antara dua vektor yang mewakili teks atau dokumen yang dibandingkan (Jurafsky & Martin, 2023).

Rumus untuk menghitung *Cosine Similarity* antara dua vektor teks atau dokumen adalah sebagai berikut (Wahyuni et al, 2017):

$$cosine\ similarity(A,B) = \frac{(A \cdot B)}{|A||B|} \quad (1)$$

A	:	Representasi vektor teks atau dokumen A
B	:	Representasi vektor teks atau dokumen B
$(A \cdot B)$	:	Hasil perkalian *dot product* antara vektor A dan vektor B
$\|A\|$	:	Panjang vektor A
$\|B\|$	:	Panjang vektor B
$\|A\|\|B\|$	:	*Cross product* antara panjang vektor A dan panjang vektor B

Hasil pengukuran Cosine Similarity berupa nilai antara 0 sampai 1. Semakin dekat nilai tersebut ke 1, semakin mirip kedua teks atau dokumen yang dibandingkan. Sebaliknya, semakin dekat nilai tersebut ke 0, semakin berbeda kedua teks atau dokumen tersebut (Jurafsky & Martin, 2023).

2.5. Perbandingan Teknik

Tahap perbandingan teknik akan dilakukan setelah implementasi teknik *Word2vec* dan *Doc2vec* serta pengukuran cosine similarity selesai. Pada tahap ini, hasil pengukuran kemiripan cosine similarity dari representasi vektor *Word2vec* dibandingkan dengan hasil pengukuran kemiripan *cosine similarity* dari representasi vektor *Doc2vec*. Hasil perbandingan ini akan diproses lebih lanjut pada tahap berikutnya yaitu tahap analisis

2.6. Analisis

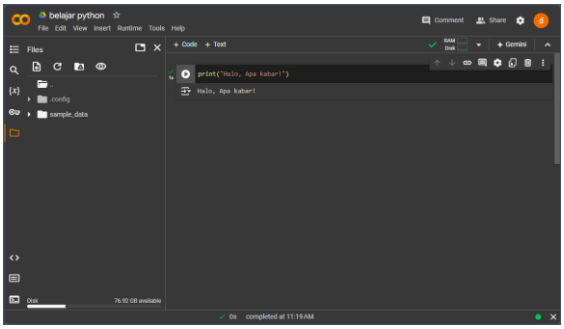
Pada tahap terakhir adalah kegiatan untuk menganalisis hasil perbandingan antara teknik *Word2vec* dan *Doc2vec* dengan menggunakan metode *cosine similarity* pada paragraf abstrak dari data jurnal ilmiah dengan fokus tema data mining yang bersumber dari portal E-Journal Universitas Gunadarma. Analisis data ini bertujuan untuk memperoleh informasi mengenai teknik mana yang lebih efektif dalam mengukur kemiripan antar paragraf abstrak dokumen jurnal.

3. HASIL DAN PEMBAHASAN

3.1 *Tools* Penelitian

Google Colab digunakan dalam penelitian ini sebagai platform utama untuk melakukan pemrosesan data, pelatihan model, dan pengukuran kemiripan dokumen. Google Colab adalah layanan *cloud* berbasis Jupyter Notebook yang disediakan oleh Google, yang memungkinkan pengguna untuk menulis dan menjalankan kode program Python secara interaktif melalui *browser web*.

Google Colab dapat diakses pada alamat https://colab.research.google.com/ dapat dilihat pada Gambar 8.



Gambar 8 Tampilan Google Colab

3.2. Proses Pengukuran Kemiripan Dokumen Jurnal

Proses ini melibatkan beberapa tahapan, yaitu *Text Mining* untuk ekstraksi teks abstrak dari file PDF, pra-pemrosesan teks untuk menghasilkan kata-kata penting, penerapan teknik *Word2vec* dan *Doc2vec* yang menghasilkan vektor, serta pengukuran *cosine similarity* yang ditampilkan dalam bentuk tabel dan daftar lima terbaik. Alur proses pengukuran kemiripan dokumen jurnal dapat dilihat pada Gambar 9.

