PENGELOMPOKAN BERITA BERDASARKAN KEMIRIPAN KONTEKSTUAL MENGGUNAKAN *K-MEANS CLUSTERING*

TESIS

Oleh:
MOHAMMAD YUSUF HIDAYAT
NIM. 220605210011

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

PENGELOMPOKAN BERITA BERDASARKAN KEMIRIPAN KONTEKSTUAL MENGGUNAKAN *K-MEANS CLUSTERING*

TESIS

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Magister (M.Kom)

Oleh:
MOHAMMAD YUSUF HIDAYAT
NIM. 220605210011

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

PENGELOMPOKAN BERITA BERDASARKAN KEMIRIPAN KONTEKSTUAL MENGGUNAKAN *K-MEANS CLUSTERING*

TESIS

Oleh:
MOHAMMAD YUSUF HIDAYAT
NIM. 220605210011

Telah diperiksa dan disetujui untuk diuji:
Tanggal 25 November 2025

Pembimbing I	Pembimbing II
Dr. M. Ainul Yaqin, M.Kom	Dr. Zainal Abidin, M.Kom
NIP. 19761013 200604 1 004	NIP. 19760613 200501 1 004

Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Prof. Dr. Ir. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

PENGELOMPOKAN BERITA BERDASARKAN KEMIRIPAN KONTEKSTUAL MENGGUNAKAN *K-MEANS CLUSTERING*

TESIS

Oleh:
MOHAMMAD YUSUF HIDAYAT
NIM. 220605210011

Telah Dipertahankan di Depan Dewan Penguji Tesis
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom.)
Tanggal: 03 Desember 2025

Susunan Dewan Penguji			**Tanda Tangan**
Penguji I	:	Dr. Totok Chamidy, M.Kom NIP. 19691222 200604 1 001	( )
Penguji II	:	Dr. Usman Pagalay, M.Si NIP. 19650414 200312 1 001	( )
Pembimbing I	:	Dr. M. Ainul Yaqin, M.Kom NIP. 19761013 200604 1 004	( )
Pembimbing II	:	Dr. Zainal Abidin, M.Kom NIP. 19760613 200501 1 004	( )

Mengetahui dan Mengesahkan
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Prof. Dr. Ir. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Mohammad Yusuf Hidayat
NIM : 220605210011
Program Studi : Magister Informatika
Fakultas : Sains dan Teknologi
Judul Tesis : "Pengelompokan Berita Berdasarkan Kemiripan Kontekstual Menggunakan *K-Means Clustering*"

Menyatakan dengan sebenarnya bahwa Tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran oang lain yang saya akui sebagai tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Tesis ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 03 Desember 2025
Yang membuat pernyataan,

Mohammad Yusuf Hidayat
NIM. 220605210011

MOTTO

"Jika kau menungguku untuk menyerah, kau akan menungguku selamanya"

PERSEMBAHAN

Dengan mengucapkan syukur *Alhamdulillahi Robbil Alamain,* Tesis ini saya persembahkan untuk:

1. Orang tua dan seluruh keluarga tercinta antara lain istri (Bdn. Khusnul Mahkhatul Sholeha, S.Tr.Keb), putri pertama (Aisha Shidqiya Yusuf), putri kedua (Nala Shofiya Yusuf) yang selalu memberikan doa dan semangat.
2. Seluruh Civitas Akademika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan kesempatan untuk menambah ilmu teknologi dan agama.
3. Seluruh Pimpinan dan Pengurus Institut Agama Islam Miftahul Ulum Lumajang dan MTs. Miftahul Ulum Jatiroto Lumajang.
4. Seluruh rekan-rekan Magister Informatika UIN Maulana Malik Ibrahim Malang yang tidak berhenti mengingatkan dan memberikan semangat.
5. Bapak, Ibu, saudara dan rekan-rekan yang tidak bisa saya sebutkan satu persatu.

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Syukur Alhamdulillah penulis haturkan kepada Allah Swt yang telah memberikan limpahan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus bisa menyeleasikan Tesis ini dengan baik. *Sholawat ma'as salam* semoga tetap tercurah limpahkan kepada Nabi Muhammad SAW.

Selanjutnya penulis haturkan ucapan terima kasih teriring doa dan harapan kepada semua pihak yang telah membantu dalam penyelesaian Tesis ini. Ucapan terima kasih khususnya penulis sampaikan kepada:

1. Bapak Dr. M. Ainul Yaqin, M.Kom. dan Bapak Dr. Zainal Abidin, M.Kom. selaku dosen pembibing Tesis, yang telah banyak memberikan pengarahan, pengalaman, dorongan, dan semangat yang sangat berharga.
2. Segenap civitas akademika Program Studi Magister Informatika, terutama seluruh Bapak/ Ibu Dosen atas ilmu dan bimbingan yang telah diberikan.
3. Keluarga tercinta yang senantiasa memberikan doa dan semangat.
4. Semua rekan-rekan seperjuangan yang ikut mendukung, membantu, mengingatkan dan memberi semangat.

Penulis menyadari bahwa dalam penyusunan Tesis ini masih terdapat kekurangan dan khilaf. Penulis berharap semoga Tesis ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. Aamiin Yaa Robbal Alamain.

Wassalamualaikum Wr. Wb.

Malang, 03 Desember 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Hidayat, Mohammad Yusuf, 2025, **Pengelompokan Berita Berdasarkan Kemiripan Kontekstual Menggunakan *K-Means Clustering*,** Tesis Program Magister Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. M. Ainul Yaqin, M.Kom (II) Dr. Zainal Abidin, M.Kom

Kata kunci *: Clustering, K-Means,* Kemiripan Konekstual*, WordNet, Keyword Extraction,* TF-IDF.

Penelitian ini membahas proses pengelompokan teks berita berbahasa Inggris berdasarkan kemiripan kontekstual menggunakan algoritma K-Means. Permasalahan utama dalam pengelompokan teks adalah bagaimana mengukur kesamaan konteks secara tepat sehingga dokumen yang memiliki topik serupa dapat dikelompokkan dengan akurat. Pada penelitian ini dilakukan ekstraksi kata kunci menggunakan TF-IDF, kemudian ditambahkan informasi semantik berbasis WordNet dengan perhitungan Wu-Palmer untuk meningkatkan pemahaman konteks antar kata. Proses Lematisasi menggunakan POS Tagging dilakukan agar kata-kata mendapatkan informasi fungsi kata yang tepat. Matriks kesamaan dibentuk menjadi representasi setiap dokumen untuk kemudian dilakukan Pengelompokan dengan K-Means. Evaluasi performa dilakukan menggunakan nilai Silhouette Score dan Davies Bouldin Index. Model yang diujikan untuk pembentukan matriks kesamaan adalah First POS, Max POS dan Max No POS. Hasil ujicoba menggunakan 1000 sampel dataset dengan beberapa skenario ujicoba. Dari hasil ujicoba, model First POS mendapatkan hasil performa paling baik pada jumlah *cluster* optimal k=3 dengan nilai Silhouette 0,505 dan nilai Davies Bouldin Index 0,667.

ABSTRACT

Hidayat, Mohammad Yusuf, 2025, **News Classification Based on Contextual Similarity Using the K-Means Clustering**, Thesis. Magister of Informatics. Postgraduate Program of Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisor: (I) Dr. M. Ainul Yaqin, M.Kom (II) Dr. Zainal Abidin, M.Kom

Keywords: Clustering, K-Means, Contextual Similarity, WordNet, Keyword Extraction, TF-IDF.

The research investigates the clustering process of English-language news texts based on contextual similarity using the K-Means algorithm. The primary challenge in text clustering lies in accurately measuring contextual similarity so that documents with similar topics can be effectively grouped. The researcher extracts keywords using TF-IDF and incorporates semantic information based on WordNet by applying the Wu-Palmer similarity measure to enhance contextual understanding between words. He performs lemmatization using POS tagging to assign accurate grammatical function information to each word. He constructs a similarity matrix to represent each document and subsequently applies K-Means clustering. The performance evaluation employs the Silhouette Score and the Davies-Bouldin Index. The researcher uses First POS, Max POS, and Max No POS models for constructing the similarity matrix. The experiments use 1,000 dataset samples across several testing scenarios. The experiment results indicate that the First POS model achieves the best performance at the optimal number of clusters, k = 3, with a Silhouette Score of 0.505 and a Davies-Bouldin Index of 0.667.

مستخلص البحث

هداية، محمد يوسف، ٢٠٢٥، تصنيف الأخبار بناءً على التشابه السياقي باستخدام الخوارزمية التصنيفية (*K-Means Clustering*)، رسالة الماجستير، قسم المعلومات بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. محمد عين اليقين، الماجستير؛ المشرف الثاني: زين العابدين، الماجستير.

الكلمات الرئيسية: تصنيف، خوارزمية تصنيفية، تشابه سياقي، *WordNet* ، استخراج كلمات مفتاحية، TF-IDF.

تتناول هذه الرسالة عملية تصنيف نصوص الأخبار باللغة الإنجليزية بناءً على التشابه السياقي باستخدام خوارزمية تصنيفية. تكمن المشكلة الرئيسية في تصنيف النصوص في كيفية قياس التشابه السياقي بدقة بحيث يمكن تجميع المستندات التي تتناول موضوعًا مشابهًا بشكل صحيح. في هذا البحث، تم استخراج الكلمات المفتاحية باستخدام TF-IDF، ثم تم إضافة المعلومات الدلالية المستندة إلى *WordNet* مع حساب *Wu-Palmer* لتعزيز فهم السياق بين الكلمات. تم تنفيذ عملية التصريف الصرفي باستخدام *POS Tagging* حتى تحصل الكلمات على معلومات الوظيفة اللغوية الصحيحة. تم تشكيل مصفوفة التشابه لتمثل كل مستند، ومن ثم تم إجراء التجميع باستخدام خوارزمية تصنيفية. تم تقييم الأداء باستخدام مؤشر *Silhouette* ومؤشر *Davies Bouldin*. النماذج التي تم اختبارها لتشكيل مصفوفة التشابه هي *First POS* و *Max POS* و *Max No POS* . جاءت نتائج نتائج الاختبار باستخدام ١٠٠٠ عينة من مجموعة البيانات مع عدة سيناريوهات اختبار. من نتائج الاختبار، حصل نموذج *First POS* على أفضل أداء عند عدد التجمعات الأمثل *k=3* مع قيمة مؤشر سيلويت ٠,٥٠٥ وقيمة مؤشر ديفيز بولدين ٠,٦٦٧.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pengelompokan teks berita atau artikel masih menjadi topik yang masih sering dikerjakan dan diteliti. Mengelompokan berita berdasarkan karakteristik yang ada dalam teks tersebut menjadi isu yang sangat penting Bisandu et al., (2018). Bagaimana suatu berita bisa dikelompokkan dan diorganisir berdasarkan topik dan isu yang relevan. Dalam dunia jurnalisme, mengelompokan berita sangat penting dilakukan agar pembaca bisa menikmati berita sesuai dengan apa yang menjadi konsennya saat ini. Selain itu, juga membantu pemahaman pembaca tentang fenomena yang sedang berkembang dan bagaimana berita tersebut bisa saling terkait. Bidang pemrosesan bahasa alami ini menjadi salah satu teknik dengan profil yang tinggi dan mempunyai wilayah pengaplikasian yang sangat luas, salah satunya yaitu document organization Yeasmin et al., (2023). Hal ini sangat membantu pembaca dalam mencari tema berita yang ingin dibacanya.

Menyajikan berita yang sesuai dengan kelompok beritanya agar pembaca mendapatkan berita sesuai isu atau tema yang diminati sejalan dengan ayat Al-Quran Surat Al-Hujurat ayat 13:

يٰٓاَيُّهَا النَّاسُ اِنَّا خَلَقْنٰكُمْ مِّنْ ذَكَرٍ وَّاُنْثٰى وَجَعَلْنٰكُمْ شُعُوْبًا وَّقَبَاۤىِٕلَ لِتَعَارَفُوْا ۚ اِنَّ اَكْرَمَكُمْ عِنْدَ اللّٰهِ اَتْقٰىكُمْ ۗاِنَّ اللّٰهَ عَلِيْمٌ خَبِيْرٌ

Wahai manusia! Sungguh, Kami telah menciptakan kamu dari seorang laki-laki dan seorang perempuan, kemudian Kami jadikan kamu berbangsa-bangsa dan bersuku-suku agar kamu saling mengenal. Sesungguhnya yang paling mulia

di antara kamu di sisi Allah ialah orang yang paling bertakwa. Sungguh, Allah Maha Mengetahui, Mahateliti. (QS. [49] Al-Hujurat: 13)

Meskipun tidak secara jelas tergambar tentang informasi, namun ayat ini membicarakan tentang pengelompokan manusia berdasarkan jenis kelamin, suku dan bangsa yang menjadi ciri khas seseorang agar manusia bisa saling mengenal. Pengelompokan yang tersirat dalam ayat tersebut bahwa sesuatu hal mempunyai ciri khas tersendiri dan menjadikan sesuatu itu bisa dikenali dari kelompok atau *cluster* yang mana.

Menurut Z. Chen *et al.*, (2023) , komponen utama pada tugas clustering yaitu representasi teks dan algoritma clustering itu sendiri. Representasi yang paling umum digunakan yaitu TF-IDF. Namun Radu mengungkapkan bahwa TF-IDF menyebabkan *dimension disaster.* Model ini menjadikan algoritma cerdas yang menggunakannya sebagai vektor representasi bekerja lebih ekstra untuk dataset atau *corpus* yang besar dan juga tidak membawa informasi semantik.

Meskipun demikian TF-IDF masih sangat relevan untuk tugas representasi teks, karena dia mengekstraksi kata-kata inti dari suatu dokumen Kim & Gil (2019). Maka pada penelitian ini diusulkan beberapa skenario yang melibatkan TF-IDF dan juga pendekatan kontekstual untuk membangkitkan representasi informasi semantik dari teks berita untuk dijadikan inputan ke dalam metode pengelompokan berita. Untuk mendapatkan hasil dengan performa yang tinggi maka penelitian ini juga menggunakan algoritma clustering yang didapatkan dari hasil studi pustaka sebagai salah satu komponen utama dari tugas *clustering*.

1.2 Pernyataan Masalah

Dari penjabaran latar belakang di atas, maka pernyataan masalah yang diambil dalam penelitian ini yaitu bagaimana mengelompokan teks berita berbahasa inggris berdasarkan kesamaan kontekstual yang dibuat dengan memanfaatkan representasi TF-IDF untuk ekstraksi *keywords* menggunakan K-Means.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengetahui model atau metode paling efektif untuk mengelompokan teks berita berdasarkan kesamaan kontekstual.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian akan berfokus pada penggunaan informasi semantik untuk pengelompokan teks berita dengan menggunakan K-Means.
2. Evaluasi kualitas klaster dilakukan menggunakan Silhouette Coefficient dan Davies-Bouldin Index (DBI)
3. Data yang digunakan adalah teks berita berbahasa inggris.

1.5 Manfaat Penelitian

Penelitian ini memberikan kontribusi sebagai berikut:

1. Manfaat Teoritis

Pengembangan metode pengelompokan teks berbasis kesamaan kontekstual melalui ekstraksi kata kunci TF-IDF diperkaya dengan informasi semantik

WordNet, kemudian dikelompokkan dengan K-Means dan di evaluasi dengan Silhouette dan DBI.

2. Manfaat Praktis

   a. Untuk pembaca dan pengguna berita digital

      Hasil pengelompokan teks berita dapat digunakan sebagai dasar pengelompokan konten berita berdasarkan kedekatan konteks, sehingga membantu proses penataan dan penyaringan informasi, meskipun keluaran penelitian masih berupa data hasil klaster.

   b. Untuk Lembaga, organisasi, atau analisis media

      Klaster berita yang dihasilkan dapat dimanfaatkan sebagai informasi awal untuk analisis tren isu, sebelum dikembangkan lebih lanjut ke dalam sistem pemantauan atau analisis lanjutan.

BAB II

STUDI PUSTAKA

2.1 Pengelompokan Artikel Berita

Pengelompokkan berita sangat berguna untuk mempermudah pengaksesan informasi dan juga pemahaman terhadap topik yang sedang hangat dibicarakan. Penelitian untuk mengelompokan berita sudah beberapa kali dilakukan. Seperti yang dilakukan Y. Chen *et al.*, (2010). yang membandingkan hasil pengelompokan teks antara metode SOM dan *k*-Means pada 420 artikel dengan topik yang berbeda. Performa yang dihasilkan yakni SOM mendapatkan nilai F-Measure 0,93 lebih bagus dibandingkan dengan *k*-Means pada hampir semua pengujian. Namun pada *k* tertentu *k*-Means masih bisa mengungguli SOM.

Guan *et al.*, (2011) melakukan pengelompokan teks dengan memberikan dua kontribusi yakni matriks kesamaan yang menggambarkan struktur informasi teks dan metode kontruksi *seeds* untuk meningkatkan proses pengelompokan *semi-supervised.* Mereka menggunakan algoritma yang berbasis Affinity Propagation yang disebut SAP (Seeds Affinity Propagation). Dari hasil ujicoba didapatkan matriks kesamaan yang lebih efektif 21% lebih tinggi dari metode AP biasa dan mempercepat konvergensi bobot yakni hanya menggunakan 76% iterasi dari AP yang asli. Metode yang diusulkan mendapatkan rata-rata F-Measure tertinggi 0,599. Namun waktu eksekusi lebih lama dibanding dengan AP biasa.

Azzopardi & Staff (2012) melakukan pengelompokan berita dari sumber yang berbeda dengan metode *incremental clustering* yang diadopsi dari k-Means dengan pembobotan TF.IDF. Pada pengujian menggunakan *Google News Corpus*,

metode yang diusulkan mendapatkan performa F-Measure 0,8370 dengan Presisi 0,9518. Namun ketika menggunakan *Reuters-RCV1 Corpus* dan *Yahoo! News Corpus*, metode yang diusulkan mendapatkn F-Measure yang lebih rendah. Peneliti mengungkapkan bahwa ini terjadi karena corpus selain *Google News* memiliki entropi yang sangat tinggi sehingga tidak ada *similarity threshold* yang jelas untuk mengklasifikasikan dokumen pada kelompok yang berbeda. Penelitian ini tidak menggunakan proses yang kompleks seperti pengukuran semantic, tujuannya adalah pengklasifikasian secara online.

Bouras & Tsogkas (2012) menggunakan WordNet dan pengukuran kesamaan teks untuk mengelompokan berita. Metode yang dipakai dalam penelitian ini adalah *k*-Means dan mereka namakan dengan W-Kmeans. Performa yang dihasilkan dari metode ini mengungguli metode lain dengan nilai F-Measure 0,61 bahkan lebih unggul dibanding *k*-Means++ yang mendapatkan nilai 0,51. Dari segi *Clustering Index* metode yang diusulkan masih belum mendapat nilai tertinggi, namun demikian waktu eksekusi yang dibutuhkan adalah yang paling singkat.

Bora *et al.*, (2012) mengajukan sebuah metode heuristik untuk mengelompokan *headlines news* yang secara gramatik dan semantik berbeda dengan badan teks yang lebih besar seperti *blog posts* dan *reviews.* Peneliti mengajukan dua versi berdasarkan metode yang diusulkan yakni *Frequent Term-based* dan *Frequent Noun-Based.* Pengujian dilakukan pada 5 dataset yakni Reuters343, Reuters2388 (News Headlines), CICLing-2002, Hep-ex, dan KnCr yang berisi abstrak ilmiah. Fitur unigram dan representasi presence dari kata

digunakan sebagai representasi vektor. Peneliti menemukan bahwa representasi TF.IDF menyebabkan kerugian karena jumlah dimensinya. Dari hasil pengujian, versi heuristik *Frequent Noun Plus* mendapatkan nilai F-Measure tertinggi 0,86 menggunakan *corpus* Reuters343 mengungguli *k*-Means. Pada percobaan dataset lain masih, *k*-Means masih lebih unggul dari metode yang diusulkan.

Zheng *et al.*, (2013) mengusulkan *framework* baru menggunakan *ensemble hirearchial clustering* yang diberi nama *PENETRATE: Personalized News recommendation framework using ensemble hieRArchial clustering*. Penelitian ini mengajukan metode untuk merekomendasikan berita berdasarkan prefrensi pembaca dengan berita dengan topik yang sama. Dari hasil pengujian ditemukan bahwa metode yang diusulkan medapat nilai F-Measure tertinggi 0,3640. Namun metode ini memerlukan daya komputer yang besar untuk data ukuran besar, dan kemungkinan hasil keliru jika terdapat *noise* atau *outlier* dalam data, serta sulit diterapkan pada data dengan dimensi atau fitur yang kompleks.

Beberapa penelitan sering menggunakan *k*-Means dan pengembangannya sebagai metode untuk pengelompokan data. Salah satu alasannya karena *k*-Means dirasa menjadi metode yang paling umum dan gampang diterapkan dalam penelitian. Salah satunya adalah penelitan Aubaidan *et al.*, (2014) yang membandingkan metode *k*-Means dengan *k*-Means++ dalam mengelompokan domain kejahatan. Peneliti menggunakan persamaan Cosine dan Jaccard Similarity untuk mengukur hasil *clustering* dengan metode yang diusulkan. Dari ujicoba *k*-Means++ mendapatkan performa lebih baik yakni F-Measure 0,89 pada 168 kejadian dan *k*-Means mendapatkan nilai 0,819. Hal ini terjadi karena K-

Means menentukan inisiasi *centroid* secara random, sedangkan *k*-Means++ memilih inisiasi *centroid* kedua secara matematis.

Penelitian selanjutnya dilakukan oleh H. C. Yang *et al.*, (2015) yang mengembangkan *maps* pada SOM secara hirarki dan menerapkan pengelompokan teks pada dataset Reuters-21578. Peneliti menekankan pendekatan yang diusulkan pad penggabungan proses *learning* dengan *text mining*. Dari hasil ujicoba ditemukan bahwa metode yang diusulkan bisa bersaing dengan metode pembanding dan berhasil mendapatkan nilai tertinggi F-Measure 0,9245 pada hirarki E5. Kekurangan dalam penelitian ini seperti yang diungkapkan penulis paper bahwa pemilihan parameter masih menjadi masalah.

Wei *et al.*, (2015) Menambahkan informasi semantik dari *ontology* seperti WordNet untuk meningkatkan kualitas pengelompokan teks. Peneliti menggunakan metode *k*-Means yang dikombinasikan dengan *Disambiguited Core Semantic (DCS).* Dari hasil ujicoba didapatkan bahwa penambahan informasi semantik yang dilakukan oleh peneliti mampu membuat *k*-Means mendapatkan performa tertingginya dalam ukuran F-Measure 0,728. Namun demikian masih ada beberapa keterbatasan dalam penelitian ini. Diantaranya yakni beberapa kata penting yang tidak termasuk dalam *WordNet Lexicon* tidak dianggap sebagai konsep untuk evaluasi kesamaan. Selain itu, metode yang diusulkan akan bekerja lebih baik jika hubungan antar kata terwakili secara menyeluruh di dalam WordNet.

Jiang *et al.*, (2016) menggunakan pendekatan berbasis Affinity Propagation Clustreing (APC) yang dikembangkan dengan metode Path untuk mengukur kesamaan semantik dan dinamakan dengan APC-PS. Peneliti bertujuan untuk memperbaiki metode yang ada untuk menangani sampel dengan distribusi yang kompleks. Dari hasil ujicoba ditemukan bahwa metode yang diusulkan mendapat nilai F-Measure 0,665 mengungguli APC biasa yang mendapatkan nilai F-Measure 0,635. Namun pada beberapa skenario ujicoba metode yang diusulkan masih bisa diungguli oleh *k*-Means yang mendapat nilai F-Measure 0,6662.