Text Mining Dokumen Jurnal

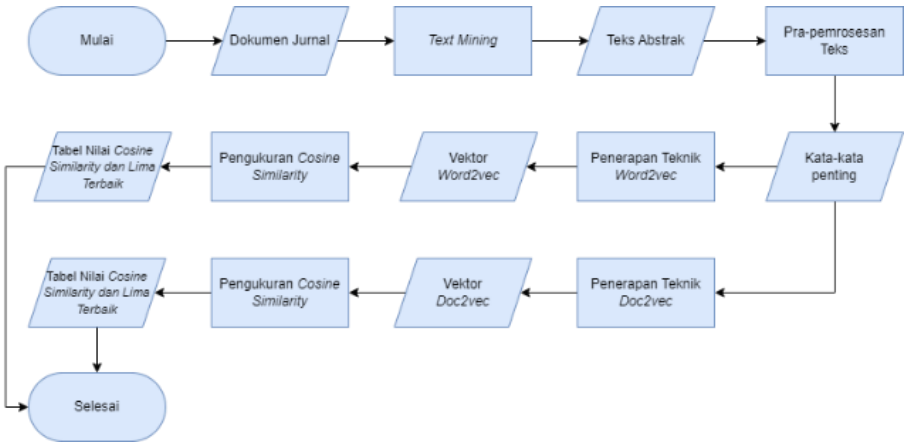
Proses *text mining* bertujuan untuk mengekstraksi teks abstrak dari file-file PDF jurnal yang telah dikumpulkan. Pada proses ini menggunakan library *Pdfplumber*. Proses ini melibatkan beberapa langkah sebagai berikut:

1. Mengakses file PDF dari Google drive;
2. Melakukan iterasi pada setiap file PDF;
3. Menyimpan hasil ekstrasi teks abstrak.

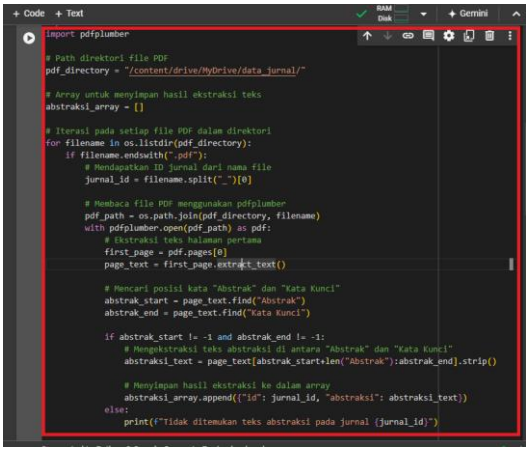
Pada proses ini, hasil teks abstrak disimpan pada variabel *abstraksi_array*. Kode program dapat dilihat pada Gambar 10 dan contoh teks abstrak pada Gambar 11.

Pra-pemrosesan Teks Abstrak

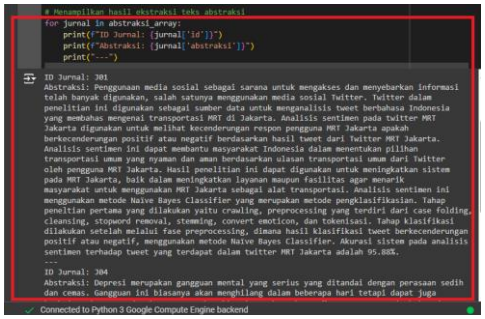
Pada tahap ini, dilakukan pra-pemrosesan teks terhadap hasil ekstraksi teks abstrak yang telah disimpan dalam variabel array *abstraksi_array*.



Gambar 9 Alur Proses Pengukuran Kemiripan Dokumen Jurnal



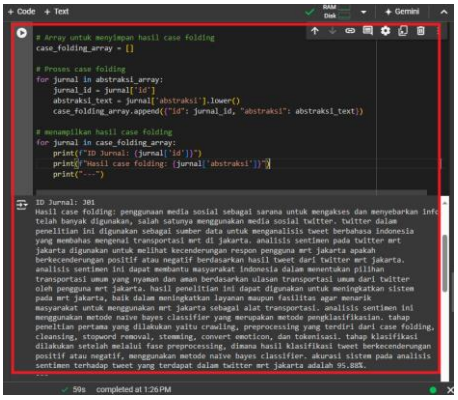
Gambar 10 Kode Program *Text Mining*



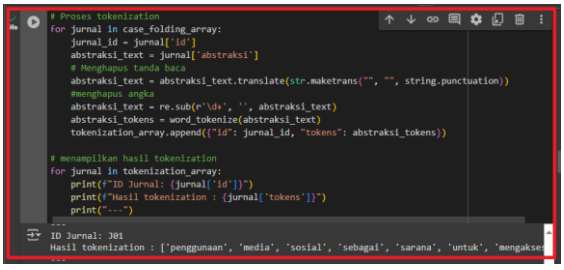
Gambar 11 Contoh Teks Abstrak hasil *Text Mining*