Z. Zhang *et al.*, (2020) memperbaiki representasi TF.IDF sebagai vektor input karena menjadi penyebab dari *dimension disaster* dengan menggunakan Bag-of-Near-Synonyms (BoNS) model. Model ini menggunakan vektor SF-IDF untuk meng-*encode* sebuah dokumen. Dan setiap dimensinya berhubungan dengan *near-synonyms set* yang dihasilkan dari word embeddingn dan agglomerative clustering, tapi bukan untuk sebuah *single word*. Model ini dijadikan input untuk mengelempokan berita dari web teknologi China dengan menggunakan Single Pass Algorithm. Untuk mereduksi komputasi, peneliti mengusulkan hashed version dari representasi SF-IDF dan dinamakan dengan hSF-IDF. Representais yang dihasilkan dikomparasikan dengan beberapa representasi *baseline* seperti TF-IDF, LDA, BoC, dan BERT-based. Dari hasil ujicoba ditemukan bahwa metode yang diusulkan mendapatkan performa tertinggi yakni F-Measure 0,8971 mengalahkan model BERT-based. Namun menurut penelitinya sendiri, model yang diusulkan memeliki keterbatasan yakni tidak bisa mengatasi masalah polisemi.

B. Zhang & Hou (2020) melakukan improvisasi pada algoritma *k*-Means untuk menghitung jumlah *cluster* yang cocok berdasarkan prinsip maksimum dan minimum. Jumlah ini yang nantinya akan diaplikasikan pada model jaringan SOM sebagai jumlah neuron. Peneliti juga menggunakan Latent Semantic Indexing pada tahap preprocessing. Dari hasil pengembangan ini ditemukan bahwa hasil evaluasi F-Measure pengelompokan medapatkan nilai 0,827 untuk 4 kategori.

Saravanakumar *et al.*, (2021) mengusulkan untuk mengelompokan aliran berita online dengan varian dari algoritma K-Means dengan dua kunci yang berbeda, yakni menghitung kesamaan antara dokumen dan cluster sepanjang serangkaian representasi, kemudian memutuskan keanggotaan cluster dengan *neural classifier.* Peneliti mengambil representasi TF-IDF yang diaplikaikan pada judul, isi, judul+isi untuk setiap dokumen, BERT yang diaplikasikan pada isi berita, dan timestamp berupa tanggal penerbitan berita. Dari hasil ujicoba ditemukan bahwa hasil dari skenario TF.IDF + E-S-BERT (*entity aware BERT fine-tuned on event similarity*) + Timestamp mendapatkan hasil F-Measure tertinggi yakni 94,76.

S. Yang & Tang (2022) menganalisis berita untuk mendeteksi topik, meskipun berdasarkan informasi semantic. Meskipun tidak langsung membahas tentang clustering, namun di dalamnya ada proses untuk itu. Di beberapa *paper* dijelaskan bahwa clustering dan ekstraksi topik adalah proses yang tidak terpisahkan. Dia menggunakan metode *incremental clustering,* dimana pengelompokannya berdasarkan hitungan *similarity* antar semua teks berdasarkan *graph kernel.* Model yang diusulkan diberi nama CSG (*Capsule Semantic Graph*)

Peneliti menggunakan dataset dari Chinese Emegency Corpus, THUCNews, dan 20newsgrup. Representasi teks menggunakan Word2Vec word embedding. Minimal similarity adalah 0,4. Dari hasil ujicoba ditemukan bahwa model yang diusulkan mendapatkan nilai F-Measure tertinggi yakni 78,95% pada dataset CEC; 76,25% pada dataset 20newsgroup; dan 75,88% pada THUCNews.

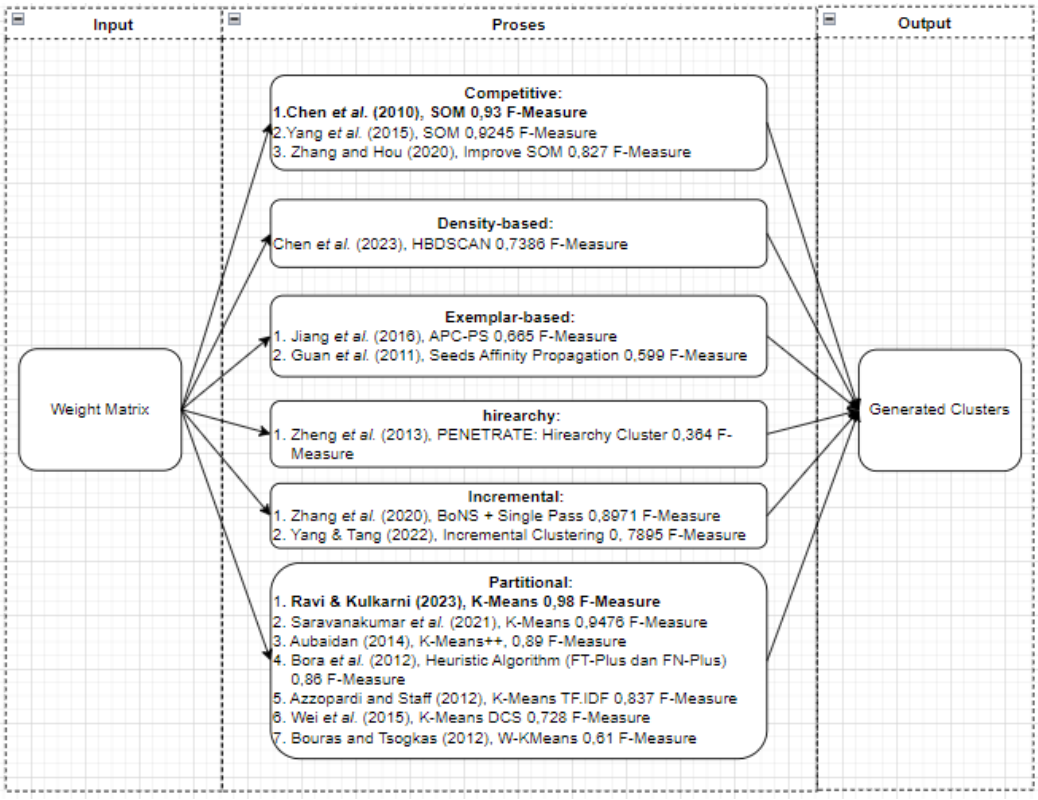
Ravi & Kulkarni (2023) mencoba mengelompokan teks dari data twitter dengan beberapa teknik text embedding. Diantaranya yakni TF.IDF, Word2Vec, GloVe, Doc2Vec, dan BERT. Dataset yang digunakan adalah data twitter dari 5 news channels yang terkenal di india. Kemudian menggunakan K-Means untuk proses Clustering. Dari hasi ujicoba performa didapatkan hasil bahwa model representasi BERT dan K-Means Clustering mendapatkan hasil yang paling baik dilihat dari internal dengan nilai Silhouette 0,65 dan DB-Score terendah 0,48. Dari sisi eksternal mendapatkan nilai F-Measure 0,98. Namun di beberapa penelitian menyebutkan bahwa model BERT-*based* menyebabkan anistropik atau ketidakseragaman.

Z. Chen *et al.*, (2023) melakukan penelitian tentang clustering text dan ekstraksi topik dan diberi nama ClusTop. Mereka mengusulkan framework yang terdiri dari 4 komponen: model training bahasa yang ditingkatkan dengan BERT-enhanced dan SimCSE-enhanced sebagai *text embedding*; reduksi dimensi dengan UMAP, t-SNE, dan PCA; clustering text dengan DBSCAN, K-Means, dan HDBSCAN; kemudian ekstraksi topik. Model *baseline* yang dijadikan perbandingan yaitu LDA, FastText, dan Top2Vec. Kemudian mengevaluasi masing-masing skenario dengan dua jensi matriks evaluasi yaitu interneal

(Silhouette Coefficient, Kalinski-Harabaz Index, dan Davies-Bouldin Index), dan eksternal (Adjusted Rand Index, Adjusted Mutual Information, Purity dan F1-Measure). Dari hasil ujicoba dengan dataset THUCNews diperoleh hasil F1-Measure tertinggi 0.738656874 dengan skenario BERT-enhanced+t-SNE+HDBSCAN. Jika dilihat dari DBI terendah maka didapatkan oleh skenario SimCSE-enhanced+UMAP+ HDBSCAN dengan skor 0.163748030, menunjukkan bahwa semakin jelas terpisahnya antar kelompok.

2.2 Kerangka Teori

Pada bagian ini, penulis memetakan beberapa metode yang dipaparkan dari beberapa penelitian diatas dan digambarkan dalam suatu kerangka teori yang terdiri dari input, proses utama, dan output.



Gambar 2.1. Kerangka Teori

Dari Gambar 2.1 ditemukan beberapa metode yang diukur menggunakan metode F-measure dan juga beberapa metode pengelompokan yang dikelompokan berdasarkan jenis *clustering*-nya. Di dalam beberapa paper yang di-*review,* sebenarnya tidak hanya diukur menggunakan satu model evaluasi saja, namun penulis memberikan gambaran dengan metode pengukuran yang sama. Perhitungan F-Measure menggunakan hasil uji performa dari segi *recall* dan presisi. Evaluasi ini sebenarnya lebih cocok untuk tugas klasifikasi. Namun bisa digunakan untuk tugas clustering jika terdapat informasi *ground truth*. Berikut adalah berbagai metode dari beberapa paper yang telah di-*review* dan di-*ranking* dari tingkat F-measure tertinggi.

Tabel 2. 1. Beberapa Metode yang diurutkan berdasarkan nilai F-Measure

No	Penulis	Dataset	Pra-Proses	Proses Utama	FM
1	Ravi and Kulkarni (2023)	Data Twitter News Channel India	BERT	K-Means	0,98
2	Saravanakumar et al. (2021)	*Standard Multilingual News Stream Clustering Dataset*	*TF.IDF + E-S-BERT (Entity aware BERT fine-tuned on event similarity) + Timestamp*	K-Means	0,9476

No	Penulis	Dataset	Pra-Proses	Proses Utama	FM
3	Chen et al. (2010)	645 Different Topic Documents	Tidak diketahui	SOM	0,93
4	Yang et al. (2015)	Reuters-21578	*Document vectors 0 and 1 for absence presence keyword + Topic identification*	SOM	0,9245
5	Zhang et al. (2020)	Data primer dari beberapa Web portal Teknologi China tanggal 8 Februari sampai 11 Maret 2019	Pendekatan Model BoNS (*Hashed version SF-IDF* untuk reduksi komputasi) yang dinamakan dengan hSF-IDF	Agglomerative Clustering	0,8971
6	Aubaidan (2014)	Crime dataset yang berisi 247 dokumen dari website Bernama News	Representasi TF-IDF, evaluasi hasil clustering dengan penghitungan similaritas cosine dan jaccard	K-Means++	0,89
7	Bora et al. (2012)	Reuters343, Reuters2388,	Pendekatan fitur *unigram*	Heuristic Frequent	0,86

No	Penulis	Dataset	Pra-Proses	Proses Utama	FM
		CICLing-2002, Hep-ex, KnCr (Scientific Abstracts)	dan representasi fitur *presence keyterm*	Term-based Clustering	
8	Azzopardi and Staff (2012)	Reuters-RCV1, Yahoo! News, Google News	Menggunakan Incremental Cluster yang yang diadopsi dari KMeans, TF-IDF	Incremental Clustering	0,837
9	Zhang and Hou (2020)	300 teks *news stories*	Latent Semantic Indexing	Improved SOM	0,827
10	Yang & Tang (2022)	THUCNews	Word2Vec Embedding, Capsule Semantic Graph	Incremental Clustering	0.7895
11	Chen et al. (2023)	THUCNews	BERT-enhanced + t-SNE	DBSCAN	0,7386
12	Wei et al. (2015)	Reuters-21578	WordNet and lexical chains	k-Means	0,728
13	Jiang et al. (2016),	UCI datasets	Path-based Similarity	Affinity Propagation Clustering	0,665
14	Bouras and Tsogkas (2012)	20 major news portals (bbc, cnn, reuters)	Term Frequency dan WordNet	W-KMeans	0,61
15	Guan et al.	Reuters-	Matrik	Seeds Affinity	0,599

No	Penulis	Dataset	Pra-Proses	Proses Utama	FM
	(2011)	21578	similarity	Propagation	
16	Zheng et al. (2013), PENETRATE: Hierarchy Clustering	Artikel berita dan Riwayat akses pengguna	*User Profile Groups*	Hirearchy Clustering	0,364

Berdasarkan tinjauan pustaka terhadap berbagai penelitian *clustering* teks dalam rentang tahun 2011-2023 pada tabel 2.1, terlihat bentuk representasi teks yang digunakan berpengaruh signifikan terhadap performa model clustering. enelitian dengan performa tertinggi dilakukan oleh Ravi & Kulkarni (2023) yang menggunakan representasi berbasis BERT embedding dan menghasilkan nilai F-Measure sebesar 0.98. Hasil ini menunjukkan bahwa model language representation berbasis transformer mampu menangkap konteks semantik secara efektif.

Di susul dengan penelitian dari Saravanakumar *et al.*, (2021) memperoleh performa 0.9476 dengan memadukan TF-IDF, embedding E-S-BERT, serta informasi temporal, diikuti oleh Y. Chen *et al.*, (2010) dan H. C. Yang *et al.*, (2015) yang masih menggunakan representasi teks tradisional dan menghasilkan nilai F-Measure di atas 0.92.

Meskipun representasi berbasis transformer memberikan performa tertinggi, pendekatan tersebut membutuhkan sumber daya komputasi yang besar dan tingkat interpretabilitas yang rendah. Sementara itu, beberapa penelitian menunjukkan bahwa peningkatan representasi berbasis TF-IDF dengan pengetahuan semantik,

seperti WordNet, dapat meningkatkan kualitas clustering dengan kompleksitas yang lebih rendah.

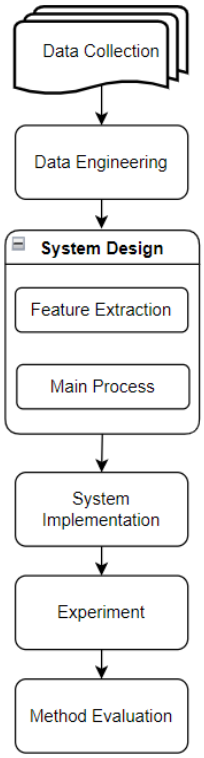
Berdasarkan permasalahan tersebut, penelitian ini secara khusus membandingkan performa TF-IDF murni dan TF-IDF yang diperkaya dengan WordNet menggunakan algoritma clustering K-Means. Tujuannya adalah untuk menguji kontribusi penambahan informasi semantik terhadap kualitas cluster tanpa harus menggunakan model embedding kompleks seperti BERT.

BAB III

Desain Penelitian

3.1 Prosedur Penelitian

Langkah-langkah yang dilakukan dalam penelitian ini digambarkan sebagai berikut.



Gambar 3. 1 Diagram Prosedur Penelitian

Dari Gambar 3.1. dijelaskan bahwa dalam penelitian ini, tahapan yang dilakukan terdiri dari 6 langkah utama yang akan dijelaskan sebagai berikut,

1. *Data Collection*, yaitu tahap pengumpulan dataset yang kita gunakan untuk eksperimen

2. *Data Engineering*, yaitu tahap penformatan data yang sudah terkumpul sesuai dengan kebutuhan kita. Pada tahap ini data di-*filter*, dibersihkan sehingga siap untuk proses selanjutnya. Tahap ini juga bisa disebut dengan tahap *preprocesing*

3. *System Design*, yaitu tahap mendesain system inti dari program yang kita buat. Di dalamnya terdapat *Feature extraction* untuk mengekstrak fitur-fitur yang akan dijadikan inputan untuk kemudian diolah pada *Main Process*. *Main Process* berisi metode yang sudah dipilih untuk pengelompokan berita.

4. *System implementation*, yaitu tahap penentuan perangkat komputer yang akan dipakai sebagai media uji coba. Termasuk minimal spesifikasi dari komputer yang akan digunakan.

5. *Experiment*, yaitu tahap penentuan skenario dalam ujicoba dataset terhadap model yang dipakai. Di dalam tahap ini juga dibahas parameter-parameter yang menentukan bahwa metode atau model yang dipakai cocok atau tidak dalam kasus dalam penelitian ini.

6. *Method Evaluation*, yaitu tahap pembahasan uji performa metode yang dipakai. Pembahasan bisa dari segi *error* atau kelemahan yang terjadi. Faktor yang menjadi penyebab dari kelemahan tersebut, dan mungkin juga bisa dijadikan saran untuk penelitian kedepan.

3.2 Data Collection

Penelitian ini menggunakan dataset sekunder atau data publik *Global News Dataset* dipublikasi oleh (Kumar Saksham, n.d.) yang tersedia di situs *Kaggle.com (https://www.kaggle.com/datasets/everydaycodings/global-news-dataset)*.

	article_id	source_name	author	title	description	url	published_at	category	full_content
0	89541	International Business Times	Paavan MATHEMA	UN Chief Urges World To 'Stop The Madness' Of ...	UN Secretary-General Antonio Guterres urged th...	https://www.ibtimes.com/un-chief-urges-world-s...	2023-10-30 10:12:35.000000	Nepal	UN Secretary-General Antonio Guterres urged th...
1	89545	The Indian Express	Editorial	Sikkim warning: Hydroelectricity push must be ...	Ecologists caution against the adverse effects...	https://indianexpress.com/article/opinion/edit...	2023-10-06 01:20:24.000000	Nepal	At least 14 persons lost their lives and more ...
2	89551	Al Jazeera English	Kaushik Raj	Pro-Israel rallies allowed in India but Palest...	India, the first non-Arab country to recognise...	https://www.aljazeera.com/news/2023/10/25/pro-...	2023-10-25 09:58:17.000000	Nepal	India, the first non-Arab country to recognise...
3	89555	The Indian Express	New York Times	No nation in the world is buying more planes t...	India's largest airlines have ordered nearly 1...	https://indianexpress.com/article/business/avi...	2023-11-02 05:48:58.000000	Nepal	Written by Alex Travelli and Hari Kumar No nat...

Gambar 3. 2. Isi Global News Dataset

Gambar 3.2 menunjukkan dataset berisi artikel yang dipublikasi pada tanggal 30 September 2023 - 10 Oktober 2023 dari beberapa sumber, diantaranya ETF Daily News, The Times of India, dan sumber yang lain.

Tabel 3. 1 Deskripsi Dataset asli

No	Nama Kolom	Tipe Data	Jumlah Nilai Non-Null	Keterangan
1	article_id	int64	105,375	ID unik artikel
2	source_id	object	24,495	ID sumber berita (banyak missing value)
3	source_name	object	105,375	Nama sumber berita
4	author	object	97,156	Nama penulis
5	title	object	105,335	Judul berita
6	description	object	104,992	Deskripsi singkat artikel
7	url	object	105,375	Link artikel
8	url_to_image	object	99,751	Link gambar artikel
9	published_at	object	105,375	Waktu publikasi
10	content	object	105,375	Isi berita
11	category	object	105,333	Kategori berita
12	full_content	object	58,432	Konten lengkap (banyak missing value)

Dilihat dari tabel 3.1, jumlah *article_id* dan *full_content* berbeda. Ini mengindikasikan ada *Missing Value* pada beberapa data. Oleh karena itu, dilakukan pemangkasan pada dataset dengan fungsi *dropna()* dari dataframe dan dilakukan reset indeks agar indeks lebih rapih, sehingga menghasilkan sejumlah 56.134 data. Setelah di-*export* dalam bentuk excel, terlihat beberapa pengkategorian umum seperti *health, stock, technology* dan lain-lain. Ada juga yang menggunakan kategori spesifik dan tidak merepresentasikan topik tematik seperti nama negara dan nama aplikasi, sehingga dilakukan filtrasi dan mendapatkan data sebanyak 22.015 dengan 23 kategori. Data inilah nanti yang akan dilakukan proses lematisasi dengan menggunakan *pos tagging*.

Tabel 3. 2 Data kategori dan jumlah dataset per kategori

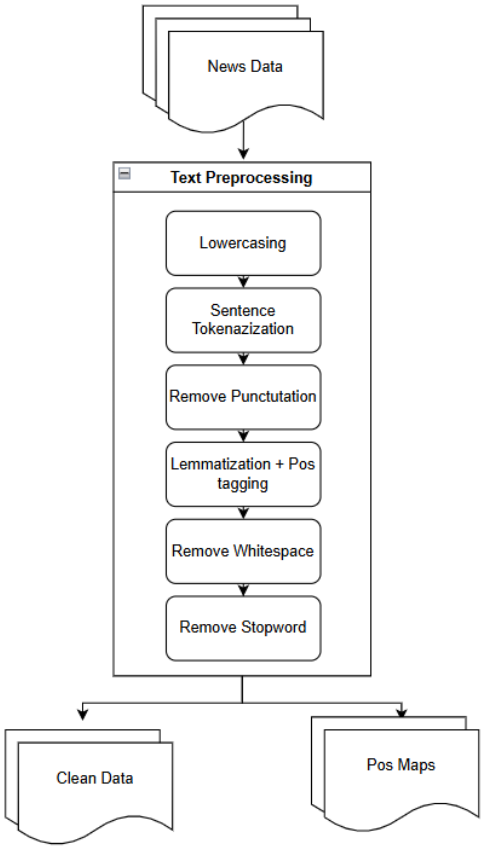
category	**count**
Stock	3677
Health	2060
Technology	1954
Real estate	1930
Finance	1773
Education	1289
Food	1087
Jobs	926
Weather	896
Science	872
Artificial Intelligence	651
Climate	643
Fashion	637
Sports	583
Music	548
Politics	512
Beauty	480
Relationships	473
Movies	419
Architecture	309

category	count
Art	166
Motivation	79
Entrepreneurship	51

Tabel 3.2 menunjukkan bahwa jumlah dataset terbanyak terbanyak terdapat pada kategori *Stock* dengan jumlah 3677 data, dan yang paling sedikit yaitu pada kategori *Entrepreneurship* dengan jumlah 51 data.

3.3 Data Engineering

Data Engineering merupakan cara untuk membersihkan data dengan cara yang dipilih sesuai kebutuhan.. Berikut adalah proses pembersihan data sesuai dengan data yang didapatkan.



Gambar 3. 3 Step by step Preprocessing

Gambar 3.3 menjelaskan *step by step* langkah pembersihan data dimulai pengertian terhadap data yang sudah diperoleh. Pengertian terhadap data ini penting karena untuk menentukan langkah apa saja yang diambil sehingga data teks yang diperoleh menjadi bersih.

Berikut langkah-langkah yang dipakai dalam studi kasus ini:

1. *Lowercasing*

Tahap *preprocessing* yang paling sederhana adalah *lowercasing*. Tahap ini bertujuan untuk mengubah huruf yang ada di dalam *corpus* menjadi huruf kecil semua.. Di dalam python, ini bisa dilakukan dengan menggunakan modul yang sudah tersedia dan tidak menggunakan *external library*.

2. *Sentence Tokenizing*

Sentence Tokenizing merupakan proses untuk memisahkan berita menjadi potongan-potongan kalimat di tampung dalam suatu list. Tahap ini dilakukan agar proses lematisasi kata dengan menggunakan pos tag sesuai dengan fungsi kata meskipun berasal dari akar kata yang sama. Sebagai contoh kata *"confirmed"* pada kalimat berikut:

Kalimat 1: "They confirmed the location.",

Kalimat 2: "Here be the confirmed location."

```
Sentence: They confirmed the location
POS Tag : [('They', 'PRP'), ('confirmed', 'VBD'), ('the', 'DT'), ('location', 'NN')]
Lemmatized: They confirm the location

Sentence: Here be the confirmed location
POS Tag : [('Here', 'RB'), ('be', 'VB'), ('the', 'DT'), ('confirmed', 'JJ'), ('location', 'NN')]
Lemmatized: Here be the confirmed location
```

Gambar 3. 4 Perbandingan hasil lematisasi dari contoh kalimat

Gambar 3.4 menunjukkan hasil lematisasi dengan proses pos tag pada 1 kalimat utuh. Pada kalimat pertama kata ini berfungsi sebagai *verb* atau kata kerja. Sedangkan pada kalimat kedua, kata "*confirmed*" berfungsi sebagai kata *adjective* atau kata sifat. Jika proses lematisasi kata dilakukan setelah tokenisasi kata, maka kata di atas akan mempunyai fungsi yang sama. Setelah diketahui pos tag dari masing-masing kata yang sudah dilematisasi, selanjutnya yakni mengambil atribut fungsi kata untuk proses penghitungan kesamaan makna.