Proses pra-pemrosesan teks melibatkan beberapa langkah sebagai berikut:

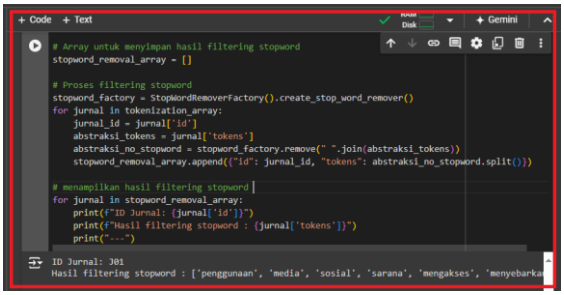
1. *Case Folding*; proses ini menggunakan fungsi *lower()*, kode program dan hasil dapat dilihat pada Gambar 12.
2. *Tokenization*; tahap ini menggunakan *library* NLTK dengan fungsi *word_tokenize()*. Kode program dan hasil dapat dilihat pada Gambar 13.
3. *Filtering*; tahap ini menggunakan *library sastrawi* dan fungsi yang digunakan *StopWordRemoverFactory*(). Kode program dan hasil dapat dilihat pada Gambar 14.



Gambar 12 Tampilan Kode Program dan Hasil *Case Folding*

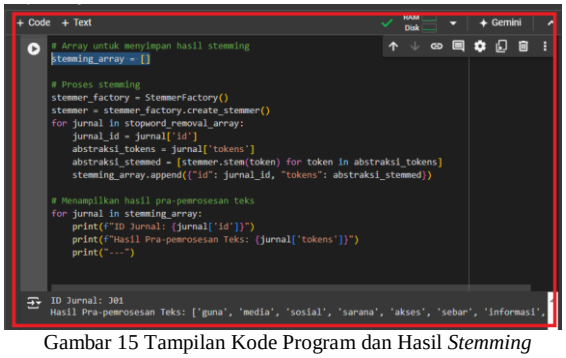


Gambar 13 Tampilan Kode Program dan Hasil *Tokenization*



Gambar 14 Tampilan Kode Program dan Hasil *Filtering*

4. *Stemming*; *library* yang digunakan sama seperti proses *filtering*, dan fungsi yang digunakan *StemmerFactory()*. Kode program dan hasil seperti pada Gambar 15.



Gambar 15 Tampilan Kode Program dan Hasil *Stemming*

Penerapan Teknik *Word2vec*

Word2vec diterapkan untuk membuat representasi vektor kata-kata dalam dokumen jurnal menggunakan arsitektur *Skip-gram*. Teknik ini merepresentasikan setiap kata sebagai vektor dalam ruang dimensi tertentu. *Skip-gram*, sebagai pendekatan *Word2vec*, bertujuan memprediksi kata-kata konteks di sekitar kata target dalam jendela konteks tertentu, dengan setiap kata sebagai input.

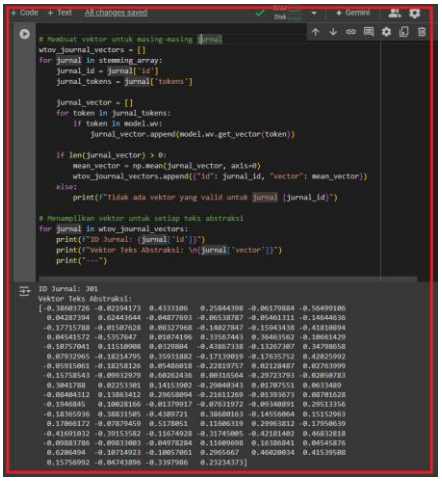
Langkah-langkah penerapan teknik *Word2vec* dengan arsitektur *Skip-gram* adalah sebagai berikut:

1. Mempersiapkan data pelatihan; data pelatihan menggunakan teks abstrak hasil pra-pemrosesan teks, data didapat dari vaiabel *stemming_array*.
2. Membuat model *word2vec*; menggunakan *library gensim*, parameter menggunakan nilai *default* hanya *min_count* yang diberi nilai 1, sementara untuk menggunakan arsitektur *skip-gram* nilai *sg* = 1.
3. Melatih model *Word2vec*; dalam melatih model fungsi yang digunakan *build_vocab()* dan *train()*, Pada penelitian ini parameter epoch adalah 300, hal ini mempertimbangkan jumlah data pelatihan relatif kecil, serupa dengan pendekatan yang dilakukan oleh Amalia et al (2020).
4. Membuat vektor kata;
5. Membuat vektor teks abstrak; dilakukan dengan menghitung rata-rata (*mean*) dari vektor kata yang terdapat dalam setiap teks abstrak, sehingga terbentuk vektor kalimat (Kenter et al, 2016).

Kode program dan hasil vektor teks abstrak dapat dilihat pada Gambar 16.

Penerapan Teknik *Doc2vec*

Doc2vec merupakan perluasan dari *Word2vec*, diterapkan untuk membuat representasi vektor dari teks abstrak jurnal. Teknik ini memungkinkan pembelajaran representasi vektor untuk dokumen atau paragraf, tidak hanya kata-kata individual. Penelitian ini menggunakan arsitektur PV-DBOW, yang bertujuan memprediksi kata-kata dalam dokumen berdasarkan representasi vektor dokumen tersebut.



Gambar 16 Tampilan Kode Program dan Hasil Vektor *Skip-gram*

Langkah-langkah pernerapan teknik *Doc2vec* dengan arsitektur PV-DBOW adalah sebagai berikut:

1. Mempersiapkan data pelatihan; menggunakan teks abstrak yang disimpan pada variabel *stemming_array*.
2. Membuat model *Doc2vec*; menggunakan *library gensim*, parameter menggunakan nilai *default*, *min_count* = 1, untuk arsitektur PV-DBOW nilai *dm* = 0.
3. Melatih model *Doc2vec*; menggunakan fungsi *build_vocab()* dan *train()*, nilai *epoch* = 300.
4. Membuat vektor teks abstrak; menggunakan fungsi *infer_vector()*.

Kode program dan hasil vektor PV-DBOW seperti pada Gambar 17.

Pengukuran *Cosine Similarity* atas Vektor *Word2vec*

Pengukuran *Cosine Similarity* dilakukan antara vektor teks abstrak yang dihasilkan oleh teknik *Word2vec*. Langkah-langkah meliputi:

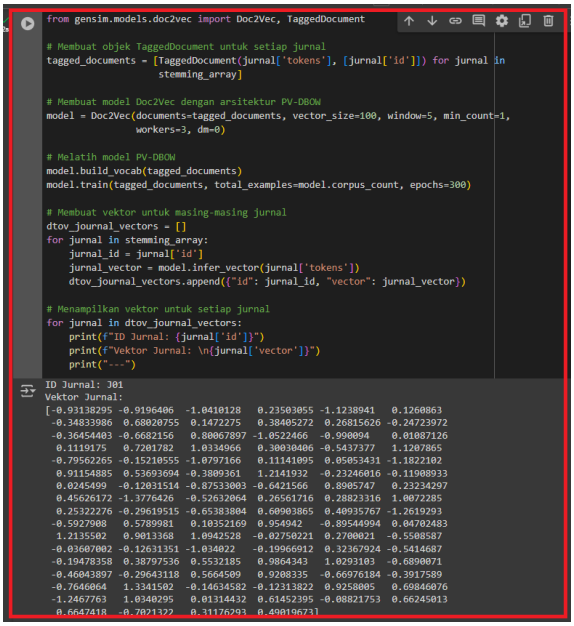
1. Mengukur *Cosine Similarity;* menggunakan *library scikit-learn* dan fungsi *cosine_similarity*.
2. Membuat tabel hasil pengukuran *Cosine Similarity*; tabel dibuat menggunakan *library pandas* dari hasil pengukuran *Cosine Similarity.*
3. Membuat daftar lima tertinggi;

Kode program tabel pengukuran dan daftar lima tertinggi dapat dilihat pada Gambar 18 dan Gambar 19.

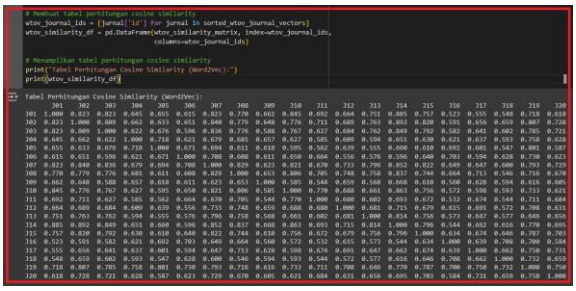
Pengukuran *Cosine Similarity* Atas Vektor *Doc2vec*

Pengukuran *cosine similarity* dilakukan antara vektor teks abstrak jurnal yang dihasilkan dari teknik *Doc2vec*. Proses ini serupa dengan pengukuran pada *Word2vec*, namun menggunakan data vektor teks abstrak jurnal dari *Doc2vec*. Kode program untuk pengukuran ini mirip dengan yang digunakan untuk *Word2vec*, dengan penyesuaian pada nama variabel yang digunakan. Detail kode program, hasil

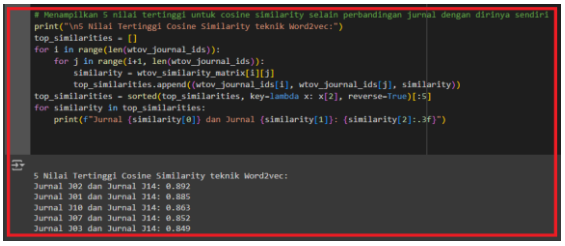
pengukuran serta daftar lima terbaik disajikan dalam Gambar 20 dan Gambar 21.



Gambar 17 Tampilan Kode Program dan Hasil Vektor PV-DBOW



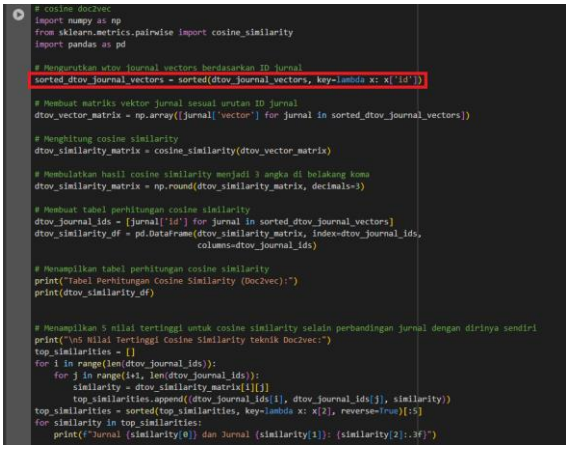
Gambar 18 Tampilan Kode Program dan hasil Tabel *Word2vec*



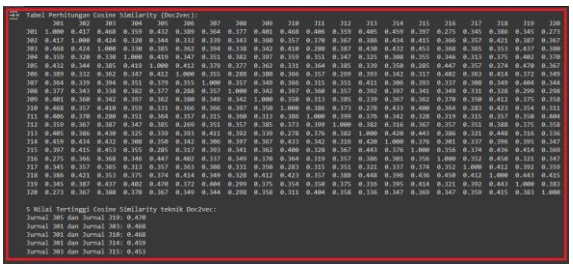
Gambar 19 Tampilan Kode Program dan Daftar lima tertinggi *Word2vec*

3.3. Perbandingan Hasil Pengukuran dan Analisis

Setelah melakukan pengukuran *cosine similarity* menggunakan teknik *Word2vec* dan *Doc2vec*, langkah selanjutnya adalah membandingkan hasil pengukuran tersebut untuk melihat perbedaan antara kedua teknik dalam menentukan kemiripan antar teks abstrak jurnal. Berikut tabel hasil pengukuran *cosine similarity* dari teknik *Word2vec* dan *Doc2vec* pada Tabel 2 dan Tabel 3.



Gambar 20 Tampilan Detail Kode Program



Gambar 21 Tampilan Tabel dan Daftar Lima Tertinggi *Doc2vec*

Berdasarkan hasil pengukuran cosine similarity, diperoleh lima nilai tertinggi untuk masing-masing teknik. Berikut adalah ringkasan perbandingan hasil pengukuran *cosine similarity* antara teknik *Word2vec* dan *Doc2vec* pada Tabel 4 dan Tabel 5.