Berikut adalah pseudocode fungsi cek_*postag* untuk proses pengambilan atribut fungsi kata.

```
FUNCTION cek_postag(kata):
    IF kata starts with 'J' THEN
        RETURN ADJECTIVE      // a or s
    ELSE IF kata starts with 'V' THEN
        RETURN VERB           // v
    ELSE IF kata starts with 'N' THEN
        RETURN NOUN           // n
    ELSE IF kata starts with 'R' THEN
        RETURN ADVERB         // r
    ELSE
        RETURN NOUN           // default
END FUNCTION
```

Sentence Tokenizing menggunakan *library* dari nltk karena jika menggunakan modul yang ada di python akhir tanda baca berupa tanda titik tidak dipisahkan.

3. Punctuation Removal

Punctuation removal adalah proses menghilangkan tanda baca dari sebuah teks, seperti titik, koma, tanda tanya, dan tanda seru, sering kali dianggap tidak relevan atau diabaikan dalam analisis teks tertentu.

4. *Lemmatization dan Pos tagging*

Dua proses ini saling berhubungan. Seperti yang dijelaskan pada tahap 3, proses ini memproses sebuah kalimat agar menghasilkan pos tagging yang sesuai. Proses Lematisasi digunakan untuk mengembalikan kata yang berimbuhan ke bentuk dasarnya. Setiap bahasa akan berbeda perlakuannya. Tujuannya adalah agar model yang digunakan dalam penelitian ini semakin mendukung proses komputasi kesamaan kontekstual. Proses pos tagging akan dijelaskan pada

5. *Remove Whitespace*

Sangat dimungkinkan dalam sebuah data terdapa spasi yang berlebih. Oleh karena itu, proses ini menjadi wajib untuk menjaga proses pembersihan data lebih maksimal.

6. *Stopword Removal*

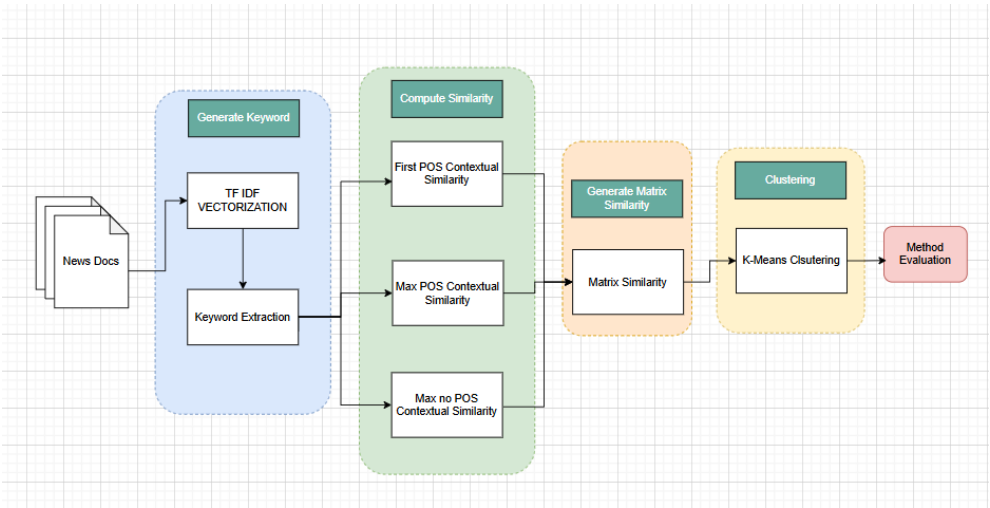
Pada tahap ini beberapa kata yang dianggap kurang penting dihapus (stoplist) dan menyimpan kata-kata yang dianggap penting (*wordlist*) dalam Bahasa Inggris seperti *that, and, or*. Dengan membuang *stoplist* dalam kumpulan *token*, kita bisa berfokus dengan kata-kata yang dianggap mempunyai informasi dalam sebuah teks dokumen.

Hasil dari tahap *preprocesing* ini adalah data yang bersih (*Clean Data*) yang siap untuk dilakukan komputasi pada tahap selanjutnya. Dan juga hasil dari pos tagging yang berisi semua pos dari kata 1 dataset. Agar komputasi selanjutnya

tidak terlalu rumit, *Pos maps* di filter berdasarkan *Clean Data* sehingga terbentuk *Clean Pos Maps* yang berisi data bersih beserta informasi POS-nya.

3.4 System Design

Desain sistem dalam penelitian ini digambarkan dengan diagram berikut.



Gambar 3. 5 Desain Sistem

Gambar 3.5 diatas menggambarkan bahwa sistem dalam penelitian ini dimulai dari pengambilan kandidat dari database sumber yang berisi dokumen berita berbahasa inggris yang pada tahap selanjutnya dilakukan *preprocessing* data. Data ini akan melalui proses *Fitur Extraction* yang terdiri dari proses Keyword extraction dan pembentukan matriks kesamaaan. Kemudian dilakukan proses clustering dengan K-Means.

3.5 Feature Extraction

Feature Extraction merupakan proses untuk menemukan ciri yang dimiliki suatu data. Di dalam penelitian ini, informasi semantik digunakan untuk mendapatkan informasi kesamaan kontekstual di dalam dokumen. Ada beberapa

tahap dalam proses ini sebelum data siap untuk diimplementasikan pada proses clustering.

3.5.1 TF-IDF Vectorization

TF-IDF dipilih karena model ini memperhitungkan pentingnya suatu kata dalam sebuah dokumen Ravi & Kulkarni (2023). Ini sangat penting dilakukan karena berhubungan dengan proses selanjunya yakni mengekstraksi kata-kata kunci yang ada di dalam suatu teks. Sangat tidak mungkin kita menghitung semua kata yang ada di dalam dokumen. Proses ini mengandalkan dua hal penting dari suatu dokumen, yakni frekuensi kata yang ada dalam suatu dokumen dan *inverse* dari jumlah kemunculan kata pada masing-masing dokumen, sehingga nilainya menggambarkan relevansi suatu kata kunci dengan dokumen tertentu Subakti *et al.*, (2022). Berikut adalah persamaan dari TF-IDF:

$$TF, IDF_{t\,d} = tf_{t\,d} \times \log\left(\frac{N}{df_t}\right)$$

keterangan:

$tf_{t\,d}$ = frekuensi dari *t* kata dalam dokumen *d*

N = jumlah dokumen yang ada

df_t = frekuensi dari dokumen yang mengandung kata *t*

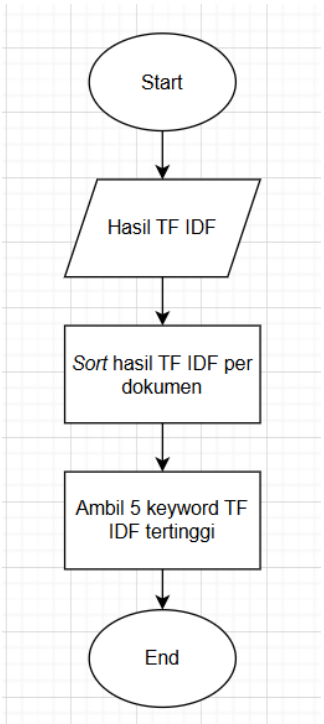
Pada penelitian ini menggunakan *library* dari *scikit-learn*. Pada penghitungan IDF di tambah dengan angka 1 untuk mencegah pembagian dengan angka 0, sehingga rumus untuk mencari IDF adalah sebagai berikut

$$IDF(t) = log\frac{N+1}{df(t)+1} + 1$$

3.5.2 Keyword Extraction

Pada proses ini kata-kata yang sudah diboboti dengan TF.IDF diseleksi dengan mempertimbangkan bobot tertinggi. Hanya kata-kata dengan bobot tertinggi saja yang dipakai untuk proses selanjutnya, karena kata-kata tersebut dianggap sebagai kata kunci yang relevan dengan dokumen tersebut Stecanella (2019).

Seleksi keyword ini dilakukan untuk mengurangi beban penghitungan dari TF-IDF. Berikut adalah *flowchart* untuk penentuan *keyword*:



Gambar 3. 6 Flowchart penentuan keyword terpilih

Pada gambar 3.6 Penentuan keyword terpilih dilakukan setelah proses perhitungan TF-IDF. Kemudian nilai TF-IDF tiap keyword diurutkan berdasarkan nilai TF-IDF tertinggi. 5 kata dengan nilai tertinggi di ambil sebagai kata yang mewakili dari setiap dokumen.

3.5.3 *Generate Similarity Matrix* berdasarkan Kesamaan Makna

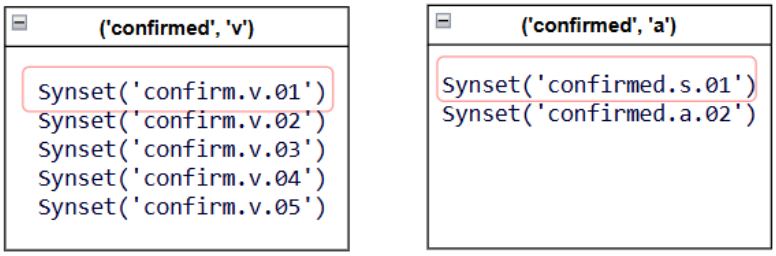
Untuk menghasilkan matriks kesamaan berdasarkan informasi kontekstual, maka digunakan *library* WordNet sebagai dataset untuk membangkitkan informasi semantik per kata.

Hasil *preprocessing* teks, salah satunya berupa *Clean Pos Maps*, yakni kumpulan token perkata dari sebuah dataset beserta atribut POS-nya. Untuk menangkap informasi POS ketika penghitungan kesamaan dengan Wu-Palmer, maka harus dibuat *Dictionary* yang menampung semua *Clean POS Maps* pada semua dataset yang diinisiasi dengan nama Global POS. Dictionary ini mengakomodasi pengambilan POS untuk fitur term. Untuk data 5 *keyword* per dataset cukup dengan pemanggilan *clean pos* sesuai urutan perhitungan dataset. Kedua jenis perlakukan POS Maps ini akan digunakan dalam tahap penghitungan kesamaan makna dengan 3 model yang akan dijadikan skenario uji coba pada penelitian ini, yakni *First* POS, *Max* POS, dan *Max No* POS.

3.5.3.1 Model First Part-of-Speech (First POS)

First POS mengambil *synsets* di dalam *library WordNet* pada *synset* pertama berdasarkan atribut POS yang melekat pada kata tersebut sesuai hasil pemetaan POS pada tahap *preprocessing*.

Diambil contoh yang sama di dalam proses sub-bab 3.3 *Data engineering*. Kata *confirmed* pada kalimat contoh pertama berfungsi sebagai *verb ('confirm', 'v').* Dan pada kalimat kedua berfungsi sebagai *adjective ('confirmed', 'a').* Jika dilihat hasil *synsets*-nya, maka akan tergambar sebagai berikut.



Gambar 3. 7. Gambaran pengambilan *synsets* WordNet model First POS

Gambar 3.7 menunjukkan perbedaan hasil pengambilan *synsets* pada kata yang sama, namun dengan atribut POS yang berbeda. Untuk model *First* POS, *synset* yang diambil ditunjukan dengan *synset* yang diberi tanda untuk kemudian dihitung nilai kemiripannya dengan metode Wu-Palmer.

Keuntungan dari model ini antara lain:

1. Efisiensi Komputasi
2. Dengan menggunakan model ini, penghitungan kesamaan akan lebih cepat karena hanya satu kata yang diproses.
3. Mempertahankan representasi kontekstual awal

4. Mempertahankan informasi *Part-of-Speech* dari setiap kata meskipun pengambilan pada posisi pertama
5. Konsistensi antar dokumen
6. Perbandingan dokumen menjadi seragam, karena masing-masing diambil dari posisi POS yang sama.

Namun demikian, keterbatasan model ini adalah tidak bisa menangkap semua informasi yang disediakan oleh *Synsets WordNet*. Sehingga, jika sysnet pertama kurang relevan dengan kontekstual, maka pengukuran kemiripan menjadi sedikit kurang tepat.

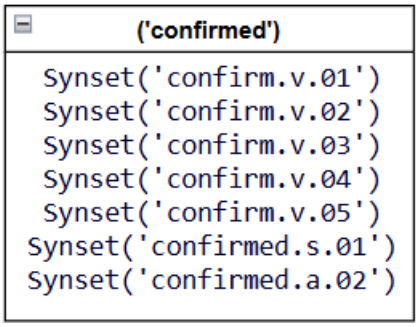
3.5.3.2 Model Max Part-of-Speech (Max POS)

Model ini hampir sama seperti model *First* POS pada langkah awalnya. Akan tetapi, nilai kemiripan antar kata diambil berdasarkan kemiripan tertinggi setelah proses perhitungan kesamaan antar synsets sesuai dengan POS-nya.

Cara ini mempertimbangkan semua kemungkinan kombinasi synset pada suatu kata sesuai dengan POS-nya. Keunggulan dari model ini adalah semua informasi dari *synsets* yang tersedia dihitung dan dipertimbangan untuk menjadi nilai kemiripan, bukan hanya synset pertama. Kelemahannya, komputasi menjadi semakin berat karena setiap kata terkadang mempunyai *sysnsets* yang banyak, meskpun sudah melalui proses *pos tagging*.

3.5.3.3 Model Max No Part-of-Speech (Max No POS)

Model terakhir ini, tidak menggunakan POS dari sebuah kata. Dia akan mengambil semua *synsets* dari suatu kata dan melakukan perhitungan kemiripan antar kata pada setiap synsets yang dimiliki. Berikut contoh semua *synsets* dari contoh kata ”*confirmed*”.



Gambar 3. 8. Contoh semua *synsets* kata tanpa atribut POS

Gambar 3.8 menunjukan semua *synsets* pada kata “*confirmed*” yang terdiri dari kata kerja (*verb*) dan kata sifat (*adjective*). Pada model ini semua informasi dari suatu kata dipertimbangkan. Tujuannya adalah meningkatkan cakupan semantik suatu kata, dan mencoba menangkap informasi kemiripan konteks yang lebih luas. Namun demikian, komputasi menjadi semakin berat dari pada dua model sebelumnya.

Model ujicoba yang dijelaskan diatas menggunakan perhitungan kesamaan Wu-Palmer. Berikut persaman wu-palmer yang dimaksud (Wu & Palmer, 1994).

$$sim_{wup}(s1\ s2) = 2 \times \frac{depth(lcs(s1\ s2))}{(depth(s1) + depth(s2))}$$

dimana:

s1, s2 = *synset* di dalam WordNet

depth(lcs) = kedalaman dari Least Common Subsumer

LCS menjelaskan konsep simpul leluhur terendah dalam taksonomi WordNet. Sebagai contoh, LCS dari *sea* dan *river* adalah body of water. Berikut contoh perhitungan:

Depth(LCS(*body_of_water*)) = 5

Depth1 = 6 (body_of_water, sea)

Depth2 = 7 (body_of_water, stream, river)

Maka,

Sim = (2 x 5) / (6+7) = 0, 769

Setiap dokumen akan menghasilkan 5 kesamaan pada setiap term fitur. Kemudian dihitung rata-rata kesamaan dan standar deviasi. Setelah itu matrik similarity mulai dibentuk dari hasil pengurangan rata-rata kesamaan dikurangi standar deviasi.

$$Sim = Mean - Std$$

Nilai kemiripan inilah yang menjadi nilai fitur term ke-*i* pada dataset tertentu. Setelah semua perhitungan kesamaan dilakukan, maka matriks *similarity* terbentuk.

3.5.4 Penskalaan Data

Beberapa hasil dari pembentukan matriks *similarity* bernilai *minus*. Hal ini dikarenakan rata-rata lebih kecil dari pada standar deviasi. Sehingga penskalaan data matriks ini harus dilakukan agar nilai berikas antara 0 dan 1. Penskalaan ini dilakukan untuk menyamakan skala fitur dan mencegah dominasi nilai ekstrim. Oleh karena itu penskalaan dilakukan dengan menggunakan *min-max scaler* Aggarwal (2015).

Diketahui min_j dan max_j merepresentasikan nilai minimal dan maksimal dari Atribut ke-j. Kemudian, nilai atribut ke-j (x_{ij}) dari *record* ke-i (X_i) dapat diskalakan dengan mengggunakan persamaan sebagai berikut

$$y_i^j = \frac{x_i^j - min_j}{max_j - min_j}$$

Menurut Aggarwal (2015) komputasi ini tidak akan efektif jika jarak nilai *max* dan *min* terlalu jauh. Misal, data usia yang harusnya ditulis 70 menjadi 700 karena kesalahan pengetikan.

3.6 Pengelompokan Menggunakan K-Means Clustering

Setelah representasi vektor matrik kesamaan didapatkan. Maka selanjutnya yakni proses perhitungan dengan metode *clustering* menggunakan Algoritma K-Means. Metide ini tergolong dalam jenis *Partitional Clustering* Bouras & Tsogkas (2012). K-Means merupakan salah satu algoritma pengelompokan yang paling sederhana dan sangat popular yang diusulkan oleh McQueen pada tahun 1967 Et-taleby *et al.*, (2020). Ide dasar dari algoritma ini adalah untuk meminimalkan error antara objek data dengan objek centroid yang terbentuk dengan memanfaatkan

informasi dari persamaan *Sum of Squarred Error (SSE)* menurut Suyanto (2017). SSE ini juga nantinya sebagai dasar untuk menghitung jumlah cluster yang optimal dengan metode *Elbow*.

Berikut adalah langkah-langkah dalam algoritma K-Means Clustering:

1. Menentukan jumlah *k* Cluster
2. Inisiasi nilai *centroid* awal, biasanya memakai bilangan acak. Pada penelitian ini menggunakan library Kmeans dari sklearn dengan *random_state*=42 dan *n_init* =10. Setelah dilakukan ujicoba dengan pengaturan seperti ini, hasil cluster menjadi konsisten.
3. Menghitung jarak kedekatan nilai centroid terhadap masing-masing data input. Pengukuran jarak yang paling populer adalah Euclidean distance. Anggap $i = (x_{i1}\ x_{i2}\ \dots\ x_{ip})$ dan $j = (x_{j1}\ x_{j2}\ \dots\ x_{jp})$ adalah dua objek yang mempunya atribut sebanyak *p*. Maka persamaan menjadi sebagai berikut Han et al. (2012b):

   $$d(i\ j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

   $$= \sqrt{\sum_{p=1}^{n}(x_{ip} - x_{jp})^2}$$

   $d(i\ j)$= jarak antara objek i dan j

   n = Jumlah atribut

   x_{ip}= nilai atribut input ke-p setiap anggota x_i

x_{jp}= nilai centroid ke-p setiap anggota x_j

4. Mengelompokan setiap data input berdasarkan jaraknya terhadap *centroid* (jarak terkecil)

5. Meng-*update* nilai centroid baru dengan rumus:

$$C_{ij} = \frac{1}{N_i}\sum_{k=0}^{N_i} x_{kj}$$

C_{ij} = centroid cluster ke-i variabel ke-j

N_i = Jumlah data yang menjadi anggota cluster ke-i

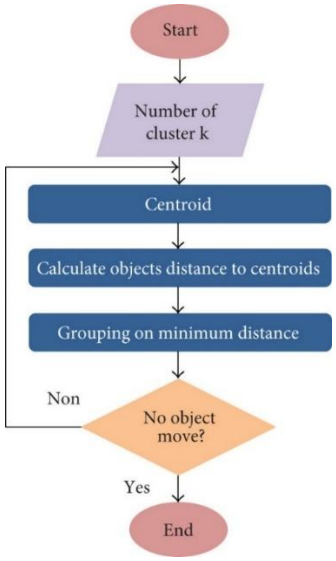
i = Indeks dari Cluster

j = Indeks dari variabel

x_{kj} = nilai data ke-k yang ada pada cluster tersebut untuk variabel ke-j

6. Melakukan iterasi dari langkah 3-5, sampai dengan kondisi konvergen atau nilai semua anggota tidak ada yang berubah.

Berikut adalah flowchart dari algoritma K-Means Clustering Et-taleby *et al.*, (2020):



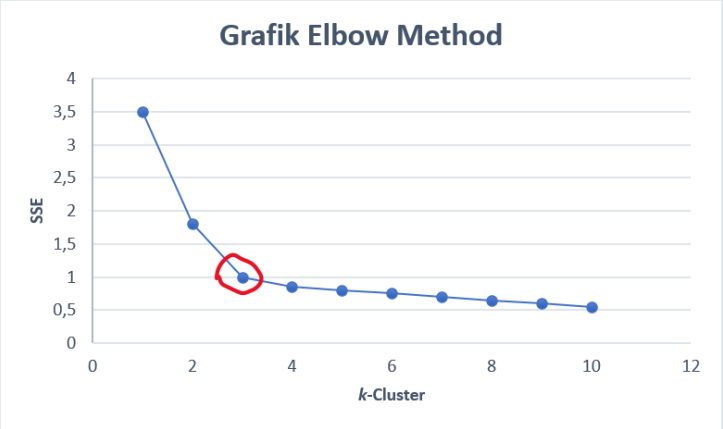
Gambar 3. 9 Flowchart K-Means (Et-taleby et al., 2020)

Gambar 3.9 menunjukkan bahwa iterasi perubahan *centroid* akan berhenti jika objek pada masing-masing *cluster* tidak ada yang berpindah pada *cluster* lain.

Hal pertama yang dilakukan pada algoritma K-Means yaitu menentukan jumlah *cluster* yang optimal. Namun, jumlah *cluster* yang berbeda biasanya menghasilkan performa yang berbeda. Maka dibutuhkan suatu metode untuk mendapatkan jumlah *cluster* yang optimal. *Elbow Method* adalah metode yang paling popular digunakan untuk mendapatkan jumlah *cluster* yang optimal. Dia bekerja dengan memanfaatkan perhitungan *Sum of Squarred Error* (SSE) yang dibandingkan dengan nilai *k* yang terus bertambah setiap iterasi. Berikut adalah Rumus persamaan SSE menurut Han *et al.*, (2012b):

$$SSE = \sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij}^{p}\, dist(o_i\ c_j)^2$$

Semakin besar nilai k, maka nilai SSE akan semakin kecil. Nilai k-Cluster dimulai dari k = 1 sampai dengan k = n. w_{ij}^{p} merupakan keanggotaan dari suatu cluster yang bernilai biner. Jika $w_{ij} > 0$, maka dilakukan penghitungan untuk $dist(o_i,c_j)$. Penghitungan SSE melibatkan anggota dari satu cluster dan centroid dari cluster itu sendiri. Tidak melibatkan anggota dari cluster atau *centroid cluster* lain, sehingga bisa dipastikan yang terlibat dalam penghitungan ini adalah anggota dari cluster.



Gambar 3. 10 Titik siku elbow method

Gambar 3.10 menunjukan bahwa titik siku dari perbanding nilai k terhadap nilai SSE. Garis landai selanjutnya menggambarkan perubahan nilai SSE yang tidak siginifikan. Maka diambilah nilai k pada titik siku untuk jumlah *cluster* yang paling optimal.

Tahap-tahap yang dilakukan pada *Elbow Method*:

1. Inisiasi nilai awal untuk k
2. Meningkatkan nilai k

3. Menghitung SSE untuk setiap nilai k
4. Mengamati penurunan nilai SSE pada grafik yang signifikan pada setiap nilai k
5. Mengambil nilai *k* pada grafik yang berbentuk siku atau atau terlihat menuju garis lurus.