Tabel 4 Hasil Ringkasan Perbandingan Lima Tertinggi Teknik *Word2vec*

Id	Teknik *Word2vec*	Teknik *Doc2vec*
J02 dengan J14	0.892	0.434
J01 dengan J14	0.885	0.459
J10 dengan J14	0.863	0.433
J07 dengan J14	0.852	0.306
J03 dengan J14	0.849	0.432

Tabel 5 Hasil Ringkasan Perbandingan Lima Tertinggi Teknik *Doc2vec*

Id	Teknik *Doc2vec*	Teknik *Word2vec*
J05 dengan J19	0.470	0.801
J01 dengan J03	0.468	0.823
J01 dengan J10	0.468	0.845
J01 dengan J14	0.459	0.885
J03 dengan J15	0.453	0.792

Hasil perbandingan pada tabel 4 dan Tabel 5 menunjukkan bahwa nilai *cosine similarity* yang dihasilkan oleh teknik *Word2vec* cenderung lebih tinggi dibandingkan dengan *Doc2vec* untuk pasangan jurnal yang sama. Perbedaan ini disebabkan oleh pendekatan yang berbeda dalam merepresentasikan dokumen, di mana *Word2vec* berfokus pada representasi kata-kata individual, sedangkan *Doc2vec* berfokus pada representasi dokumen secara keseluruhan.

Tabel 2 Hasil Pengukuran *Cosine Similarity* Teknik *Word2vec*

Id	J01	J02	J03	…	J07	…	J10	…	J20
J01	1.000	0.823	0.823	…	0.823	…	0.845	…	0.618
J02	0.823	1.000	0.809	…	0.840	…	0.776	…	0.728
J03	0.823	0.809	1.000	…	0.836	…	0.767	…	0.721
J04	0.645	0.662	0.622	…	0.679	…	0.627	…	0.628
…	…	…	…	…	…	…	…	…	…
J13	0.751	0.763	0.762	…	0.796	…	0.661	…	0.656
J14	0.885	0.892	0.849	…	0.852	…	0.863	…	0.695
J15	0.757	0.820	0.792	…	0.822	…	0.756	…	0.703
…	…	…	…	…	…	…	…	…	…
J18	0.548	0.659	0.602	…	0.600	…	0.593	…	0.659
J19	0.718	0.807	0.785	…	0.793	…	0.733	…	0.750
J20	0.618	0.728	0.721	…	0.729	…	0.621	…	1.000

Tabel 3 Hasil Pengukuran *Cosine Similarity* Teknik *Doc2vec*

Id	J01	J02	J03	J04	J05	…	J18	J19	J20
J01	1.000	0.417	0.468	0.359	0.432	…	0.386	0.345	0.273
J02	0.417	1.000	0.424	0.320	0.344	…	0.421	0.387	0.367
J03	0.468	0.424	1.000	0.330	0.385	…	0.353	0.437	0.380
…	…	…	…	…	…	…	…	…	…
J09	0.401	0.360	0.342	0.397	0.362	…	0.412	0.375	0.358
J10	0.468	0.357	0.410	0.359	0.331	…	0.423	0.354	0.311
J11	0.406	0.370	0.280	0.351	0.364	…	0.357	0.350	0.404
…	…	…	…	…	…	…	…	…	…
J18	0.386	0.421	0.353	0.375	0.374	…	1.000	0.443	0.415
J19	0.345	0.387	0.437	0.402	0.470	…	0.443	1.000	0.383
J20	0.273	0.367	0.380	0.370	0.367	…	0.415	0.383	1.000

Word2vec tampak lebih sensitif terhadap kesamaan semantik antara kata-kata, sementara *Doc2vec* mungkin lebih baik dalam menangkap makna dokumen secara keseluruhan.

Dalam penelitian ini, hasil perbandingan menunjukkan bahwa teknik *Word2vec* lebih efektif dalam mengukur kemiripan dokumen berdasarkan representasi kata-kata yang terkandung di dalamnya. Kemampuan *Word2vec* dalam menangkap hubungan semantik antara kata-kata memungkinkannya untuk mengidentifikasi kemiripan antara jurnal-jurnal dengan lebih akurat. Namun, penting untuk diingat bahwa hasil ini spesifik untuk dataset dan konfigurasi yang digunakan dalam penelitian ini, dan dapat bervariasi tergantung pada karakteristik dataset, pra-pemrosesan yang dilakukan, dan parameter yang digunakan dalam pelatihan model *Word2vec* dan *Doc2vec*.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil analisis perbandingan teknik *Word2vec* dan *Doc2vec* dalam mengukur kemiripan dokumen menggunakan cosine similarity pada penelitian ini, dapat disimpulkan beberapa poin penting. Kedua teknik tersebut terbukti mampu merepresentasikan teks abstrak jurnal menjadi vektor numerik, memungkinkan pengukuran kemiripan dokumen dengan *cosine similarity*. Hasil perbandingan menunjukkan bahwa teknik Word2vec menghasilkan nilai cosine similarity yang lebih tinggi dibandingkan dengan teknik Doc2vec untuk pasangan jurnal yang sama. Sebagai contoh, pasangan jurnal J02 dengan J14 memiliki nilai cosine similarity 0.892 pada teknik Word2vec, sedangkan pada teknik Doc2vec nilai cosine similarity-nya adalah 0.434. Dalam lingkup penelitian ini, teknik Word2vec terbukti lebih efektif dalam menangkap kemiripan semantik antara jurnal-jurnal dibandingkan dengan teknik Doc2vec.

Berdasarkan hasil penelitian ini, beberapa saran dapat diberikan untuk penelitian selanjutnya. Saran utama adalah memperbanyak jumlah dan variasi jurnal yang digunakan sebagai dataset penelitian untuk memberikan gambaran yang lebih menyeluruh tentang kinerja teknik *Word2vec* dan *Doc2vec*. Selanjutnya, disarankan untuk menggali lebih dalam penggunaan teknik pra-pemrosesan teks yang lebih canggih guna meningkatkan kualitas representasi vektor dan hasil pengukuran kemiripan dokumen. Penting juga untuk membandingkan teknik *Word2vec* dan *Doc2vec* dengan teknik representasi teks lainnya demi mendapatkan pemahaman yang lebih komprehensif tentang kelebihan dan kekurangan setiap teknik. Selain itu, perlu dilakukan penelitian lebih lanjut mengenai pengaruh parameter-parameter dalam pelatihan model *Word2vec* dan *Doc2vec* untuk mengoptimalkan performa dan akurasi pengukuran kemiripan. Terakhir, disarankan untuk menerapkan teknik *Word2vec* dan *Doc2vec* pada kasus penggunaan yang lebih spesifik guna mengetahui efektivitas teknik-teknik tersebut dalam penerapan praktis di dunia nyata.

DAFTAR PUSTAKA

AGGARWAL, C. C. 2015. *Data Mining*. Springer International Publishing. https://doi.org/10.1007/978-3-319-14142-8

ALSHAMMERI, M., ATWELL, E., & ALSALKA, M. A. 2021. *Detecting Semantic-based Similarity between Verses of the Quran with Doc2vec. Procedia CIRP, 189.* https://doi.org/10.1016/j.procs.2021.05.104

AMALIA, A., SALIM SITOMPUL, O., BUDHIARTI NABABAN, E., & MANTORO, T. (2020*). A Comparison Study Of Document Clustering Using Doc2vec Versus Tfidf Combined With Lsa For Small Corpora*. *Journal of Theoretical and Applied Information Technology*, *15*, 17. www.jatit.org

BUDIMAN, A. E., & WIDJAJA, A. 2020. *Analisis Pengaruh Teks Preprocessing Terhadap Deteksi Plagiarisme Pada Dokumen Tugas Akhir*. *Jurnal Teknik Informatika Dan Sistem Informasi*, *6*(3). https://doi.org/10.28932/jutisi.v6i3.2892

CAHYANI, D. E., & PATASIK, I. 2021. *Performance comparison of tf-idf and word2vec models for emotion text classification. Bulletin of Electrical Engineering and Informatics*, *10*(5), 2780–2788. https://doi.org/10.11591/eei.v10i5.3157