3.7 System Implementation

Pada penelitian ini eksperiman dilakukan pada komputer dengan spesifikasi mesin yang ditunjukkan pada tabel berikut:

RAM (Random Access Memory)	16384 MB (16 GB)
CPU	11th Gen Intel (R) Core (TM) i7-11600H @ 2.90 GHz (12 CPUs), ~2.9 GHz
Windows	Windows 11 Home Single Language 64-bit
Harddisk	500GB SSD

Implementasi pada penelitian ini dilakukan pada *code editor Visual Stuido Code* yang sudah didukung dengan *Jupyter notebook*. Dilansir dari portal *website*-nya *jupyter.org*, *jupyter notebook* merupakan aplikasi web untuk membuat dan berbagi dokumen komputasi. Tampilan dan cara kerjanya hampir sama dengan *google collabs*. Perbedaannya adalah *jupyter notebook* terdapat versi yang bisa digunakan secara *offline*.

Database sqlite3 digunakan untuk menyimpan data matriks *similairity* dan *similarity_cache* agar untuk mempercepat penghitungan kesamaan makna.

3.8 Experiment

Experimen Dari dataset yang dipakai, penelitian dilakukan dengan menggunakan kolom *title* dan *description* dari dataset yang dipakai. Jumlah dokumen yang dipakai adalah 1000 artikel dari total 56134 artikel. Limitasi dokumen pada penelitian ini bertujuan untuk menjaga beban komputasi tetap terkelola, terutama pada proses pembangkitan matriks kemiripan untuk model Max POS dan Max No POS. Kedua model tersebut memerlukan perhitungan synset yang sangat intensif, sehingga waktu proses meningkat secara signifikan seiring bertambahnya jumlah data.

	article_id	source_name	author	title	description
0	93416	Business Insider	Tom Carter	WeWork's inevitable retreat is here	WeWork has started to close some of its cowork...
1	93415	Business Insider	Jennifer Ortakales Dawkins	TJMaxx is quietly closing stores in New York a...	The closures in New York and Chicago come as p...

Gambar 3. 11 kolom dataset *title* dan *description*

Gambar 3.11 menunjukan data yang dipakai untuk penelitian ini adalah gabungan dari kolom *title* dan *description*. Skenario yang dipakai dalam pengujian ini berdasarkan pada proses pengambil informasi semantik pada *library WordNet.*

Tabel 3. 3 Skenario Uji Coba

Kode	Skenario	Kategori	Distribusi per cat	Total data	Keterangan singkat
FP-20	*First POS Similarity*	20	*50 doc / cat (balanced)*	1000	*Synset* pertama berdasarkan POS
MP	*Max POS*	20	*50 doc / cat*	1000	Ambil *similarity* maksimum

Kode	Skenario	Kategori	Distribusi per cat	Total data	Keterangan singkat
-20	*Similarity*		*(balanced)*		antar synset (dengan POS)
MN-20	*Max No POS Similarity*	20	*50 doc / cat (balanced)*	1000	Ambil *similarity* maksimum tanpa POS
FP-10	*First POS Similarity*	10	*100 doc / cat (balanced)*	1000	*Synset* pertama berdasarkan POS
MP-10	*Max POS Similarity*	10	*100 doc / cat (balanced)*	1000	Ambil *similarity* maksimum antar synset (dengan POS)
MN-10	*Max No POS Similarity*	10	*100 doc / cat (balanced)*	1000	Ambil *similarity* maksimum tanpa POS
FP-5	*First POS Similarity*	5	*30: 30: 20: 10: 10 (unbalanced)*	1000	*Synset* pertama berdasarkan POS
MP-5	*Max POS Similarity*	5	*30: 30: 20: 10: 10 (unbalanced)*	1000	Ambil *similarity* maksimum antar synset (dengan POS)
MN-5	*Max No POS Similarity*	5	*30: 30: 20: 10: 10 (unbalanced)*	1000	Ambil *similarity* maksimum tanpa POS

Pada eksperimen ini, total data yang digunakan adalah 1000 dokumen untuk setiap skenario. Tabel 3.3 menunjukkan variasi skenario dibedakan berdasarkan jumlah kategori yaitu 20, 10, dan 5. Distribusi data juga divariasikan menjadi seimbang (*balanced*) dan tidak seimbang (*unbalanced*) untuk menguji stabilitas metode clustering dalam kondisi berbeda. Tiga pendekatan perhitungan kesamaan *WordNet* digunakan untuk mengevaluasi pengaruh strategi pemilihan *synset* terhadap kualitas *clustering*.

3.9 Metode Evaluasi

Untuk mengevaluasi kinerja metode dari hasil eksperimen yang dilakukan, maka penelitian ini menggunakan metode evaluasi secara internal, yakni Silhouette Coefficient dan David-Bouldin Index.

3.9.1 Silhouette Coefficient (SC)

Model ini mengukur sejauh mana objek dengan kelompoknya sendiri dibanding dengan kelompok lain yang berdekatan. Pertama kali diperkenalkan

oleh Rousseeuw, (1987). Nilai untuk metode ini berkisar antara -1 sampai dengan 1. Semakin tinggi nilanya maka hasil pengelompokan semakin baik. Berikut adalah persamaan *Silhouette Coefficient* yang dirangkum dari buku yang ditulis oleh Žižka *et al.*, (2020) dan paper asli Rousseeuw (1987):

Untuk setiap objek *i* yang berada pada dataset D. Kemudian dibagi menjadi beberapa partisi *cluster*. Dicontohkan A adalah cluster awal yang di hitung sebagai sampel dan C adalah salah satu cluster lain. Maka persamaan menjadi seperti berikut:

$$s(i) = \left\{ \begin{matrix} 1 - \dfrac{a_i}{b_i} & if\ a_i < b_i \\ 0 & if\ a_i = b_i \\ \dfrac{b_i}{a_i} - 1 & if\ a_i > b_i \end{matrix} \right\} = \frac{b_i - a_i}{\max\{a_i\ b_i\}}$$

dimana:

a_i = rata-rata ketidaksamaan (jarak) objek *i* dengan semua objek yang ada dalam satu *cluster* A

$d_{i\,C}$ = rata-rata ketidaksamaan (jarak) objek *i* dengan semua objek yang ada dalam cluster lain C dimana A ≠ C

b_i = nilai minimum dari $d_{i\,C}$

Rumus untuk mencari nilai a_i dan b_i dalam Han *et al.*, (2012) ditulis sebagai berikut:

$$a_i = \frac{\sum dist\ (i\ i`)_{i` \epsilon\ A \quad i \neq i`}}{|A| - 1}$$

$$b_i = \min\left\{\frac{\sum dist\ (i\ i`)_{i` \epsilon\ C \quad i \neq i`}}{|C|}\right\}$$

$|A|$= jumlah anggota pada cluster A

$|C|$= jumlah anggota pada cluster C

i = objek atau anggota cluster

$i`$ = objek atau anggota cluster selain i

3.9.2 Davies-Bouldin Index (DBI)

Model ini mengukur seberapa jelas dan terpisahnya setiap cluster yang terjadi. Pertama kali diperkenalkan oleh Davies & Bouldin (1979). Semakin rendah nilai index maka semakin baik pengelompokan yang terjadi Vergani & Binaghi(2018).

Berikut adalah persamaan yang dipakai untuk menghitung DB index Davies & Bouldin (1979):

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N} R_i$$

dimana:

$\bar{R}$ = rata-rata keseluruhan ukuran kesamaan dari setiap cluster dengan cluster yang paling mirip.

N = jumlah cluster

R_i= nilai maksimum dari $R_{ij}\ i \neq j$

R_{ij} adalah perbandingan antara rata-rata kesamaan setiap cluster dengan cluster yang paling mirip dan jarak antar centroid cluster. Untuk mengitung R_{ij}, maka menggunakan persamaan berikut:

$$R_{i\,j} = \frac{S_i + S_j}{M_{ij}}$$

dimana S_i dan S_j merupakan penyebaran dari cluster *i* dan *j* dengan persamaan sebagai berikut:

$$S_i = \left\{ \frac{1}{T_i} \sum_{j=1}^{T_i} \left| X_j - A_i \right|^q \right\}^{\frac{1}{q}}$$

dimana:

T_i = jumlah vektor dari cluster i

X_j = vektor dalam i

A_i = Centroid dari Cluster i

Ketika *q*=1, maka S_i adalah rata-rata jarak Euclidean dari anggota vektor ke centroid *i*. Jika *q*=2, maka S_i adalah Standar Deviasi metrik pada sampel dalam *cluster* sehubungan dengan centroidnya.

Kemudian menghitung jarak antar Centroid menggunakan persamaan berikut:

$$M_{i\,j} = \left\{ \sum_{k=1}^{N} \left| a_{ki} - a_{kj} \right|^p \right\}^{\frac{1}{p}}$$

dimana:

a_{ki} = anggota ke-*k* dari vektor a_i yang merupakan centroid dari cluster i

a_{kj} = anggota ke-*k* dari vektor a_j yang merupakan centroid dari cluster j

N = jumlah anggota vektor cluster.

$M_{i\,j}$ adalah metrik Minkowski dari centroid *i* dan *j*. Ketika p=1, maka disebut sebagai jarak Manhattan, dan ketika p=2, maka disebut jarak Euclidean. Perlu digarisbawahi bahwa jika nilai $p = q = 2$, maka $R_{i\,j}$ adalah ukuran kesamaan *Fisher* yang dihitung antara cluster *i* dan cluster j.

3.10 Instrumen Penelitian

Sistem yang akan dibangun menggunakan beberapa variable yang disebut dengan variable bebas, terikat dan intervening. Variabel bebas merupakan variable yang menjadi input dari model yang diusulkan. Varibael bebas disini merupakan teks dokumen yang diekstrak menjadi bobot pada tahap feature extraction, yakni matriks kesamaan. Selanjutnya adalah variabel *intervening* yang merupakan *output* dari proses *clustering* yakni berupa data-data yang sudah dikelompokkan.

Tabel 3. 2 Variabel Penelitian

Independet Variable	**Proses**	**Intervening Variable**	**Dependent Variable**
Matriks Similaritas		*Cluster Result*	SC Score dan DBI

Tabel 3.2 menunjukkan keterkaitan antar variabel. Terakhir, yakni variabel terikat yang merupakan nilai hasil dari penghitungan evaluasi metode dengan menggunakan Silhouette Coefficient dan Davies-Boulin Index.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Desain Sistem

Dari 22.015 data, sistem melakukan *preprocesing* dengan beberapa perlakuan. Data yang dipakai adalah data *text*, yakni gabungan dari data kolom *title* dan *description.* Kemudian dilakukan proses tokenisasi kalimat pada data *text*. Lematisasi dilakukan pada token kalimat berkolabosari dengan *pos tagging*. Dari hasil ujicoba, proses *pos tagging* pada satu kalimat utuh dan langsung pada kata setelah tokenisasi kata memperoleh hasil pos *tag* yang berbeda seperti yang sudah dijelaskan pada subab 3.3. Hasil dari *preprocesing* adalah kumpulan data siap pakai berupa teks bersih (*clean text*) dan *clean pos maps,* yakni hasil pos maps semua token yang sudah di filter dengan *clean text*.

Berikut adalah gambaran teks sebelum dan sesuai preprocessing beserta POS Maps nya.

Tabel 4. 1 Hasil Preprocessing satu dataset

WeWork's inevitable retreat is here. WeWork has started to close some of its coworking spaces as it reportedly prepares to file for bankruptcy.	Teks asli dan Pos Maps
{'wework': 'n', 's': 'n', 'inevitable': 'a', 'retreat': 'n', 'be': 'v', 'here': 'r', 'have': 'v', 'start': 'v', 'to': 'n', 'close': 'v', 'some': 'n', 'of': 'n', 'it': 'n', 'coworking': 'n', 'space': 'n', 'a': 'n', 'reportedly': 'r', 'prepare': 'v', 'file': 'v', 'for': 'n', 'bankruptcy': 'n'}	
wework inevitable retreat wework start close coworking space reportedly prepare file bankruptcy	Teks *Preprocessed* dan Clean POS Maps
{'wework': 'n', 'inevitable': 'a', 'retreat': 'n', 'start': 'v', 'close': 'v', 'coworking': 'n', 'space': 'n', 'reportedly': 'r', 'prepare': 'v', 'file': 'v', 'bankruptcy': 'n'}	

Tabel 4.1 menunnjukkan hasil *preprocessing* dan data Clean POS Maps yang merupakan data pos maps difilter dengan kolom bersih. yang merupakan data pos maps yang difilter dengan kolom bersih. dari satu data.

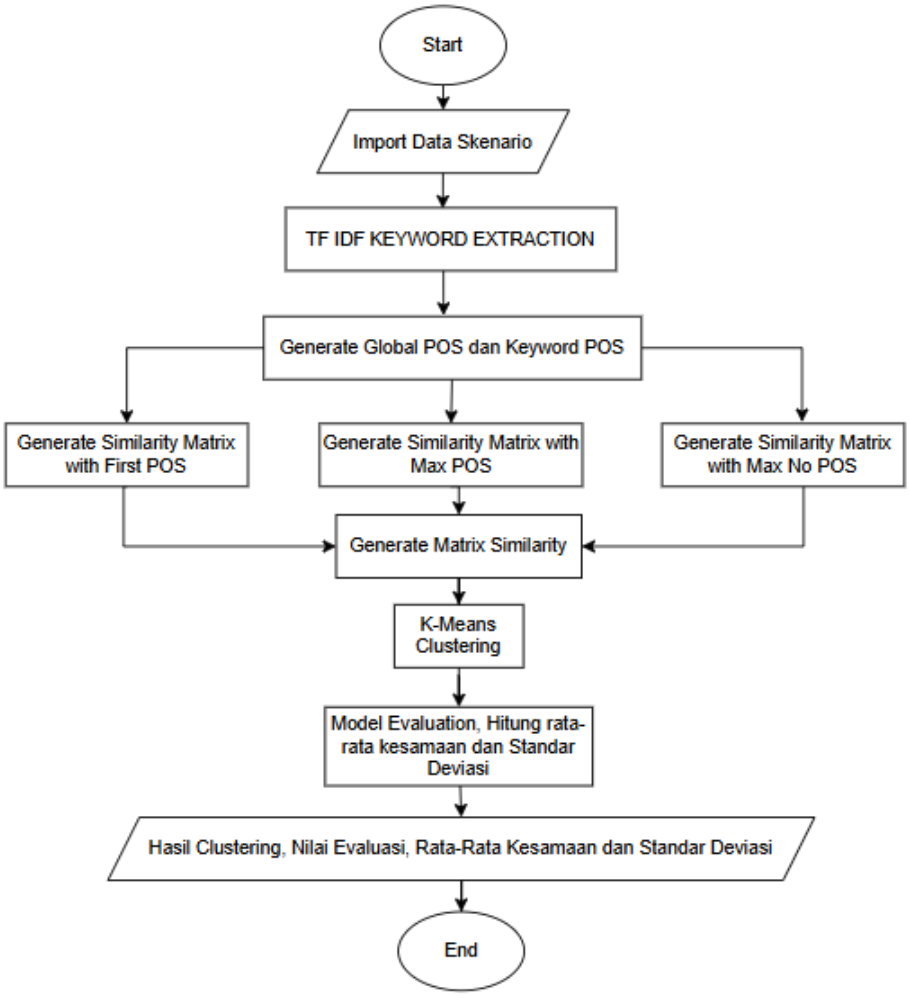
Tabel 4. 2 Hasil *Preprocessing* dan *Clean POS Maps*

article_id	**source_name**	**author**	**title**	**text**	**bersih**	**pos_maps**	**clean_pos_maps**
93416	Business Insider	Tom Carter	*WeWork's inevitable retreat is here*	WeWork has started to close …	wework inevitable retreat wework …	{'wework': 'n', 's': 'n', 'inevitable': 'a', 'retreat': 'n', …}	{'wework': 'n', 'inevitable': 'a', 'retreat': …
93415	Business Insider	Jennifer O. Dawkins	*TJMaxx is quietly closing stores …*	TJX, the parent company of TJ Maxx …	tjmaxx quietly close store new york …	{'tjmaxx': 'n', 'be': 'v', 'quietly': 'r', 'close': 'v', …}	{'tjmaxx': 'n', 'quietly': 'r', 'close': 'v', …
93417	Boing Boing	Jennifer Sandlin	*Sisters win workplace Halloween costu…*	Sisters win workplace Halloween …	sister win workplace halloween …	{'sister': 'n', 'win': 'v', 'workplace': 'n', 'halloween': 'a', }	{'sister': 'n', 'win': 'v', 'workplace': 'n', …

Tabel 4.2 menunjukkan hasil dari *preprocessing* yang menghasilkan teks *preprocessed* pada kolom bersih, dan clean pos maps secara keseluruhan.

Selanjutnya, data melalui proses penyaringan tambahan untuk menghilangkan entri yang tidak terdiri dari karakter alfabet atau tidak menggunakan bahasa Inggris dan kolom *description* yang berisi "*No description*".

Tahap ini memastikan bahwa hanya teks yang valid dan relevan yang dipertahankan, sehingga diperoleh 21.495 dataset untuk proses analisis berikutnya dan disimpan dengan nama file *news_clean.csv*, sehingga untuk proses selanjutnya tinggal memanggil data yang sudah bersih.



Gambar 4. 1 Desain Sistem Skenario Eksperimen

Gambar 4.1 menjelaskan pekerjaan yang dilakukan untuk menguji performa model. Pembentukan representasi teks berupa matriks kesamaan melibatkan synset berdasarkan POS kata dalam teks. Sistem dimulai dari import data yang sudah melalui tahap preprocessing yang berisi data Pos Maps dan Clean Pos

Maps. Data Pos Maps nantinya akan menjadi rujukan untuk mengambil pos pada term atau fitur. Kemudian, Clean Pos Maps akan menjadi rujukan Keyword per data untuk mengambil informasi POS nya. Di dalam penelitian ini juga dibahas tentang rata-rata kesamaan dalam 1 cluster yang diukur menggunakan Cosine similarity, karena sangat baik bekerja pada high dimensional data Lim *et al.,* (2021). Dari penelitiannya juga menyebutan bahwa Santhisree & Damodaram (2011) melakukan pengukuran mean similarity dan standar deviasi. Semakin tinggi nilai cosine maka kedekatan antar dokumen juga semakin tinggi. Semakin tinggi nilai kedekatan di dalam 1 cluster (*intra-cluster*) maka pengelompokan terbentuk secara bagus menuurut Lim *et al.*, (2021)

4.2 *Import* Data Skenario

Data yang digunakan untuk pengujian berjumlah 1000 data. Terdapat 3 jenis pemilahan kategori sesuai yang dijelaskan pada sub-bab 3.8.

Tabel 4. 3 Dataset 20 kategori *balanced*

No	Category	Jumlah Dokumen
1	Architecture	50
2	Artificial Intelligence	50
3	Beauty	50
4	Climate	50
5	Education	50
6	Fashion	50
7	Finance	50
8	Food	50
9	Health	50
10	Jobs	50
11	Movies	50
12	Music	50
13	Politics	50
14	Real Estate	50
15	Relationships	50

No	Category	Jumlah Dokumen
16	Science	50
17	Sports	50
18	Stock	50
19	Technology	50
20	Weather	50

Data pertama untuk scenario pengujian ditunjukkan pada tabel 4.3 dengan jumlah 1000 dataset. Masing-masing kategori berjumlah 50 data (*balanced*)

Tabel 4. 4 Dataset 10 katageri *balanced*

No	Category	Jumlah Dokumen
1	Education	100
2	Finance	100
3	Food	100
4	Health	100
5	Jobs	100
6	Real Estate	100
7	Science	100
8	Stock	100
9	Technology	100
10	Weather	100

Data kedua ditunjukkan pada Table 4.4 dengan 10 kategori. Masing-masing kategori berjumlah 100 data. Tergolong sama dengan Data uji pertama yakni dengan jumlah seimbang (*balanced*)

Tabel 4. 5 Dataset 5 Kategori *unbalanced*

No	Category	Jumlah Dokumen
1	Health	300
2	Stock	300
3	Technology	200
4	Finance	100
5	Real Estate	100

Tabel 4.5 menunjukkan persentase data uji yang berbeda. Jumlah dataset berbeda setiap kategorinya. Pengaturan data diatur dengan perbandingan 30: 30: 20: 10: 10.

4.3 TF-IDF Keyword Extraction

Hasil dari proses ini adalah Ekstraksi 5 Keyword dari setiap dataset berdasarkan nilai TF-IDF tertinggi.

Tabel 4. 6 Contoh hasil ekstraksi keyword menggunakan TF-IDF Data Uji 1

No	Dokumen	Top Keywords (TF-IDF)
0	accenture plc nyse acn share acquire cetera advisor network llc …	cetera (0.374), acn (0.364), accenture (0.364), network (0.326), plc (0.307)
1	casa system set cable industry milestone docsis …	casa (0.624), system (0.346), vccap (0.156), milestone (0.156), docsis (0.156)
2	nothing bring imessage android phone …	nothing (0.642), app (0.330), phone (0.321), imessage (0.189), bubble (0.189)
3	euronet partner banco pichincha modernize card processing …	pichincha (0.350), issuing (0.350), euronet (0.350), ecuador (0.350), banco (0.350)
4	amd compete intel anymore threadripper win …	threadripper (0.476), amd (0.456), insane (0.252), anymore (0.252), actually (0.252)
5	…	…

Dari Tabel 4.6. Diketahui hasil ekstraksi 5 keyword pada Data Uji pola pertama. Pada data uji ini diperoleh jumlah term sebanyak 6509. Sehingga dimensi data menjadi 1000 x 6509 untuk proses TF-IDF

Tabel 4. 7 Hasil Ekstraksi Keyword Data Uji 2

No	Dokumen	Top Keywords (TF-IDF Score)
0	unexpected friendship graveyard keeper polio survivor lampedusa …	polio (0.276), longtime (0.276), lampedusa (0.276), keeper (0.276), graveyard (0.276)
1	belarus link forcible transfer ukrainian child …	child (0.425), yale (0.287), forcible (0.287), belarus (0.287), transfer (0.271)
2	cnl strategic capital announces operate result third quarter …	cnl (0.532), strategic (0.430), capital (0.301), orlando (0.177), loa (0.177)

No	Dokumen	Top Keywords (TF-IDF Score)
3	congress make malicious attempt push country towards slavery yogi adityanath …	yogi (0.317), slavery (0.317), adityanath (0.317), attempt (0.287), towards (0.269)
4	virginia democrat win full control state legislature …	legislature (0.457), democrat (0.431), state (0.237), youngkin (0.228), counterweight (0.228)

Dari Tabel 4.7. Diketahui hasil ekstraksi 5 keyword pada Data Uji pola kedua. Pada data uji ini diperoleh jumlah term sebanyak 6003. Sehingga dimensi data menjadi 1000 x 6003 untuk proses TF-IDF

Tabel 4. 8 Ekstraksi Keyword Data Uji 3

No	Dokumen	Top Keywords (TF-IDF Score)
0	corefirst bank trust decrease stock holding visa inc …	visa (0.499), corefirst (0.477), trust (0.311), bank (0.307), decrease (0.297)
1	northern superior resource cve sup stock price …	sup (0.397), superior (0.357), resource (0.357), northern (0.357), cve (0.333)
2	vanguard total international stock etf nasdaq vxus holding raise bfsg llc …	vxus (0.456), bfsg (0.456), vanguard (0.348), international (0.286), etf (0.279)
3	principal financial group inc decrease stock holding lam research …	lrcx (0.454), lam (0.454), principal (0.410), financial (0.236), research (0.226)
4	goldman sachs group downgrade imperial oil tse imo …	imo (0.483), imperial (0.341), sachs (0.289), goldman (0.289), oil (0.275)

Dari Tabel 4.8. Diketahui hasil ekstraksi 5 keyword pada Data Uji pola ketiga. Pada data uji ini diperoleh jumlah term sebanyak 5190. Sehingga dimensi data menjadi 1000 x 5190 untuk proses TF-IDF.

Dari ketiga model data uji, perlakuan data *Out of Vocabulary* (OOV) dibiarkan. Reduksi data dilakukan dengan mengecek dan menghapus *Zero-Constant Feature* setelah proses generate matriks similarity.

4.4 Global POS dan Keyword POS

Sebelum menuju proses selanjutnya, dua bentuk POS Maps harus dibuat untuk menangani Data term dan data keyword dalam mengambil atribut POSnya.

```
global_pos={}
for cleanposmap in data_ujicoba['clean_pos_dict']:
 for word, pos in cleanposmap.items():
      global_pos[word]= [pos]


keyword_pos = data_ujicoba['clean_pos_dict']
```

Penggalan *source code* di atas akan menghasilkan *Dictionary global_pos* dan *Dataframe keyword_pos*.

4.5 Model *First POS Similarity*

Model First POS digunakan untuk proses *generate* matriks kesamaan yang tergolong paling cepat, karena hanya menghitung synset pertama yang dimiliki oleh sebuah kata. Pada praktiknya, dilakukan *caching* pasangan data yang sudah dihitung dan disimpan dalam database sqlite3 pada tabel similarity_cache_firspos agar tidak terjadi penghitungan yang berulang jika ada data yang sama. Tabel cache ini hanya menampung nilai kesamaan antara 0 dan 1. Hal ini bertujuan agar perhitungan lebih cepat. Nilai 1 muncul jika kedua kata yang dihitung memiliki sintaks yang persis meskipun kata itu termasuk OOV (*Out of Vocabulary*) dalam

WordNet. Ini dilakukan agar penghitungan tidak kehilangan informasi jika data tersebut dominan dan termasuk pada kategori OOV, seperti contoh nama aplikasi, negara atau pun orang. Kelemahannya, banyak data yang memang tidak memiliki arti tidak tereliminasi. Perlakuan ini berlaku pada semua model percobaan.

key TEXT	value REAL
ab.n|network.n	0,285714285714286
abc.n|network.n	0,25
abduct.v|network.n	0,2
ability.n|network.n	0,4
able.a|network.n	0,25
aboard.r|network.n	0,25
abortion.n|network.n	0,266666666666667
abroad.r|network.n	0,25
abruptly.r|network.n	0,25
absence.n|network.n	0,307692307692308
absolutely.r|network.n	0,25

Gambar 4. 2. Contoh data similarity cache dari model First POS

Dari Gambar 4.2 diatas, dapat diketahui bahwa sebelum data cache disimpan, dua kata yang dihitung dilakukan *sort* sehingga tidak terjadi *redundancy data.* Hal yang sama dilakukan ketika proses pencarian. Matriks yang terbentuk juga disimpan dalam database sqlite3 pada tabel similarity_matrix_first_pos.

doc_id # INTEGER	term ABC TEXT	sim # REAL
0	ability	-0,0988854381999832
1	ability	-0,049071198499986
2	ability	-0,0656908649368789
3	ability	-0,0473968728700587
4	ability	0,0130693606237084
5	ability	0,0127787389494361
6	ability	0,117088858175126
7	ability	-0,00885750454904752
8	ability	0,0137448156142219
9	ability	0,132705623463544

Gambar 4. 3 Contoh isi tabel similairty_matrix_firstpos

Pada gambar 4.3 diketahui bahwa data matriks pada posisi long ketika disimpan dalam database. Ketika proses clustering bentuk ini akan dijadikan model *wide matrix* dengan menggunakan *library Dataframe* sehingga penghitungan menjadi lebih mudah dan jelas. Sebelum digunakan untuk proses clustering, terlebih dahulu dilakuan pengecekan dan penghapusan data *features with low variance* pada data matrix similarity. Hal ini dilakukan karena data dengan variasi rendah seperti terdapat nilai 0 pada semua dataset, fitur tersebut tidak memberikan informasi yang berguna (Das & Mert Cakmak, 2018). Dilakukan juga proses penskalaan min max, karena terdapat data dengan nilai minus seperti yang dijelaskan pada sub-bab 3.5.4.

4.6 Model *Max POS Similarity*

Proses pembuatan matriks similarity dengan model ini, menimbulkan komputasi yang lebih berat dari pada model sebelumnya. Maka caching data sangat diperlukan. Pada model ini similarity cache disimpan di dalam database sqlite 3 pada tabel similarity_cache_maxpos yang terdiri dari kolom *key* dan

kolom *value,* bentuknya disamakan model Dictionary sehingga pencarian cache berdasarkan *key* yang sudah di *sorted* sama seperti proses pada model firstpos.

key TEXT	value REAL
ab.n|network.n	0,307692307692308
abc.n|network.n	0,25
abduct.v|network.n	0,2
ability.n|network.n	0,4
able.a|network.n	0,25
aboard.r|network.n	0,25
abortion.n|network.n	0,333333333333333
abroad.r|network.n	0,25
abruptly.r|network.n	0,25
absence.n|network.n	0,4

Gambar 4. 4. Similarity Cache Model Maxpos

Pada gambar 4.4 diketahui bahwa pola penyimpanan similarity cache sama seperti model firstpos. Hasnya saja nilai yang disimpan adalah nilai tertinggi dari penghitungan kesamaan antat synsets sesuai dengan POS-nya.

doc_id INTEGER	term TEXT	sim REAL
0	abduct	-0,0494427190999916
1	abduct	-0,030519937164544
2	abduct	-0,0283989839337304
3	abduct	-0,0215451212643219
4	abduct	0,0174258141649447
5	abduct	0,00798465157324231
6	abduct	0,134380351162499
7	abduct	0,00706001930385401

Gambar 4. 5 Similarity matriks Max POS

Dapat diketahui dari Gambar 4.5 bahwa pola penyimpanan sama yakni memakai long dengan tabel terdiri dari kolom *doc_id* yang berfungsi sebagai

indeks ketika dirubah ke bentuk wide. Kolom term yang befungsi sebagai fitur dan kolom nilai similarity.

4.7 Model *Max No POS Similarity*

Model Max No POS sangat berbeda dengan kedua Model sebelumnya. Model ini tidak tergantung hasil POS Tag, dan menghitung persamaan semua synsets pada suatu kata. Model ini mencoba mendapatkan informasi dari semua synsets dan mengambil nilai kemiripan tertinggi. Namun demikian, komputasi menjadi sangat berat pada awal penghitungan. Similarity cache sangat diperlukan. Perhitungan selanjutnya kadang menjadi semakin cepat, karena beberapa kata sudah terhitung dan tinggal memanggil pada tabel similarity_cache_maxnopos.

key ABC TEXT	value # REAL
ab|network	0,307692307692308
abc|network	0,25
abduct|network	0,222222222222222
ability|network	0,4
able|network	0,285714285714286
abo|network	0,6
aboard|network	0,285714285714286
abortion|network	0,333333333333333
abroad|network	0,285714285714286

Gambar 4. 6 Similarity Cache Max No POS

Dari gambar 4.6 dapat diketahui bahwa cara penyimpanann untuk similarity cache juga ada perbedaaan, yakni pada kolom *key,* data tidak membawa atribut POS-nya. Penghitungan nilai kemiripan hampir sama dengan model maxpos.

doc_id # INTEGER	term ABC TEXT	sim # REAL
0	abduct	-0,0549363545555462
1	abduct	-0,030519937164544
2	abduct	0,00682547707322256
3	abduct	-0,052726015334711
4	abduct	0,0675268683849718
5	abduct	0,0521189636766201

Gambar 4. 7. *Similarity Matrix Max No POS*

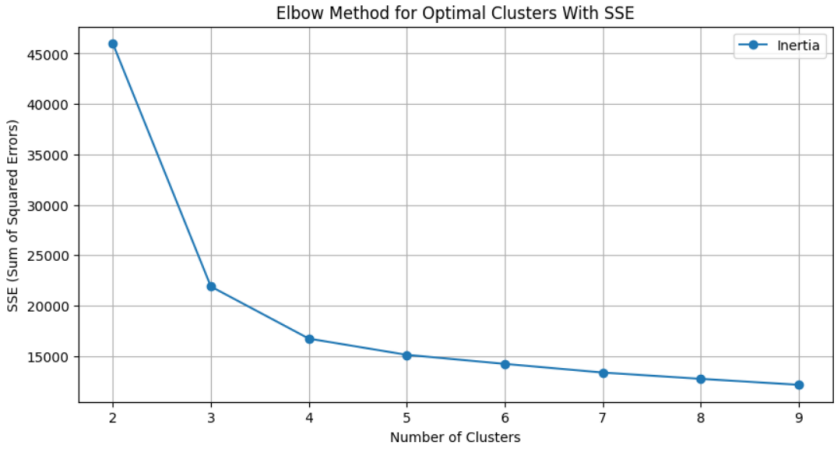
Dari gambar 4.7 bisa diketahui cara penyimpanan sama dengan model sebelumnya. Perlakuan sebelum masuk ke metode Clustering sama pada semua model.

4.8 Skenario Pengujian 20 Kategori (*Balanced*)

Skenario pengujian pertama dilakukan pada dataset berjumlah 1000 dengan 20 kategori. Setiap kategori berisi 50 dataset berita.

4.8.1 Ujicoba Data FP-20

Penentuan jumlah cluster optimal dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 8 Elbow Method FP-20

Gambar 4.8 menunjukkan bahwa berdasarkan hasil perhitungan SSE pada berbagai nilai k, tampak bahwa penurunan paling signifikan terjadi pada rentang k = 2 hingga k = 3. Setelah titik tersebut, penurunan SSE masih terjadi namun tidak lagi menunjukkan perubahan yang substansial. Dengan demikian, titik k = 3 dapat dipandang sebagai titik siku (elbow), karena pada titik inilah terjadi perubahan gradien terbesar yang menandai peralihan dari penurunan tajam menjadi penurunan yang lebih landai. Oleh karena itu, k = 3 dipilih sebagai jumlah cluster optimal pada skenario ini.

Tabel 4. 9. Hasil Clustering pada *k*=3 pada FP-20

Cluster	Jumlah Dokumen	Mean Similarity	Std. Deviation
0	203	0.9926	0.0051
1	507	0.9873	0.0085
2	290	0.9950	0.0025

Tabel 4.9 menunjukkan hasil proses clustering menggunakan nilai k optimal. Pada nilai k = 3, model menghasilkan tiga cluster dengan karakteristik yang berbeda. Cluster 0 berisi 203 dokumen dengan nilai mean similarity sebesar 0.9926 dan standard deviation 0.0051. Nilai tersebut menunjukkan bahwa dokumen dalam cluster ini memiliki tingkat kemiripan internal yang tinggi dan variasi konteks yang relatif kecil.

Cluster 1 merupakan cluster dengan jumlah dokumen terbesar, yaitu 507 dokumen. Mean similarity pada cluster ini adalah 0.9873 dengan standard deviation tertinggi, yaitu 0.0085. Hal ini mengindikasikan bahwa cluster ini

memiliki tingkat heterogenitas konteks yang lebih tinggi dibandingkan dua cluster lainnya, sehingga dokumen dalam cluster ini lebih beragam secara kontekstual.

Sementara itu, Cluster 2 berisi 290 dokumen dengan mean similarity tertinggi, yaitu 0.9949, serta standard deviation terendah (0.0025). Kondisi ini menunjukkan bahwa cluster tersebut memiliki tingkat keseragaman konteks paling tinggi, dengan dokumen-dokumen yang sangat mirip secara kontekstual. Secara keseluruhan, distribusi tersebut menunjukkan bahwa representasi fitur yang digunakan mampu memisahkan dokumen berdasarkan kedekatan konteks, meskipun terdapat variasi tingkat homogenitas antar cluster.

Hasil evaluasi dengan silhouette Score dan DBI disajikan dalam tabel berikut:

Tabel 4. 10 Hasil Evaluasi pada k-cluster

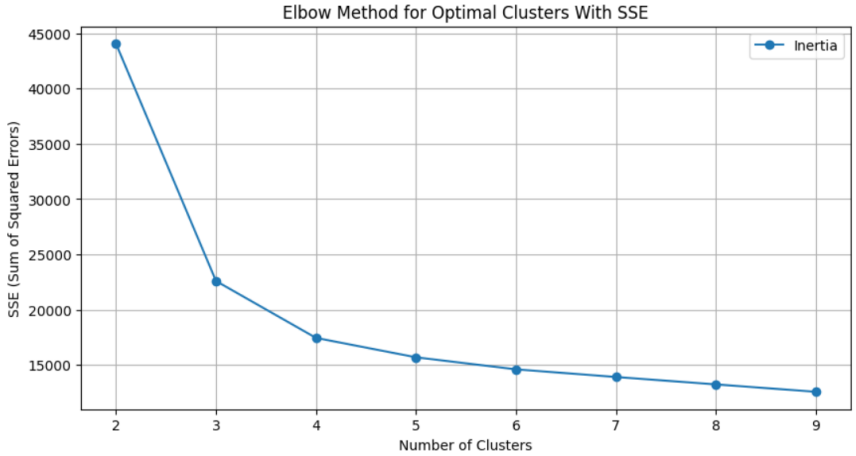
K	SSE	Silhouette	DBI
2	45,988.89	0.507	0.721
3	21,889.43	0.505	0.667
4	16,719.47	0.430	0.896
5	15,107.88	0.373	1.231
6	14,211.04	0.352	1.430
7	13,352.10	0.347	1.431
8	12,727.89	0.249	1.663
9	12,136.79	0.248	1.586

Dari tabel 4.10 menunjukkan bahwa Nilai Silhouette terbesar yakni berada pada k=2 dengan nilai 0.513. Pada saat k-optimal=3, nilai silhouette masih tergolong bagus yakni dengan nilai 0,505 dan masih bisa dijadikan acuan pengelompokan. Nilai silhouette pada k > 3 menurun cukup drastis, menunjukkan kualitas pemisahan antar cluster semakin buruk Selanjutnya nilai DBI terkecil

juga terletak pada *k*=3 dengan nilai 0.667, yang berarti struktur cluster paling kompak dan terpisah dengan baik dibanding nilai k lainnya.

4.8.2 Ujicoba Data MP-20

Penentuan jumlah cluster optimal dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 9 Titik Elbow pada Skenario MP-20

Dari Gambar 4.9 diketahui bahwa titik siku atau elbow terlihat pada k=3. Setelah titik k=3 garis elbow mulai melandai. Maka jumlah k Optimal adalah k=3.

Tabel 4. 11 Mean Sim. dan Std. MN-20

Cluster	Jumlah Dok.	*Mean Similarity*	*Std.*
Cluster 0	205	0.9899	0.0083
Cluster 1	288	0.9942	0.0033
Cluster 2	507	0.9871	0.0090

Tabel 4.11 menunjukkan 3 Cluster dengan Jumlah Data per cluster, *Mean Similarity* dan Standar Deviasinya. Cluster 1 memiliki nilai mean similarity paling tinggi (0.9942) dan standar deviasi paling kecil (0.0033), yang menunjukkan

bahwa dokumen pada cluster ini memiliki tingkat kemiripan kontekstual paling tinggi dan paling homogen. Cluster 2 merupakan cluster dengan jumlah dokumen terbanyak 507 dokumen, namun memiliki mean similarity terendah 0.9871 dan standar deviasi tertinggi 0.0090, mengindikasikan bahwa cluster ini lebih heterogen dan terdiri atas topik yang lebih bervariasi dibanding cluster lainnya. Cluster 0 berada di posisi tengah baik dari segi jumlah dokumen maupun tingkat homogenitas.

Evaluasi dilakukan dengan melihat nilai Silhouette dan DBI pada cluster optimal dan juga pada semua cluster.

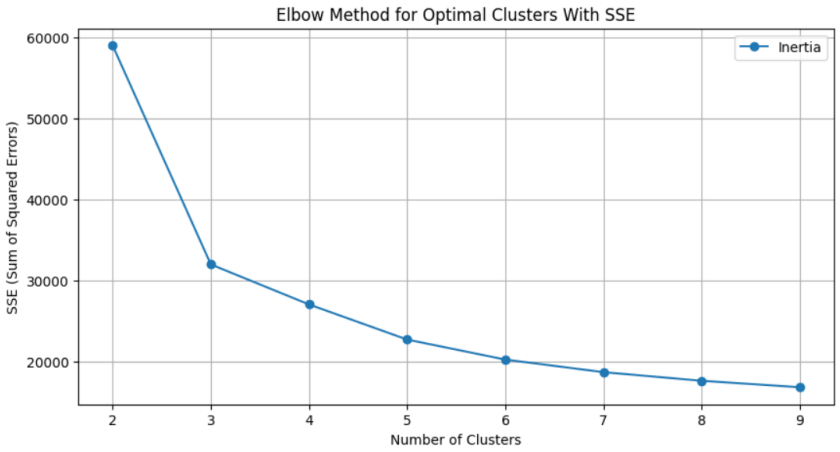
Tabel 4. 12 Nilai Silhouette dan DBI pada MP-20

k	SSE	Silhouette Score	DBI
2	44,056.77	0.508	0.721
3	22,583.77	0.497	0.688
4	17,436.84	0.423	0.884
5	15,680.92	0.370	1.177
6	14,591.28	0.371	1.255
7	13,898.17	0.348	1.415
8	13,227.71	0.269	1.528
9	12,558.75	0.247	1.572

Dari table 4.12 diketahui meskipun Nilai silhouette terbesar pada k=2, namun pada k=3, nilai DBI adalah terendah. Nilai silhouette pada k=3 juga masih tergolong bagus. Maka jumlah cluster memang optimal pada k=3.

4.8.3 Ujicoba Data MN-20

Penentuan jumlah cluster optimal dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 10 Metode Elbow pada model MN-20

Dari Gambar 4.10 diketahui bahwa titik siku atau elbow terlihat pada k=3. Setelah titik k=3 garis elbow mulai melandai. Maka jumlah k Optimal adalah k=3.

Tabel 4. 13 Nilai Mean Sim. Dan Std pada MN-20

Cluster	Jumlah Dokumen	Mean Similarity	Std. Deviation
Cluster 0	282	0.9921	0.0044
Cluster 1	389	0.9836	0.0129
Cluster 2	329	0.9856	0.0093

Dari tabel 4.13. diketahui bahwa Cluster 0 merupakan cluster dengan tingkat kemiripan kontekstual tertinggi, ditunjukkan oleh mean similarity 0.9921 dan std deviation 0.0044 yang paling rendah. Hal ini menandakan bahwa dokumen dalam cluster ini sangat homogen atau memiliki kesamaan konteks yang kuat. Cluster 1 memiliki jumlah dokumen terbanyak (389 dokumen), namun memiliki mean similarity terendah (0.9836) dan std deviation tertinggi (0.0129) di antara seluruh cluster. Kondisi ini mengindikasikan bahwa dokumen dalam cluster ini lebih bervariasi dan kurang homogen dibanding cluster lainnya. Cluster 2 berada pada posisi tengah dengan mean similarity 0.9856 dan std deviation

0.0093, menunjukkan tingkat keseragaman yang cukup baik namun tidak sehomogen Cluster 0.

Tabel 4. 14 Nilai Evaluasi pada MN-20

k	SSE	Silhouette	DBI
2	58,953.26	0.559	0.624
3	31,938.11	0.498	0.693
4	27,000.33	0.446	0.805
5	22,663.89	0.380	1.179
6	20,194.67	0.376	1.176
7	18,642.60	0.374	1.187
8	17,579.05	0.299	1.387
9	16,783.27	0.293	1.440

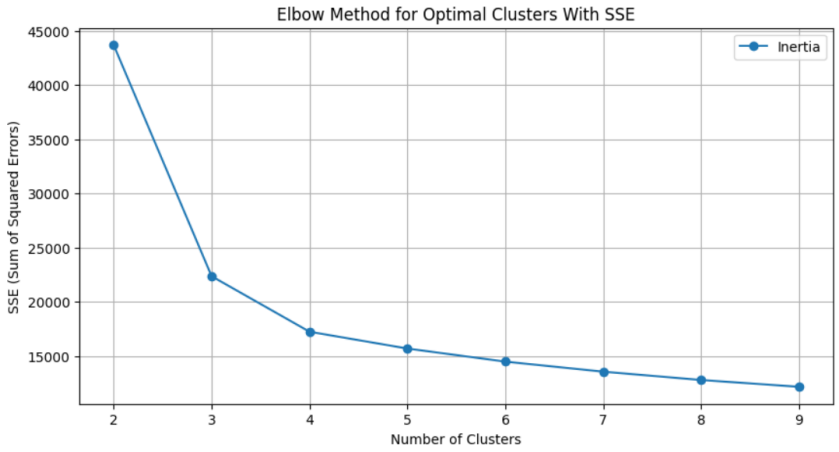
Dari tabel 4.14 diketahui bahwa nilai terbesar Silhouetee dan nilai terkecil DBI berada pada k=2. Namun pada k=3 menurut metode elbow. Nilai Silhouette tergolong tinggi, dan DBI juga rendah. Semua nilai berada pada posisi ke-2. Maka jika *k*=3, masih tergelong jumlah cluster yang bagus.

4.9 Skenario Pengujian 10 Kategori (*Balanced*)

Skenario pengujian kedua melibatkan 10 kategori masing-masing 100 data. Data di *setting* dengan perbandingan yang sama untuk mengetahui performa model pada dataset yang berbeda.

4.9.1 Ujicoba Data FP-10

Pada pengujian FP-10 ini, hasil yang didapatkan tidak jauh berbeda dari FP-20. Metode elbow menunjukkan jumlah k optimal terletak pada titik k=3.



Gambar 4. 11. Elbow Method FP-10

Gambar 4.11. menunjukkan bahwa titik siku paling terlihat pada k=3. Ketika k=4, garis mulai melandai.

Tabel 4. 15 Hasil Mean dan Std. FP-10

Cluster	**Jumlah Dokumen**	**Mean Similarity**	**Std. Deviation**
0	212	0.9919	0.0051
1	501	0.9863	0.0085
2	287	0.9949	0.0025

Jika dilihat pada tabel 4.15 jumlah masing-masing dokumen, nilai mean similarity dan juga standar deviasi tidak menunjukan perubahan yang signifikan dengan Hasil FP-20.

Evaluasi dilakukan pada masing-masing jumlah cluster untuk melihat perubahan nilai yang terjadi.

Tabel 4. 16 Hasil Evaluasi Model pada FP-10

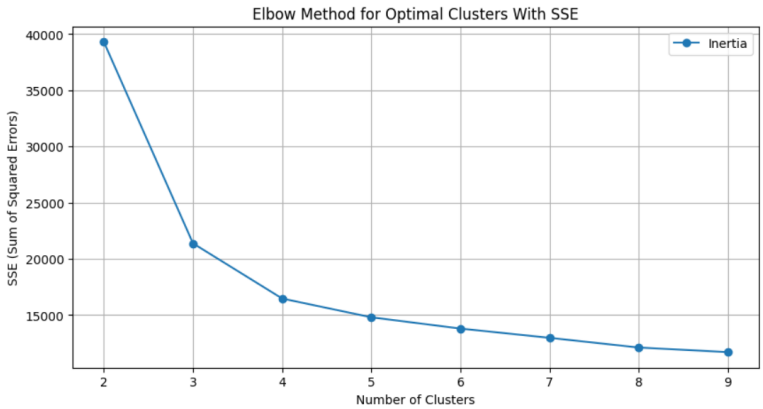
k	SSE	Silhouette Score	DBI
2	58,953.26	0.559	0.624
3	31,938.11	0.498	0.693
4	27,000.33	0.446	0.805
5	22,663.89	0.380	1.179

k	SSE	Silhouette Score	DBI
6	20,194.67	0.376	1.176
7	18,642.60	0.374	1.187
8	17,579.05	0.299	1.387
9	16,783.27	0.293	1.440

Tabel 4.16 menunjukkan nilai Silhouette tertinggi terjadi pada k = 2 (0.559), kemudian menurun menjadi 0.498 pada k = 3, dan terus menurun cukup tajam pada nilai k lebih besar. Penurunan ini menunjukkan bahwa pemisahan antar cluster semakin tidak jelas ketika jumlah cluster bertambah. Meskipun k=2 terbaik dalam hal silhouette, nilai k=3 masih cukup tinggi dan stabil dibanding k≥4. DBI terkecil berada pada k = 2 (0.624) yang menunjukkan cluster paling kompak dan terpisah. Namun nilai DBI masih cukup baik pada k = 3 (0.693) sebelum meningkat tajam pada k≥4. Semakin besar nilai k, cluster menjadi semakin tumpang tindih.