CAHYONO, S. C. 2019. *Comparison of document similarity measurements in scientific writing using Jaro-Winkler Distance method and Paragraph Vector method*. *IOP Conference Series: Materials Science and Engineering*, *662*(5). https://doi.org/10.1088/1757-899X/662/5/052016

CARNEIRO, T., DA NOBREGA, R. V. M., NEPOMUCENO, T., BIAN, G. BIN, DE ALBUQUERQUE, V. H. C., & FILHO, P. P. R. 2018. *Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications*. *IEEE Access*, *6*, 61677–61685. https://doi.org/10.1109/ACCESS.2018.2874767

CHEN, Q., & SOKOLOVA, M. 2021. *Specialists, Scientists, and Sentiments: Word2Vec and Doc2Vec in Analysis of Scientific and Medical Texts. SN Computer Science*, *2*(5). https://doi.org/10.1007/s42979-021-00807-1

FATARUBA, F. 2018. *Penerapan Metode Cosine Similarity Untuk Pengecekan Kemiripan Jawaban Ujian Siswa*.

HACOHEN-KERNER, Y., MILLER, D., & YIGAL, Y. 2020. *The influence of preprocessing on text classification using a bag-of-words representation. PLoS ONE*, *15*(5). https://doi.org/10.1371/journal.pone.0232525

HASANAH, U., & MUTIARA, D. A. 2019. *Perbandingan Metode Cosine Similarity Dan Jaccard Similarity Untuk Penilaian Otomatis Jawaban Pendek*.

HICKMAN, L., THAPA, S., TAY, L., CAO, M., & SRINIVASAN, P. 2022. *Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations*. *Organizational Research Methods*, *25*(1), 114–146. https://doi.org/10.1177/1094428120971683

JURAFSKY, D., & MARTIN, J. H. 2023. *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft*.

KENTER, T., BORISOV, A., & DE RIJKE, M. 2016. *Siamese CBOW: Optimizing Word Embeddings for Sentence Representations*. https://github.com/ryankiros/

LAU, J. H., & BALDWIN, T. 2016. *An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation*. http://arxiv.org/abs/1607.05368

LE, Q. V., & MIKOLOV, T. 2014. *Distributed Representations of Sentences and Documents*. http://arxiv.org/abs/1405.4053

MIKOLOV, T., CHEN, K., CORRADO, G., & DEAN, J. 2013. *Efficient Estimation of Word Representations in Vector Space*. http://arxiv.org/abs/1301.3781

NURDIN, A., ANGGO, B., AJI, S., BUSTAMIN, A., & ABIDIN, Z. 2020. *Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks*. *Jurnal TEKNOKOMPAK*, *14*(2), 74.

PARWITA, W. G. S. 2020. *A document recommendation system of stemming and stopword removal impact: A web-based application. Journal of Physics: Conference Series*, *1469*(1). https://doi.org/10.1088/1742-6596/1469/1/012050

RAHMAN, S., SEMBIRING, A., SIREGAR, D., KHAIR, H., PRAHMANA, G., PUSPADINI, R., & ZEN, M. 2023. *Python : Dasar Dan Pemrograman Berorientasi Objek. Tahta Media Group*.

RIYANI, A., ZIDNY NAF'AN, M., & BURHANUDDIN, A. 2019. *Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen*. In *JLK* (Vol. 2, Issue 1).

RONG, X. 2014. *word2vec Parameter Learning Explained*. http://arxiv.org/abs/1411.2738

SASTROASMORO, S. 2007. *Beberapa Catatan tentang Plagiarisme**. *Maj Kedokt Indon*, *57*(8), 239–244.

SONG, X., SALCIANU, A., SONG, Y., DOPSON, D., & ZHOU, D. 2020. *Fast WordPiece Tokenization*. http://arxiv.org/abs/2012.15524

SUYANTO, A. H., DJATNA, T., & WIJAYA, S. H. 2023. *Mapping and predicting research trends in international journal publications using graph and topic modeling*. *Indonesian Journal of Electrical Engineering and Computer Science*, *30*(2), 1201–1213. https://doi.org/10.11591/IJEECS.V30.I2.PP1201-1213

TALA, F. Z. 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*.

WAHYUNI, R. T., PRASTIYANTO, D., & SUPRAPTONO, D. E. 2017. *Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi*. *9*(1), 18–23.