4.9.2 Ujicoba Data MP-10

Penentuan jumlah cluster optimal pada Skenario MP-10 dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 12 Elbow Method pada Skenario MP-10

Dari Gambar 4.12 Diketahui titik sikuk berada pada k=3. Untuk k=4 dan seterusnya garis elbow cenderung melandai. Sehingga sudah tidak optimal untuk dijadikan acuan.

Tabel 4. 17 Mean Sim. dan Std pada MP-10

Cluster	Jumlah Dokumen (*num_docs*)	*Mean Similarity*	*Std. Deviation*
Cluster 0	212	0.9901	0.0074
Cluster 1	501	0.9869	0.0091
Cluster 2	287	0.9944	0.0032

Tabel 4.17 menunjukkan nilai Cluster 2 memiliki mean similarity tertinggi (0.9944) dan standard deviation terendah (0.0032), yang menunjukkan bahwa dokumen pada cluster ini sangat homogen dan memiliki tingkat kemiripan konteks yang kuat. Cluster ini dapat dianggap sebagai kelompok dengan fokus topik paling jelas dan terdefinisi baik. Cluster 1 memiliki jumlah dokumen terbesar (501 dokumen) namun memiliki mean similarity terendah (0.9869) serta std deviation tertinggi (0.0091). Hal ini mengindikasikan bahwa cluster ini paling heterogen atau terdiri atas variasi konteks yang lebih luas dibandingkan cluster lainnya. Cluster 0 berada di posisi menengah baik dalam hal ukuran cluster

maupun homogenitas, dengan mean similarity 0.9901 dan std deviation 0.0074. Ini menunjukkan keseragaman yang cukup baik, namun tidak sekuat Cluster 2.

Selanjutnya mengevaluasi setiap cluster dengan Silhouette Score dan DBI.

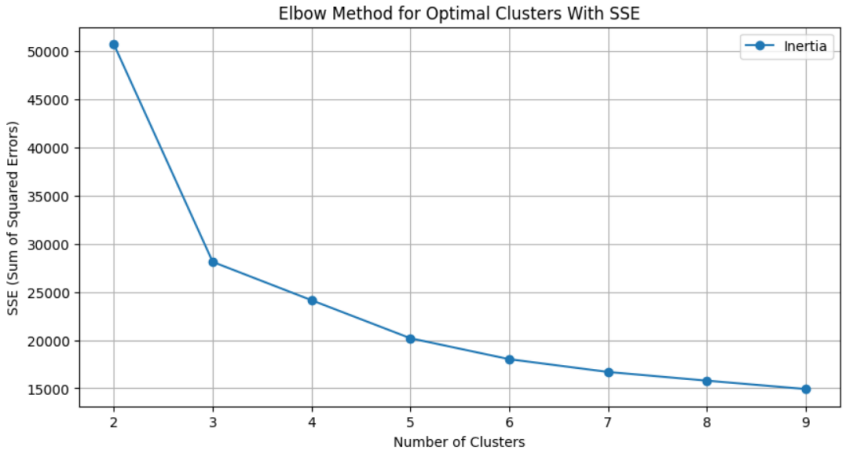
Tabel 4. 18 Nilai Evaluasi Model pada MP-10

k	SSE	Silhouette Score	DBI
2	39311.41	0.521	0.699
3	21384.43	0.491	0.707
4	16475.16	0.414	0.896
5	14805.15	0.375	1.076
6	13801.39	0.370	1.234
7	12977.55	0.346	1.405
8	12114.05	0.349	1.291
9	11706.76	0.336	1.361

Tabel 4.18 menunjukan nilai Silhouette tertinggi terdapat pada k=2 (0.521) yang menunjukkan pemisahan cluster yang paling baik. Namun, nilai ini terlalu tinggi karena pembagian hanya menjadi dua kelompok besar sehingga kurang representatif untuk variasi berita. Nilai DBI terbaik (terendah) juga pada k=2 (0.699), yang berarti jarak antar cluster paling terpisah ketika cluster berjumlah dua.

4.9.3 Ujicoba Data MN-10

Penentuan jumlah cluster optimal pada Skenario MN-10 dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 13 Titik Elbow pada MN-10

Dari gambar 4.13 secara singkat, diketahui bahwa titik elbow ditunjukkan pada k=3.

Tabel 4. 19 Mean Sim. dan Std pada MN-10

Cluster	Jumlah Dokumen	Mean Similarity	Std. Deviation
Cluster 0	336	0.9859	0.0093
Cluster 1	382	0.9851	0.0125
Cluster 2	282	0.9919	0.0045

Dari Tabel 4.19 diketahui bahwa Cluster 2 memiliki nilai mean similarity tertinggi (0.9919) dan standar deviasi terendah (0.0045). Hal ini menunjukkan bahwa dokumen dalam cluster 2 memiliki kemiripan konten yang sangat tinggi dan distribusi yang paling homogen. Dengan kata lain, cluster ini paling stabil dan representatif karena variasi antar dokumen sangat kecil. Cluster 0 memiliki mean similarity (0.9859) dan standar deviasi (0.0093) yang masih menunjukkan konsistensi internal yang baik. Cluster ini cukup homogen, meskipun tidak setinggi cluster 2. Cluster 1 memiliki mean similarity paling rendah (0.9851) dan standar deviasi tertinggi (0.0125) dibandingkan cluster lainnya.

Berikut adalah evaluasi Nilai Silhouette dan DBI untuk masing-masing jumlah cluster.

Tabel 4. 20 Nilai Evaluais per cluster pada MN-10

k	SSE	Silhouette	DBI
2	50,685.59	0.558	0.627
3	28,114.03	0.494	0.707
4	24,138.31	0.438	0.822
5	20,194.94	0.375	1.178
6	18,021.09	0.369	1.200
7	16,695.50	0.365	1.224
8	15,792.43	0.357	1.272
9	14,925.97	0.288	1.437

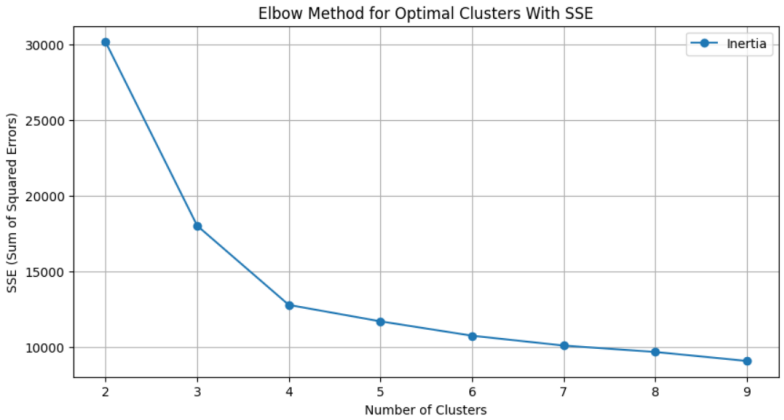
Dari tabel 4.20 diketahui bahwa nilai Silhouette paling tinggi terdapat pada k = 2 (0.558), yang menunjukkan tingkat pemisahan cluster yang paling baik. Namun penggunaan k=2 terlalu sederhana untuk dataset yang kompleks dan beragam. Setelah k=2, nilai Silhouette menurun, dan k = 3 (0.494) masih memberikan kualitas pemisahan yang cukup baik sebelum penurunan drastis terjadi pada k>4. Semakin kecil nilai DBI, semakin baik kualitas cluster. Nilai DBI terendah juga berada pada k = 2 (0.627), namun k = 3 (0.707) masih memberikan nilai yang cukup optimal sebelum DBI meningkat signifikan mulai dari k=4 ke atas.

4.10 Skenariio Pengujian 5 Kategori (*Unbalanced*)

Skenario pengujian ketiga menggunakan 5 kategori dengan jumlah data 300, 300, 200, 100, dan 100. Data di *setting* dengan perbandingan yang berbeda untuk mengetahui performa model pada dataset yang terjadi secara *real*.

4.10.1 Ujicoba Data FP-5

Penentuan jumlah cluster optimal pada Skenario FP-5 dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 14 Titik Elbow pada FP-5

Gambar 4.14 menunjukkan pada FP-5 Elbow mulai terlihat berbeda dari skenario sebelumnya. Jika dilihat titik siku berada pada k=4.

Tabel 4. 21 Mean Sim dan Std. pada FP-5

Cluster	Jumlah Dok.	Mean Similarity	Std. Deviation
Cluster 0	226	0.9957	0.00212
Cluster 1	326	0.9912	0.00434
Cluster 2	307	0.9956	0.00179
Cluster 3	141	0.9922	0.00544

Dari tabel 4.21 bisa diketahui bahwa Cluster 0 dan Cluster 2 memiliki nilai mean similarity tertinggi (0.9957 dan 0.9956) serta nilai Std. Deviation terendah (0.00212 dan 0.00179). Hal ini menunjukkan bahwa dokumen-dokumen Cluster 0 dan Cluster 2sangat homogen dan memiliki konteks bahasan yang sangat fokus serta konsisten. Cluster 1 memiliki jumlah dokumen paling banyak (326) dengan mean similarity sedikit lebih rendah (0.9912) dan standar deviasi lebih tinggi

(0.00434). Kondisi ini menunjukkan bahwa cluster tersebut lebih beragam dalam hal konteks karena menghimpun dokumen dengan variasi tema yang lebih luas. Cluster 3 memiliki jumlah dokumen paling sedikit (141) dan standar deviasi tertinggi (0.00544). Ini memperlihatkan bahwa meskipun tingkat kesamaan konteks masih tinggi, variasi internal cluster ini lebih besar dibanding cluster lainnya, yang berarti terdapat beberapa sub-topik yang berbeda namun masih relevan dalam satu kelompok besa.

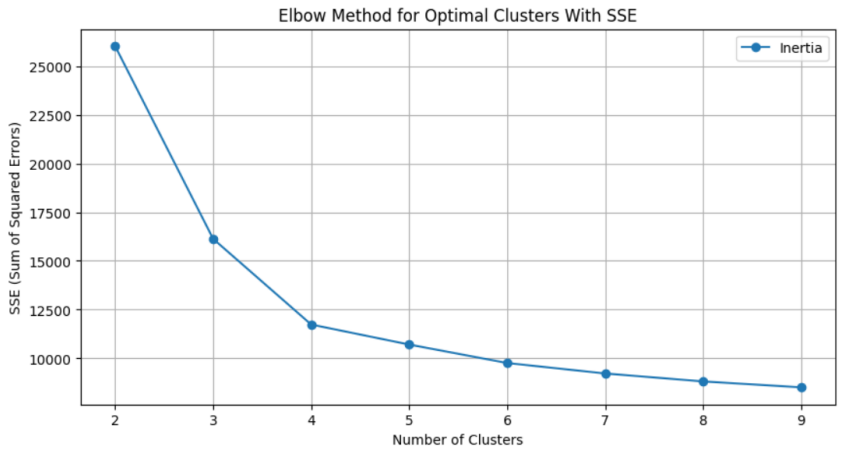
Tabel 4. 22 Nilai Evaluasi pada FP-5

k	SSE	Silhouette Coefficient	DBI
2	30154.10	0.539	0.717
3	18009.75	0.481	0.683
4	12774.61	0.401	0.908
5	11689.30	0.370	1.184
6	10739.48	0.340	1.362
7	10088.33	0.339	1.354
8	9663.91	0.266	1.541
9	9071.32	0.266	1.492

Tabel 4.22 menunjukkan Silhouette tertinggi berada pada k=2 (0.539), namun pembentukan hanya dua cluster cenderung terlalu menyederhanakan variasi konteks pada dataset. Nilai DBI terbaik (terendah) terdapat pada k=3 yaitu 0.683, menunjukkan pemisahan cluster yang paling baik dan struktur internal cluster yang paling kompak. Berdasarkan kombinasi indikator SSE, Silhouette Coefficient, dan DBI, nilai k=3 merupakan jumlah cluster optimal untuk dataset ini.

4.10.2 Ujicoba Data MP-5

Penentuan jumlah cluster optimal pada Skenario MP-5 dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 15 Titik Elbow pada MP-5

Diketahui dari gambar 4.15 bahwa titik Elbow juga berada pada k=4. Namun k=3 juga tergolong bagus. Diatas k=4 *line* mulai melandai.

Tabel 4. 23 Mean Sim. dan Std. pada MP-5

Cluster	Jumlah Dokumen	Mean Similarity	Std. Deviation
Cluster 0	226	0.9957	0.00245
Cluster 1	326	0.9921	0.00372
Cluster 2	307	0.9963	0.00108
Cluster 3	141	0.9903	0.00783

Diketahui dari tabel 4.23 Cluster 2 memiliki nilai mean similarity tertinggi (0.9963) dan Std. Deviation terendah (0.00108). Ini menunjukkan bahwa dokumen dalam cluster tersebut sangat seragam dan memiliki topik yang sangat fokus serta spesifik secara kontekstual. Cluster 0 juga memiliki tingkat kemiripan konteks yang sangat tinggi (mean 0.9957) dan variasi kecil (std 0.00245),

sehingga dapat dikategorikan sebagai cluster yang stabil. Cluster 1 memiliki jumlah dokumen terbesar (326) dengan mean similarity lebih rendah (0.9921) dan standar deviasi lebih tinggi (0.00372). Hal ini mengindikasikan bahwa cluster ini menampung variasi konteks yang lebih luas dibanding cluster 0 dan 2. Cluster 3 memiliki jumlah dokumen paling sedikit (141) dan standar deviasi tertinggi (0.00783). Ini menunjukkan bahwa cluster ini merupakan kelompok yang paling heterogen, berpotensi berisi kombinasi beberapa sub-topik yang masih berada dalam satu konteks besar sehingga pembentukan sub-cluster tambahan mungkin layak dipertimbangkan.

Evaluasi Nilai Silhouette dan DBI pada semuua cluster dijabarkan berikut ini.

Tabel 4. 24 Evaluasi pada MP-5

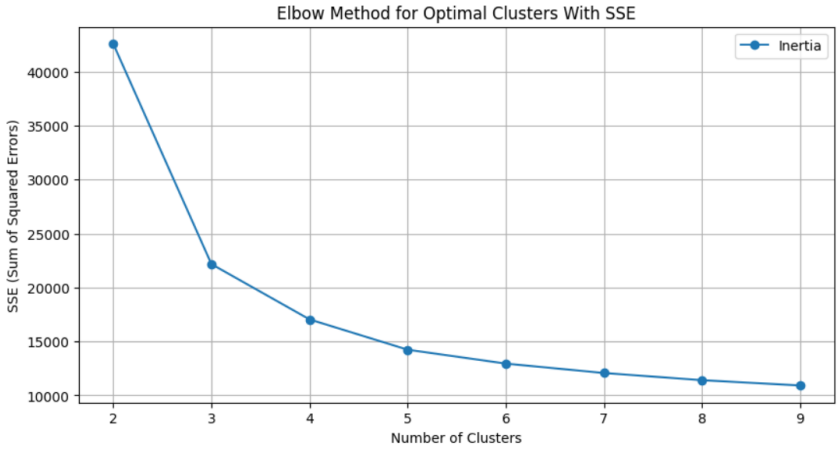
k	SSE	Silhouette	DBI
2	26017.54	0.538	0.723
3	16130.75	0.476	0.714
4	11735.37	0.391	0.922
5	10707.13	0.368	1.122
6	9759.66	0.345	1.288
7	9222.15	0.342	1.388
8	8814.01	0.282	1.496
9	8510.79	0.271	1.532

Tabel 4.24 menunjukan bahwa nilai Silhouette tertinggi berada pada k=2 (0.538), menunjukkan pemisahan cluster terbaik jika hanya mempertimbangkan kedekatan dan keterpisahan. Nilai DBI terbaik (paling kecil) berada pada k=3 (0.714), menunjukkan kualitas separasi cluster paling baik pada titik tersebut. Berdasarkan analisis gabungan metrik SSE, Silhouette Coefficient, dan DBI, nilai

k optimal adalah k = 3, karena memberikan keseimbangan terbaik antara kompaksi cluster dan keterpisahan antar cluster.

4.10.3 Ujicoba Data MN-5

Penentuan jumlah *cluster* optimal pada Skenario MN-5 dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (*Sum of Squared Errors*).



Gambar 4. 16 Titik elbow pada MN-5

Gambar 4.16 menunjukan bahwa titik elbow berbeda dengan 2 skenario sebelumnya, yakni berada pada k=3.

Tabel 4. 25 Mean Sim. dan Std. pada MN-5

Cluster	Jumlah Dok.	Mean Similarity	Std. Deviation
Cluster 0	251	0.9937	0.00360
Cluster 1	231	0.9870	0.00853
Cluster 2	518	0.9877	0.00938

Tabel 4.25 menunjukan bahwa Cluster 0 memiliki mean similarity tertinggi (0.9937) dengan standar deviasi terendah (0.00360). Hal ini menunjukkan bahwa

cluster ini merupakan kelompok dokumen yang paling homogen dan sangat konsisten dalam kesamaan konteks. Cluster 1 memiliki mean similarity terendah (0.9870) dan std tertinggi (0.00853), menunjukkan bahwa dokumen dalam cluster ini lebih bervariasi sehingga cluster ini lebih heterogen dibanding cluster lainnya. Hal ini mengindikasikan bahwa cluster ini kemungkinan mencakup beberapa sub-topik dengan variasi konteks yang lebih luas. Cluster 2 memiliki jumlah dokumen terbesar (518 dokumen) dengan mean similarity 0.9877. Nilai std yang lebih tinggi (0.00938) menunjukkan bahwa semakin banyak dokumen yang tergabung, semakin besar pula variasi konteks di dalam cluster tersebut.

Tabel 4. 26 Nilai Evaluasi pada MN-5

k	SSE	Silhouette	DBI
2	42,591.31	0.584	0.583
3	22,176.36	0.504	0.691
4	17,049.08	0.447	0.798
5	14,238.16	0.412	1.086
6	12,951.63	0.405	1.158
7	12,078.58	0.403	1.167
8	11,413.66	0.336	1.369
9	10,921.81	0.329	1.408

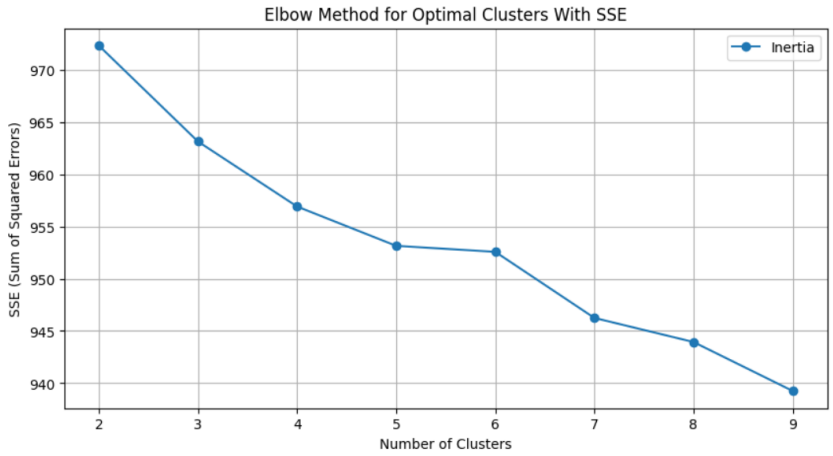
Tabel 4.26 menunjukkan bahwa nilai Silhouette tertinggi pada k=2 (0.584), menunjukkan pemisahan cluster paling jelas jika hanya mempertimbangkan dua cluster. Namun, meski lebih rendah pada k=3 (0.504), nilai ini masih cukup tinggi, menunjukkan cluster masih terpisah dengan baik. Nilai DBI terendah pada k=2 (0.583), namun k=3 (0.691) masih tergolong baik. Berdasarkan kombinasi metrik SSE, Silhouette, dan DBI, jumlah cluster optimal adalah k = 3

4.11 Analisis Pengujian Metode Baseline TF-IDF

Metode baseline bukan termasuk pada skenario ujicoba, namun hanya sebagai pembanding hasil *clustering* dengan metode yang diusulkan.

4.11.1 Data Skenario 1 (TF-IDF-S1)

Penentuan jumlah cluster optimal pada Skenario 1 TF-IDF dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 17 Titik elbow dengan TF-IDF Skenario 1

Dilihat dari Gambar 4.14 menunjukkan bahwa titik elbow berada pada k=5, kemudian titik melandai.

Tabel 4. 27 Mean Sim. dan Std. pada TF-IDF-S1

Cluster	Jumlah Dok.	Mean Similarity	Std. Deviation
Cluster 0	85	0.0594	0.0841
Cluster 1	81	0.0963	0.0818
Cluster 2	664	0.0075	0.0202
Cluster 3	138	0.1328	0.0644
Cluster 4	32	0.1940	0.1078

Dari tabel 4.27 diketahui bahwa semua cluster memiliki mean similarity sangat rendah yakni kurang dari 0.2, menandakan dokumen dalam cluster memiliki kemiripan kontekstual yang minim. Hal ini bisa terjadi jika dataset bersifat sangat heterogen atau representasi vektor dokumen menangkap konteks yang sangat berbeda antar dokumen.

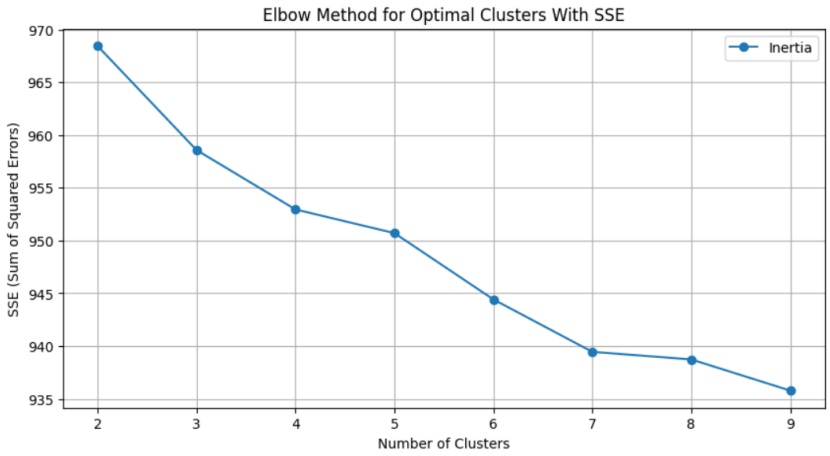
Tabel 4. 28 Evaluasi model TF-IDF-S1

k	SSE	Silhouette	DBI
2	972.30	0.009	6.709
3	963.13	0.009	6.317
4	956.91	0.011	5.209
5	953.15	0.011	6.100
6	952.57	0.011	8.710
7	946.25	0.013	8.156
8	943.94	0.013	7.510
9	939.27	0.010	7.532

Dari Tabel 4.28 terlihat bahwwa nilai Silhouette sangat rendah yakni kurang dari 0.02 untuk semua jumlah k, menunjukkan bahwa dokumen dalam cluster hampir tidak memiliki pemisahan yang jelas. Nilai DBI lebih dari 5 pada semua k, menunjukkan cluster kurang kompak dan tidak terpisah dengan baik.

4.11.2 Data Skenario 2 (TF-IDF-S2)

Penentuan jumlah cluster optimal pada Skenario 2 TF-IDF dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 18 Titik elbow pada TF-IDF-S2

Diketahui dari Gambatr 4.18 bahwa titik elbow berada pada k=4. Pada titik k selanjutnya menunjukan ketidak konsistenan bentuk elbow dan cenderung landai.

Tabel 4. 29 Mean Sim dan Std. pada TF-IDF-S2

Cluster	Jumlah Dok.	Mean Similarity	Std. Deviation
Cluster 0	91	0.0855	0.0923
Cluster 1	76	0.1186	0.0868
Cluster 2	664	0.0086	0.0234
Cluster 3	169	0.1236	0.0598

Diketahui secara singkat pada tabel 4.29 bahwa semua cluster memiliki mean similarity rendah yakni kurang dari 0.13, menunjukkan dokumen-dokumen dalam masing-masing cluster memiliki tingkat kemiripan yang sangat rendah.

Tabel 4. 30 Evaluasi pada TF-IDF-S2

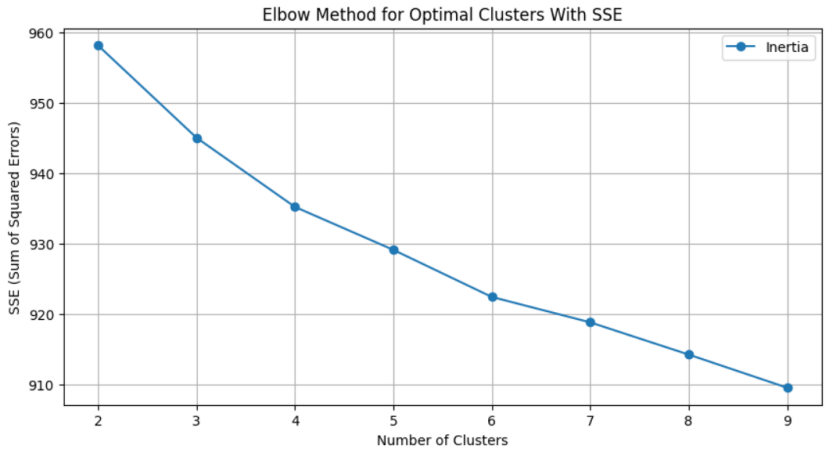
k	SSE	Silhouette	DBI
2	968.45	0.009	5.696
3	958.58	0.011	6.564
4	952.95	0.012	5.928
5	950.71	0.012	9.117
6	944.42	0.015	6.393

k	SSE	Silhouette	DBI
7	939.45	0.016	6.001
8	938.73	0.012	7.332
9	935.78	0.010	6.661

Dari tabel 4.30 dijelaskna bahwa nilai sangat rendah yakni kurang dari 0.02 untuk semua k, menunjukkan cluster sangat tumpang tindih dan dokumen dalam cluster hampir tidak terpisah dengan jelas. Nilai DBI relatif tinggi yakni lebih dari 5, menunjukkan bahwa cluster kurang kompak dan tidak terpisah dengan baik. DBI terbaik (terendah) terdapat pada k=2 dengan nilai 5.696, namun kualitas cluster tetap rendah.

4.11.3 Data Skenario 3 (TF-IDF-S3)

Penentuan jumlah cluster optimal pada Skenario 3 TF-IDF dilakukan menggunakan metode Elbow dengan memanfaatkan nilai SSE (Sum of Squared Errors).



Gambar 4. 19 Titik elbow pada TF-IDF-S3

Gambar 4.19 menunjukkan bahwa titik elbow kurang begitu terlihat. Namun yang paling dominan letak titik sikunya adalah pada k=4

Tabel 4. 31 Mean Sim. dan Std. pada TF-IDF-S3

Cluster	Jumlah Dok.	Mean Similarity	Std. Deviation
Cluster 0	462	0.0102	0.0297
Cluster 1	283	0.1011	0.0556
Cluster 2	72	0.2114	0.0888
Cluster 3	183	0.0691	0.0686

Dilihat dari tabel 4.31 diketahui bahwa Cluster 2 memiliki mean similarity tertinggi 0.2114, menunjukkan bahwa dokumen dalam cluster ini memiliki tingkat kesamaan konteks paling tinggi dibanding cluster lain, meskipun masih rendah. Cluster 0 memiliki *mean similarity* paling rendah 0.0102 dengan standar deviasi kecil 0.0297, menandakan cluster ini berisi dokumen yang hampir tidak memiliki kemiripan satu sama lain

Tabel 4. 32 Evaluasi pada TF-IDF-S3

k	SSE	Silhouette	DBI
2	958.10	0.013	6.369
3	945.01	0.016	5.621
4	935.20	0.020	6.067
5	929.10	0.022	6.099
6	922.43	0.024	5.644
7	918.82	0.017	5.988
8	914.25	0.025	5.387
9	909.56	0.018	5.798

Dari Tabel 4.32 diketahui bahwa nilai Silhouette pada semua cluster sangat rendah yakni kurang dari 0.03, menandakan bahwa dokumen dalam cluster hampir tidak terpisah dan cluster sangat tumpang tindih. Nilai DBI relatif tinggi uakni lebih dari 5 untuk semua k, menunjukkan cluster kurang kompak dan *overlap* antar cluster tinggi.

4.12 Perbandingan Hasil Ujicoba dengan Baseline TF-IDF

Perbandingan dilakukan untuk melihat performa pengelompokkan pada masing-masing model. Perbandingan akan dilakukan sesuai pembagian data skenario. Perbandingan hasil ujicoba yang pertama dilakukan pada Nilai *Mean Similarity* dan Standar Deviasi, untuk melihat pola kemiripan pada masing-masing model. Dan yang kedua, pada nilai evaluasi dengan tujuan untuk melihat performa model yang diusulkan. Jumlah cluster yang dipakai adalah *k*=3, karena rata-rata hasil ujicoba metode elbow menunjukkan *k*=3.

4.12.1 Perbandingan Hasil Ujicoba pada Skenario 1

Perbandingan pertama dilihat dari nilai *mean similairity* dan *Standar Deviasi per-cluster*. Disini juga terlihat jumlah dari dokumen dari setiap *cluster* yang terbentuk.

Tabel 4. 33 Perbandingan Hasil ujicoba pada Mean Sim dan Std.

	Number of Doc.				**Mean Sim.**				**Std**			
Cluster	**FP-20**	**MP-20**	**MN-20**	**TF-IDF-S1**	**FP-20**	**MP-20**	**MN-20**	**TF-IDF-S1**	**FP-20**	**MP-20**	**MN-20**	**TF-IDF-S1**
0	203	205	282	734	0.9926	0.9899	0.9921	0.00793	0.0051	0.0083	0.0044	0.02125
1	507	288	389	137	0.9873	0.9942	0.9836	0.13340	0.0085	0.0033	0.0129	0.06468
2	290	507	329	129	0.9950	0.9871	0.9856	0.07748	0.0025	0.0090	0.0093	0.08380

Dari Tabel 4.33 diketahui bahwa pada Cluster 0, baseline memiliki jumlah dokumen sangat besar yakni 734. Sedangkan model yang diuji mendapatkan

anggota cluster yang cenderung seragam yakni FP 203, MP 205, dan MN 282 yang sedikit lebih besar. Mean similarity pada model yang diuji cenderung di atas 0,9. Dan untuk baseline paling besar 0.1334. Standar deviasi TF-IDF-S1 memiliki nilai terendah di angka 0.02. Lebih tinggi dari pada model yang diuji.

Perbandingan kedua dilihat dari nilai Silhouette dan DBI per cluster.

Tabel 4. 34 Perbandingan Nilai Evaluasi pada Skenario 1

k	Silhouette Score				DBI			
	FP-20	MP-20	MN-20	TF-IDF-S1	FP-20	MP-20	MN-20	TF-IDF-S1
2	0.507	0.508	0.559	0.009	0.721	0.721	0.624	6.709
3	0.505	0.497	0.498	0.009	0.667	0.688	0.693	6.317
4	0.430	0.423	0.446	0.011	0.896	0.884	0.805	5.209
5	0.373	0.370	0.380	0.011	1.231	1.177	1.179	6.100
6	0.352	0.371	0.376	0.011	1.430	1.255	1.176	8.710
7	0.347	0.348	0.374	0.013	1.431	1.415	1.187	8.156
8	0.249	0.269	0.299	0.013	1.663	1.528	1.387	7.510
9	0.248	0.247	0.293	0.010	1.586	1.572	1.440	7.532

Tabel 4.34 menunjukan perbandingan hasil uji performa pada masing-masing cluster. Pada jumlah cluster optimal k=3 Nilai silhouette konsisten berada di atas baseline TF-IDF.

4.12.2 Perbandingan Hasil Ujicoba pada Skenario 2

Hasil perbandingan Ujicoba Skenario 2 ditunjukkan dengan 2 model perbandingan pada tabel 4.33 dan 4.34.

Tabel 4. 35 Perbandingan nilai Mean Sim dan Std pada Skenario 2

Cluster	**Number of Doc.**				**Mean Sim.**				**Std**			
	FP-10	**MP-10**	**MN-10**	**TF-IDF-S2**	**FP-10**	**MP-10**	**MN-10**	**TF-IDF-S2**	**FP-10**	**MP-10**	**MN-10**	**TF-IDF-S2**

				S2								
0	212	212	336	667	0.9919	0.9901	0.9859	0.0087	0.0051	0.0074	0.0093	0.0242
1	501	501	382	169	0.9863	0.9869	0.9851	0.1236	0.0085	0.0091	0.0125	0.0598
2	287	287	282	164	0.9949	0.9944	0.9919	0.0727	0.0025	0.0032	0.0045	0.0721

Tabel 4.35 menunjukan perbedaan hasil mean dan Std pada skenario 2. TF-IDF-S2 membentuk cluster lebih besar dan heterogen pada cluster 0, dan cluster lebih kecil tapi cukup beragam pada cluster 1 & 2. Dokumen dalam cluster tidak terlalu mirip secara kontekstual, terlihat dari mean similarity rendah dan std tinggi. FP-10, MP-10, MN-10, Cluster lebih kecil tapi sangat homogen. Menunjukkan metode ini lebih fokus mengelompokkan dokumen yang sangat mirip, sehingga cluster lebih stabil dan rapat.

Tabel 4. 36. Perbandingan Nilai Silhoutte dan DBI

k	Silhouette Score				DBI			
	FP-10	MP-10	MN-10	TF-IDF-S2	FP-10	MP-10	MN-10	TF-IDF-S2
2	0.513	0.521	0.558	0.009	0.712	0.699	0.627	5.696
3	0.492	0.491	0.494	0.011	0.686	0.707	0.707	6.564
4	0.417	0.414	0.438	0.012	0.927	0.896	0.822	5.928
5	0.363	0.375	0.375	0.012	1.282	1.076	1.178	9.117
6	0.351	0.370	0.369	0.015	1.357	1.234	1.200	6.393
7	0.347	0.346	0.365	0.016	1.392	1.405	1.224	6.001
8	0.353	0.349	0.357	0.012	1.292	1.291	1.272	7.332
9	0.258	0.336	0.288	0.010	1.516	1.361	1.437	6.661

Tabel 4.36 juga menunjukan dari semua nilai evaluasi Silhouette dan DBI mengungguli nilai evaluasi semua cluster pada *baseline*

4.12.3 Perbandingan Hasil Ujicoba pada Skenario 3

Perbandingan hasil ujicoba pada Skenario 3 juga dibandingkan dengan model *baseline* TF-IDF. Perbandingan ini untuk melihat secara langsung perbedaan kualitas kesamaan teks dalam satu *cluster.*

Tabel 4. 37 Perbandingan Mean similarity dan Std Skenario 3 terhadap *baseline*

Cluster	Number of Doc.				Mean Sim.				Std			
	FP-5	MP-5	MN-5	TF-IDF-S3	FP-5	MP-5	MN-5	TF-IDF-S3	FP-5	MP-5	MN-5	TF-IDF-S3
0	141	627	251	283	0.9922	0.9902	0.9937	0.1011	0.0054	0.0062	0.0036	0.0556
1	629	141	231	71	0.9890	0.9903	0.9870	0.2144	0.0068	0.0078	0.0085	0.0880
2	230	232	518	646	0.9956	0.9955	0.9877	0.0131	0.0022	0.0026	0.0094	0.0343

Tabel 4.37 menunjukkan perbedaan karakteristik, yakni TF-IDF-S3 membentuk cluster sedang dengan dokumen cukup heterogen, dokumen tidak terlalu mirip. Sedangakan FP/MP/MN membentuk cluster lebih kecil tapi sangat homogen, dokumen sangat mirip.

Tabel 4. 38 Perbandingan Nilai Evaluasi Skenario 3 terhadap baseline TF-IDF-S3.

k	Silhouette Score				DBI			
	FP-5	MP-5	MN-5	TF-IDF-S3	FP-5	MP-5	MN-5	TF-IDF-S3
2	0.539	0.538	0.584	0.013	0.717	0.723	0.583	6.369
3	0.481	0.476	0.504	0.016	0.683	0.714	0.691	5.621
4	0.401	0.391	0.447	0.020	0.908	0.922	0.798	6.067
5	0.370	0.368	0.412	0.022	1.184	1.122	1.086	6.099
6	0.340	0.345	0.405	0.024	1.362	1.288	1.158	5.644
7	0.339	0.342	0.403	0.017	1.354	1.388	1.167	5.988
8	0.266	0.282	0.336	0.025	1.541	1.496	1.369	5.387
9	0.266	0.271	0.329	0.018	1.492	1.532	1.408	5.798

Tabel 4.38 menunjukan perbandingan dari semua cluster. Jika dilihat dari cluster optimal k=2, nilai silhouette rata-rata > 0.5. Sedangan baseline nilainya di bawa 0.03.

4.13 Pembahasan

Dari Perbandingan Hasil ujicoba Skenario 1 Tabel 4.31, Nilai *mean similarity* dari setiap cluster yang terbentuk menunjukkan kemiripan yang signifikan. Nilai berada diatas 0,9. Menunjukkan setiap cluster memiliki nilai kemiripan kontekstual yang sangat dekat. TF-IDF sebagai baseline, mendapatkan nilai tertinggi 0.13. menunjukkan dokumen dalam satu cluster lebih beragam. Diketahui bahwa TF-IDF-S1 menghasilkan cluster lebih kecil tapi lebih bervariasi secara kemiripan internal dibanding tiga metode lainnya. FP-20, MP-20, MN-20 cenderung menghasilkan cluster lebih besar dengan dokumen sangat mirip, sehingga cluster mereka mungkin terlalu homogen. perbedaan mean similarity dan std menunjukkan TF-IDF-S1 lebih sensitif terhadap variasi konten dokumen, sementara metode berbasis FP/MP/MN lebih agresif dalam mengelompokkan dokumen yang hampir mirip menjadi satu cluster. Secara aplikasi, TF-IDF-S1 mungkin lebih baik untuk menilai keragaman topik dalam cluster, sementara FP/MP/MN cocok untuk mengelompokkan dokumen yang sangat mirip.

Nilai performa pada pengujian 1 juga menunjukkan konsistensi nilai Silhouette untuk semua cluster model yang diuji. Pada cluster optimal terlihat Nilai Silhouette berada pada angka 0.505 dengan DBI 0.693. Sedangkan Baseline memliki angak SC 0.009 dan DBI 6.317 menunjukkan bahwa model yang di uji berada pada level yang bagus.

Pada perbandingan skenario 2 dan 3 juga demikian, tidak terlalu menunjukkan perubahan hasil perbandingan yang signifikan. Secara umum, metode FP, MP, dan MN secara konsisten menghasilkan cluster yang relatif homogen dengan nilai mean similarity yang tinggi >0.9 dan standar deviasi rendah, sementara metode TF-IDF pada semua skenario cenderung membentuk cluster yang lebih heterogen, ditandai dengan mean similarity yang rendah dan standar deviasi lebih tinggi.

Pada perbandingan performa antar model yang diusulkan, dengan jumlah cluster optimal pada k=3 dapat diketahuai bahwa, pada Skenario 1 model dengan performa tertinggi adalah FP-20 dengan Nilai Silhouette tertinggi 0,505 dengan nilai DBI paling kecil 0,667. Pada Skenario 2, Nilai Silhouette tertinggi terdapat pada model MN-10, namun perbedaan ini tidak terlalu signifikan yakni di angka 0,494 dan FP-10 0,492 dengan DBI terendah 0,686. Pada Skenario 3, nilai Sillhouette terbesar terletak pada MN-5 dengan angka silhouette 0,504. Namun demikian, nilai DBI terendah terdapat pada FP-5 dengan nilai 0, 683. Dengan demikian model First POS masih bisa dikatakan mengungguli pada semua skenario dengan DBI terendah. Pada pengukuran Silhoutte memang tidak mendapatkan nilai performa tertinggi, namun nilainya tidak terlampau jauh.

Dengan demikian, meskipun terdapat variasi jumlah dokumen pada masing-masing cluster, pola distribusi dan karakteristik kualitas cluster tidak jauh berbeda antara skenario pengujian, sehingga menunjukkan bahwa perubahan parameter yang diterapkan tidak memberikan dampak signifikan terhadap struktur cluster yang terbentuk.

Namun demikian, Dari skenario 1 dari 20 kategori hanya mendapatkan jumlah cluster optimal k=3. Ini mengindikasikan bahwa terjadi tumpeng tindih data berdasarkan kategori. Jika dilihat dari data teks hasil clustering optimal, ditemukan beberapa data yang membahas topik yang mirip berada pada kategori berbeda.

Tabel 4. 39. Contoh data dalam 1 cluster

corpus	category	cluster
What’s Israel’s Hannibal Directive? A former IDF soldier tells all. The controversial policy to avoid capture of Israeli soldiers isn't formally in place now. But echoes persist in Gaza.	Architecture	0
Israeli-Palestinian Peace Camp Shaken But Determined. The Israel-Palestinian peace camp has long promoted dialogue against hatred and bloodshed but the passions inflamed by the deadliest Gaza war yet pose entirely new challenges for the movement.	Climate	0
Pro-Palestinian Israelis face threats, but vow to keep fighting for peace. Activists in Israel say as right-wing, pro-war voices dominate the discourse, their work has never felt more important.	Climate	0
Resistance calibrating pressure on Israel and backers amid Gaza war: Iran FM. Iranian Foreign Minister Hossein Amir-Abdollahian has told members of the Iranian Parliament the regional resistance forces are carefully calibrating their response to the crimes of Israel in Gaza as he warns that continued Israeli aggression on the besieged …	Fashion	0
Iran warns US, Israel of 'harsh consequences' if war crimes continue in Gaza. Iran's foreign minister has warned the United States and Israel that they will face "harsh consequences" if they fail to permanently stop war crimes in the Gaza Strip committed during the genocidal war on the besieged Palestinian territory.	Food	0

corpus	category	cluster
EU to boost Gaza aid amid Israel-Hamas truce. The bloc welcomed the opportunity of the four-day window; also says development aid is not being stolen by Hamas.	Food	0
Children screamed in street as we fled 2am Gaza air strike. Power runs out in Gaza after another night of massive destruction by Israeli forces.	Food	0
Israel Announces Tentative Deal With Hamas to Free Some Hostages, Pause Fighting. A tentative deal has been reached between Israel and Hamas to release at least 50 women and children hostages currently being held by Hamas in exchange for a brief pause in fighting and the release of an unspecified number of Palestinian prisoners.	Health	0
The Health Ministry in Hamas-ruled Gaza Strip says the Palestinian death toll in Israel-Hamas war has passed 11,000. The Health Ministry in Hamas-ruled Gaza Strip says the Palestinian death toll in Israel-Hamas war has passed 11,000	Health	0
Moscow gives update on evacuation of Russians from Gaza. More than half of the Russian citizens in Gaza have been evacuated as fighting between Israel and Hamas continues, Moscow has said Read Full Article at RT.com	Health	0
Israel takes responsibility for Gaza ambulance attack. Israel has admitted to attacking an ambulance outside Gaza's Al-Shifa Hospital, resulting in casualties. The incident led to the deaths of at least 15 people and injured 50 others, according to Hamas-run health authorities. Israel claimed that the ambulance w…	Health	0
Children in Gaza being denied right to life and health: UNICEF. The United Nations International Children's Emergency Fund (UNICEF) says children in Gaza are facing a dire humanitarian situation amid an ongoing Israeli aggression which has paralyzed medical and healthcare services in the Palestinian territory.	Health	0

corpus	category	cluster
A disabled Israeli teenager with muscular dystrophy is a hostage held by Hamas in Gaza after she was abducted from the Supernova music festival. Israeli Rut Perez, a 17-year-old wheelchair user, is being held hostage by Hamas militants in Gaza, after being abducted at the Supernova festival.	Music	0

Jika dilihat sekilas berita pada tabel 4.39. menunjukkan bahwa banyak dokumen dalam dataset memiliki konteks yang sama (isu Gaza–Israel), menyebabkan berbagai kategori masuk ke cluster yang sama karena pola bahasa dan terminologi sangat serupa. Oleh karena itu, meskipun kategori editorialnya berbeda, secara kontekstual dokumen-dokumen tersebut berada dalam ruang vektor yang berdekatan dan mengarah pada pembentukan cluster yang sama.

Hasil pengelompokan menunjukkan adanya hubungan atau kedekatan makna antar berita meskipun berasal dari kategori yang berbeda. Penelitian ini tidak bertujuan untuk menggantikan sistem kategorisasi berita yang telah ada, melainkan untuk mengelompokkan berita berdasarkan kedekatan makna atau konteks, sehingga hubungan antar isu yang saling berkaitan dapat diidentifikasi sebagai dasar analisis lebih lanjut. Evaluasi menggunakan Silhouette Coefficient dan Davies-Bouldin Index menunjukkan bahwa metode atau model yang diusulkan mampu menghasilkan kualitas klaster yang lebih baik dibandingkan pendekatan baseline.

4.14 Pengelompokan Berita menurut Pandangan Islam

Proses pengelompokan informasi merupakan kegiatan yang sangat dianjurkan dalam Islam. Tujuannya yakni untuk memisahkan yang benar dan

salah, memahami berita dan mengelola pengetahuan, dan menghasilkan pemahaman yang lebih baik. Dengan mengelompokan berita, kita dapat memahami berita yang saling berhubungan meskipun diletakan pada kategori yang berbeda. Pengelompokan secara kontekstual juga membantu kita memahami dan menganilisis keputusan yang tepat untuk menjustifikasi berita yang terjadi.

Didalam latar belakang penelitian ini, disebutkan Surat Hujurat ayat 13:

يٰٓاَيُّهَا النَّاسُ اِنَّا خَلَقْنٰكُمْ مِّنْ ذَكَرٍ وَّاُنْثٰى وَجَعَلْنٰكُمْ شُعُوْبًا وَّقَبَاۤىِٕلَ لِتَعَارَفُوْاۚ اِنَّ اَكْرَمَكُمْ عِنْدَ اللّٰهِ اَتْقٰىكُمْۗ اِنَّ اللّٰهَ عَلِيْمٌ خَبِيْرٌ ١٣

Wahai manusia! Sungguh, Kami telah menciptakan kamu dari seorang laki-laki dan seorang perempuan, kemudian Kami jadikan kamu berbangsa-bangsa dan bersuku-suku agar kamu saling mengenal. Sesungguhnya yang paling mulia di antara kamu di sisi Allah ialah orang yang paling bertakwa. Sungguh, Allah Maha Mengetahui, Mahateliti. (QS. [49] Al-Hujurat: 13)

Di dalam tafsir tahlili yang diakses pada laman *quran.nu.or.id* tentang Surat ini dijelaskan bahwa Allah menciptakan manusia dari golongan laki-laki dan Perempuan dan menjadikannya berbangsa -bangsa dan bersuku-suku, berbeda-beda warna kulit bukan untuk saling mencemoohkan, tetapi untuk saling mengenal dan menolong.

Dalam Tafsir Al-Munir karya Syaikh Wahbah Az-Zuhaili (Az-Zuhaili, 2013) menerangkan bahwa ayat ini menjelaskan 3 hal yakni, persamaan, saling mengenal antar komunitas Masyarakat, dan tolok ukur kemuliaan seseorang berdasarkan ketakwaan dan amal saleh. Demikian juga pengelompokan, harus ada persamaan untuk mengelompokan berita, dalam hal ini mengelompokkan berdasarkan kemiripan kontekstual. Kemudian saling mengenal, pengelompokan

juga memberikan wawasan kepada kita, untuk mengenali dan memahami berita yang sedang terjadi, keterkaitan dengan berita lain dalam satu kelompok. Dan yang terakhir, yakni tolok ukur kemuliaan. Pada pengelompokan berita juga ada tolok ukur untuk mengelompokan berita kepada cluster yang sama.

Di dalam Al-Quran Surat Al-Waqiah ayat 7-10 juga disebutkan:

وَّكُنْتُمْ اَزْوَاجًا ثَلٰثَةً ۗ ﴿٧﴾ فَاَصْحٰبُ الْمَيْمَنَةِ ۙ مَآ اَصْحٰبُ الْمَيْمَنَةِ ۗ ﴿٨﴾ وَاَصْحٰبُ الْمَشْئَمَةِ ۙ مَآ اَصْحٰبُ
الْمَشْئَمَةِ ۗ ﴿٩﴾ وَالسّٰبِقُوْنَ السّٰبِقُوْنَ ۙ ﴿١٠﴾

Yang artinya:

Kamu menjadi tiga golongan, yaitu golongan kanan, alangkah mulianya golongan kanan itu, dan golongan kiri, alangkah sengsaranya golongan kiri itu. Selain itu, (golongan ketiga adalah) orang-orang yang paling dahulu (beriman). Merekalah yang paling dahulu (masuk surga).

Di dalam Tafsir Al-Misbah (Shihab & Shihab, 2012), dijelaskan bahwa ayat ini menerangkan tiga golongan manusia pada hari kiamat berdasarkan karakteristik dan kualitas amalnya selama hidup. Jika amalnya baik maka mereka termasuk orang yang beruntung yakni golongan kanan. Jika amalnya tidak baik, maka termasuk golongan kiri, yakni golongan orang yang merugi. Dan golongan terakhir yakni golongan orang terdahulu yang beriman. Golongan teakhir ini adalah umat Nabi terdahulu yang beriman dan juga sebagian kecil umat Nabi Muhammad SAW.

Sejalan dengan itu pengelompokan berita juga bertujuan untuk mengenali berita yang sedang terjadi, hubungan antar berita, perbedaan pada setiap berita,

yang bertujuan untuk mengenali alur berita yang berjalan agar tidak tergiring opini yang keliru.

Didalam penjelasan surat ini, juga disebutkna hadis yang diriwayatkan oleh Ibnu Ḥibbān dan at-Tirmiżī dari Ibnu ‘Umar:

طَافَ رَسُوْلُ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ عَلَى رَاحِلَتِهِ الْقَصْوَاءِ يَوْمَ الْفَتْحِ وَاسْتَلَمَ الرُّكْنُ بِمِحْجَنِهِ وَمَا وَجَدَ لَهَا مُنَاخًا فِى الْمَسْجِدِ حَتَّى أُخْرِجَتْ إِلَى بَطْنِ الْوَادِيِّ فَأُنِيْخَتْ ثُمَّ حَمِدَ اللهَ وَأَثْنَى عَلَيْهِ ثُمَّ قَالَ: أَمَّا بَعْدُ أَيُّهَا النَّاسُ فَإِنَّ اللهَ قَدْ أَذْهَبَ عَنْكُمْ عُبِيَّةَ الْجَاهِلِيَّةِ يَا أَيُّهَا النَّاسُ إِنَّمَا النَّاسُ رَجُلاَنِ: بِرٌّ تَقِيٌّ كَرِيْمٌ عَلَى رَبِّهِ وَفَاجِرٌ شَقِيٌّ هَيِّنٌ عَلَى رَبِّهِ ثُمَّ تَلاَ (يَا أَيُّهَا النَّاسُ إِنَّا خَلَقْنَاكُمْ مِّنْ ذَكَرٍ وَّأُنْثَى وَجَعَلْنَاكُمْ شُعُوْبًا وَقَبَائِلَ لِتَعَارَفُوْا) حَتىَّ قَرَأَ الآيَةَ ثُمَّ قَالَ: أَقُوْلُ قَوْلِى هَذَا وَاَسْتَغْفِرُ اللهَ لِيْ وَلَكُمْ) (رواه ابن حبان والترمذي عن ابن عمر

yang artinnya :

“Rasulullah saw melakukan tawaf di atas untanya yang telinganya tidak sempurna (terputus sebagian) pada hari Fatḥ Makkah (Pembebasan Makkah). Lalu beliau menyentuh tiang Ka‘bah dengan tongkat yang bengkok ujungnya. Beliau tidak mendapatkan tempat untuk menderumkan untanya di masjid sehingga unta itu dibawa keluar menuju lembah lalu menderumkannya di sana. Kemudian Rasulullah memuji Allah dan mengagungkan-Nya, kemudian berkata, “Wahai manusia, sesungguhnya Allah telah menghilangkan pada kalian kesombongan dan keangkuhan Jahiliah. Wahai manusia, sesungguhnya manusia itu ada dua macam: orang yang berbuat kebajikan, bertakwa, dan mulia di sisi Tuhannya. Dan orang yang durhaka, celaka, dan hina di sisi Tuhannya. Kemudian Rasulullah membaca ayat: yā ayyuhan-nās innā khalaqnākum min żakarin wa unṡā… Beliau membaca sampai akhir ayat, lalu berkata, “Inilah yang aku katakan, dan aku memohon ampun kepada Allah untukku dan untuk kalian. (Riwayat Ibnu Ḥibbān dan at-Tirmiżī dari Ibnu ‘Umar).”

Disebutkan juga dalam hadis di atas pengelompokan manusia ada 2 macam yakni orang yang mulia dan orang yang celaka. Maka demikian, pengelompokan berita juga mengantarkan kita termasuk orang yang benar dalam memahami berita. Bukan termasuk orang yang keliru dalam memahami berita.

BAB V

KESIMPULAN

5.1 Kesimpulan

Pada penelitian ini sudah dilakukan beberapa tahap untuk melakukan pengujian. Diantaranya adalah menggunakan proses lematization POS tagging pada preprocessing data untuk mendapatkan informasi POS secara akurat, melakukan TF-IDF keyword extraction untuk mendapatkan 5 kata yang mewakili setiap dokumen. Menghitung kemiripan kontekstual dengan beberapa model yakni Fisrt POS, Max POS, dan Max No POS, dan clustering menggunakan K-Means.

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pengelompokan teks berita berbahasa Inggris berdasarkan kesamaan kontekstual dapat dilakukan secara efektif dengan mengombinasikan ekstraksi fitur TF-IDF dan pengayaan informasi semantik menggunakan WordNet, kemudian dikelompokkan menggunakan algoritma K-Means. Penambahan informasi semantik melalui pengukuran kemiripan Wu-Palmer terbukti mampu meningkatkan performa pengelompokan dibandingkan metode dasar berbasis TF-IDF, yang ditunjukkan oleh peningkatan nilai Silhouette Score, penurunan Davies-Bouldin Index, serta nilai SSE yang lebih baik. Integrasi semantic similarity juga menghasilkan nilai kemiripan rata-rata yang lebih tinggi dan distribusi standar deviasi yang lebih stabil dalam setiap klaster. Dari beberapa model yang diuji, model First POS menunjukkan performa terbaik dengan nilai Silhouette tertinggi sebesar 0,505 dan nilai Davies-Bouldin Index terendah

sebesar 0,667, serta memiliki beban komputasi yang lebih ringan karena hanya menggunakan satu synset dalam perhitungan kemiripan Wu-Palmer.

Selain itu, hasil evaluasi menggunakan metrik Silhouette Score, Davies-Bouldin Index, dan SSE menunjukkan bahwa model yang diusulkan mampu membentuk struktur cluster yang lebih baik dibandingkan metode baseline. Dengan demikian, pendekatan pengelompokan teks berbasis pengayaan semantik ini dapat menjadi alternatif yang efektif dalam mengelompokkan berita berbahasa Inggris berdasarkan kesamaan kontekstual, serta bersifat melengkapi sistem kategorisasi berita yang telah ada.

5.2 Saran

Penelitian selanjutnya disarankan untuk menggunakan dataset yang lebih besar dan lebih beragam agar diperoleh tingkat generalisasi model yang lebih komprehensif. Selain itu, pendekatan semantik berbasis WordNet pada penelitian ini dapat dikembangkan lebih lanjut dengan memanfaatkan metode contextual embedding seperti BERT, atau Sentence-BERT untuk memperoleh representasi konteks yang lebih mendalam.

Penelitian lanjutan juga dapat melakukan perbandingan hasil pengelompokan menggunakan algoritma lain, seperti DBSCAN, Agglomerative Clustering, atau Spectral Clustering, guna memperoleh gambaran performa metode yang lebih luas dalam pengelompokan teks berita.

DAFTAR PUSTAKA

Aggarwal, C. C. (2015). *Data Mining: The Textbook*. Springer International Publishing. https://doi.org/10.1007/978-3-319-14142-8

Aubaidan, B., Mohd, M., Albared, M., & Author, F. (2014). Comparative study of k-means and k-means++ clustering algorithms on crime domain. *Journal of Computer Science*, *10*(7), 1197–1206. https://doi.org/10.3844/jcssp.2014.1197.1206

Azzopardi, J., & Staff, C. (2012). Incremental clustering of news reports. *Algorithms*, *5*(3), 364–378. https://doi.org/10.3390/a5030364

Az-Zuhaili, W. (2013). *Tafsir Al-Munir Jilid 13 (Juz 25-26)* (I). Gema Insani. https://ia904603.us.archive.org/8/items/terjemah-tafsir-al-munir-mktbhazzaen/Terjemah%20Tafsir%20Al%20Munir%20-%2013.pdf

Bisandu, D. B., Prasad, R., & Liman, M. M. (2018). Clustering news articles using efficient similarity measure and N-grams. *International Journal of Knowledge Engineering and Data Mining*, *5*(4), 333. https://doi.org/10.1504/IJKEDM.2018.095525

Bora, N. N., Mishra, B. S. P., & Dehuri, S. (2012). Heuristic Frequent Term-Based Clustering of News Headlines. *Procedia Technology*, *6*, 436–443. https://doi.org/10.1016/j.protcy.2012.10.052

Bouras, C., & Tsogkas, V. (2012). A clustering technique for news articles using WordNet. *Knowledge-Based Systems*, *36*, 115–128. https://doi.org/10.1016/j.knosys.2012.06.015

Chen, Y., Qin, B., Liu, T., Liu, Y., & Li, S. (2010). The Comparison of SOM and K-means for Text Clustering. *Computer and Information Science*, *3*(2), 268–274. https://doi.org/10.5539/cis.v3n2p268

Chen, Z., Mi, C., Duo, S., He, J., & Zhou, Y. (2023). *ClusTop: An unsupervised and integrated text clustering and topic extraction framework*. 1–26.

Das, S., & Mert Cakmak, U. (2018). *Hands-On Automated Machine Learning*. Sciendo. https://doi.org/10.0000/9781788622288

Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

Et-taleby, A., Boussetta, M., & Benslimane, M. (2020). Faults Detection for Photovoltaic Field Based on K-Means, Elbow, and Average Silhouette

Techniques through the Segmentation of a Thermal Image. *International Journal of Photoenergy*, *2020*(1), 6617597. https://doi.org/10.1155/2020/6617597

Guan, R., Shi, X., Marchese, M., Yang, C., & Liang, Y. (2011). Text clustering with Seeds Affinity Propagation. *IEEE Transactions on Knowledge and Data Engineering*, *23*(4), 627–637. https://doi.org/10.1109/TKDE.2010.144

Han, J., Kamber, M., & Pei, J. (2012). Cluster Analysis. In *Data Mining* (pp. 443–495). Elsevier. https://doi.org/10.1016/B978-0-12-381479-1.00010-1

Jiang, Y., Liao, Y., & Yu, G. (2016). Affinity propagation clustering using path based similarity. *Algorithms*, *9*(3), 1–13. https://doi.org/10.3390/a9030046

Kim, S.-W., & Gil, J.-M. (2019). Research paper classification systems based on TF-IDF and LDA schemes. *Human-Centric Computing and Information Sciences*, *9*(1), 30. https://doi.org/10.1186/s13673-019-0192-7

Kumar Saksham. (n.d.). *Global News Dataset* [Dataset]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/7105651

Lim, Z.-Y., Ong, L.-Y., & Leow, M.-C. (2021). A Review on Clustering Techniques: Creating Better User Experience for Online Roadshow. *Future Internet*, *13*(9), 233. https://doi.org/10.3390/fi13090233

Ravi, J., & Kulkarni, S. (2023). Text embedding techniques for efficient clustering of twitter data. *Evolutionary Intelligence*, *16*(5), 1667–1677. https://doi.org/10.1007/s12065-023-00825-3

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Santhisree, M., & Damodaram, D. (2011). SSM-DBSCANand SSM-OPTICS : Incorporating a new similarity measure for Density based Clustering of Web usage data. *International Journal on Computer Science and Engineering*, *3*.

Saravanakumar, K. K., Ballesteros, M., Chandrasekaran, M. K., & McKeown, K. (2021). Event-driven news stream clustering using entity-aware contextual embeddings. *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, 2330–2340. https://doi.org/10.18653/v1/2021.eacl-main.198

Shihab, M. Q., & Shihab, M. Q. (2012). *Tafsîr Al-Mishbâḥ: Pesan, Kesan, dan Keserasian al-Qur'an Volume 13* (Cetakan V, Vol. 13). Lentera Haiti.

https://ia903106.us.archive.org/22/items/etaoin/Tafsir%20Al-Mishbah%20Jilid%2013%20-Dr.%20M.%20Quraish%20Shihab.pdf

Subakti, A., Murfi, H., & Hariadi, N. (2022). The performance of BERT as data representation of text clustering. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00564-9

Vergani, A. A., & Binaghi, E. (2018). A Soft Davies-Bouldin Separation Measure. *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–8. https://doi.org/10.1109/FUZZ-IEEE.2018.8491581

Wei, T., Lu, Y., Chang, H., Zhou, Q., & Bao, X. (2015). A semantic approach for text clustering using WordNet and lexical chains. *Expert Systems with Applications*, *42*(4), 2264–2275. https://doi.org/10.1016/j.eswa.2014.10.023

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics* -, 133–138. https://doi.org/10.3115/981732.981751

Yang, H. C., Lee, C. H., & Hsiao, H. W. (2015). Incorporating self-organizing map with text mining techniques for text hierarchy generation. *Applied Soft Computing Journal*, *34*, 251–259. https://doi.org/10.1016/j.asoc.2015.05.005

Yang, S., & Tang, Y. (2022). News topic detection based on capsule semantic graph. *Big Data Mining and Analytics*, *5*(2), 98–109. https://doi.org/10.26599/BDMA.2021.9020023

Yeasmin, S., Afrin, N., & Huq, M. R. (2023). Transformer-Based Text Clustering for Newspaper Articles. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, *490 LNICST*(June), 443–457. https://doi.org/10.1007/978-3-031-34619-4_35

Zhang, B., & Hou, Z. (2020). The study of an improved text clustering algorithm for self-organizing maps. *IOP Conference Series: Earth and Environmental Science*, *428*(1). https://doi.org/10.1088/1755-1315/428/1/012024

Zhang, Z., Chen, L., Yin, F., Zhang, X., & Guo, L. (2020). Improving Online Clustering of Chinese Technology Web News with Bag-of-Near-Synonyms. *IEEE Access*, *8*, 94245–94257. https://doi.org/10.1109/ACCESS.2020.2995516

Zheng, L., Li, L., Hong, W., & Li, T. (2013). PENETRATE: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems*

with Applications, *40*(6), 2127–2136. https://doi.org/10.1016/j.eswa.2012.10.029

Žižka, J., Dařena, F., & Svoboda, A. (2020). *Text mining with machine learning: Principles and techniques*. CRC Press, Taylor & Francis Group. https://doi.org/10.1201/9780429469275

Stecanella, B. (2019, May 11). *Understanding TF-ID: A Simple Introduction*. Diambil kembali dari monkeylearn.com: https://monkeylearn.com/blog/what-is-tf-idf/

Suyanto. (2017). *Data Mining untuk Klasifikasi dan Klasterisasi Data*. Bandung: Penerbit Informatika

LAMPIRAN

Langkah-langkah penghitungan dalam sistem

1. Tahapan TF-IDF Vectorization

Contoh data yang digunakan adalah 10 data dengan 5 kategori yang sudah melalui *preprocessing* dan pos tagging. Maka disini diambil bagian kolom bersih dan clean_pos_maps.

	article_id	bersih	clean_pos	category
0	98221	finally new work calvin hobbes bill watterson ...	{'finally': 'r', 'new': 'a', 'work': 'n', 'cal...	Art
1	98228	code geass get truly wild blu ray collection i...	{'code': 'n', 'geass': 'n', 'get': 'v', 'truly...	Art
2	98715	gen xer work home year want go back office com...	{'gen': 'n', 'xer': 'n', 'work': 'v', 'home': ...	Relationships
3	98718	criminal sanction buster exploit uk secrecy lo...	{'criminal': 'n', 'sanction': 'n', 'buster': '...	Relationships
4	99824	expect year black friday deal black friday sti...	{'expect': 'v', 'year': 'n', 'black': 'a', 'fr...	Beauty
5	99825	entrepreneurship skill billion dollar entrepre...	{'entrepreneurship': 'n', 'skill': 'n', 'billi...	Beauty
6	101318	linkedin hit billion user launch ai chatbot co...	{'linkedin': 'a', 'hit': 'n', 'billion': 'n', ...	Artificial Intelligence
7	101322	top meta scientist claim ai win wipe humanity ...	{'top': 'a', 'meta': 'n', 'scientist': 'n', 'c...	Artificial Intelligence
8	101815	amazon shopify seller use ai save time well un...	{'amazon': 'n', 'shopify': 'n', 'seller': 'n', ...	Motivation
9	101817	chatgpt prompt present less stress business us...	{'chatgpt': 'n', 'prompt': 'n', 'present': 'a'...	Motivation

Untuk menghitung TF-IDF digunakan data pada kolom bersih.

Contoh perhitungan pada dokumen 1:

finally new work calvin hobbes bill watterson graphic novel mystery cartoonist bill watterson end legendary comic strip calvin hobbes many fan wonder project would take next aside illustration piece write landscape painting

Menghitung TF:

TF (*work*) pada dokumen 1 = 1

TF (*founder*) pada dokumen 1 = 0

TF (*absurd*) pada dokumen 1 = 0

Sehingga,

TERM	TF									
	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
absurd	0	1	0	0	0	0	0	0	0	0
accuse	0	0	0	0	0	0	0	1	0	0
...	0	0	0	0	0	0	2	2	2	0
founder	0	0	0	0	0	2	0	1	0	0
..	0	0	0	0	0	1	0	0	0	0
work	1	0	2	0	0	0	0	0	0	2
	32	38	24	20	18	23	29	26	25	21

Menghitung DF:

term	df	dokumen_index
ai	3	6,7,8
work	3	0,2,9
billion	2	5,6
entrepreneur	2	5,8
founder	2	5,7
legendary	2	0,1
time	2	5,8
use	2	8,9
week	2	4,9
write	2	0,8
year	2	2,4
absurd	1	1
accuse	1	7
amazon	1	8
dst..	..	..

Menghitung IDF:

Untuk *term* dengan jumlah 3, 2, 1:

$$IDF\ (work) = log\ \frac{10+1}{3+1} + 1 = \ log\ \frac{11}{4} + 1 = log\ (2.75) + 1 = 2.0116$$

$$IDF\ (founder) = log\ \frac{10+1}{2+1} + 1 = \ log\ \frac{11}{3} + 1 = log\ (3.67) + 1 = 2.2993$$

$$IDF\ (absurd) = log\ \frac{10+1}{1+1} + 1 = \ log\ \frac{11}{2} + 1 = log\ (5.5) + 1 = 2.7047$$

Menghitung TF. IDF:

TF IDF (work) pada Dokumen 1 = 1 x 2.0116 = 2.0116

TF IDF (founder) pada Dokumen 1 = 0 x 2.2993 = 0

TF IDF (absurd) pada Dokumen 1 = 0 x 2.7047 = 0

Normalisai L2 Norm (Sklearn):

$$TF.IDF\ (L2\ Norm) = \frac{TF.IDF}{\sqrt{\sum TF.IDF^2}}$$

Hasil TF IDF pada Dokumen 1

TERM	TF.IDF
absurd	0
accuse	0
ai	0
…	…
wonder	2,7047
work	2,0116
would	2,7047
write	2,2993
xer	0
yann	0
year	0

$$\|TF.IDF\|\ (Dokumen\ 1) = \sqrt{\sum TF.IDF^2}$$

$$= \sqrt{0^2 + 0^2 + 0^2 + \cdots + 2{,}7047^2 + 2{,}0116^2 + 2{,}7047^2 + 2{,}29930^2 + 0^2 + \cdots}$$

$$= \sqrt{36.57831121}$$

$$= 6.048000596$$

$$TF.IDF\ (work) = \frac{2.0116}{6.048000596} = 0.119094405$$

$$TF.IDF\ (founder) = \frac{0}{6.048000596} = 0$$

$$TF.IDF\ (absurd) = \frac{0}{6.048000596} = 0$$

2. Tahapan Keyword Extraction

TF.IDF Dokumen 1

TERM	D0
absurd	0
accuse	0
ai	0
amazon	0
among	0
analyze	0
anime	0
appear	0

SORTED

TERM	D0
bill	0,320262696
calvin	0,320262696
hobbes	0,320262696
watterson	0,320262696
would	0,160131348
aside	0,160131348
cartoonist	0,160131348
comic	0,160131348

Apabila beberapa term memiliki bobot TF-IDF yang sama, maka pengurutan selanjutnya dilakukan berdasarkan indeks term dalam vocabulary. Dengan demikian, pemilihan kata kunci bersifat deterministik, yakni menghasilkan keluaran yang konsisten dan identik untuk setiap eksekusi apabila diberikan masukan dan parameter yang sama pada semua dataset.

Hasil ekstraksi untuk 10 data contoh :

doc	keywords
finally new work calvin hobbes bill watterson graphic novel mystery cartoonist bill watterson end legendary comic strip calvin hobbes many fan wonder project would take next aside illustration piece write landscape painting	{'watterson': 0.3203, 'hobbes': 0.3203, 'calvin': 0.3203, 'bill': 0.3203, 'would': 0.1601}
code geass get truly wild blu ray collection image crunchyroll crunchyroll late absurd mega collection anime box set bring together code geass one place throw entire chess set among thing good measure goro taniguchi legendary sci fi mecha	{'set': 0.2895, 'geass': 0.2895, 'crunchyroll': 0.2895, 'collection': 0.2895, 'code': 0.2895}
gen xer work home year want go back office combat loneliness gen xer hop return office seven year work	{'xer': 0.3476, 'office': 0.3476, 'loneliness': 0.3476, 'gen': 0.3476,

doc	keywords
remotely partly experience loneliness isolation	'year': 0.2955}
criminal sanction buster exploit uk secrecy loophole seychelles company link putin inner circle exploit uk loophole hundred firm bbc find	{'uk': 0.3922, 'loophole': 0.3922, 'exploit': 0.3922, 'seychelles': 0.1961, 'secrecy': 0.1961}
expect year black friday deal black friday still week away never early start make wishlist expect year event	{'friday': 0.4031, 'expect': 0.4031, 'black': 0.4031, 'year': 0.3427, 'wishlist': 0.2016}
entrepreneurship skill billion dollar entrepreneur vc free takeoff vcs fire founder ceo time control founder ceo control keep x wealth create seek create wealth	{'wealth': 0.3589, 'create': 0.3589, 'control': 0.3589, 'ceo': 0.3589, 'founder': 0.3051}
linkedin hit billion user launch ai chatbot coach wednesday november linkedin reveal reach billion member introduce artificial post linkedin hit billion user launch ai chatbot coach appear first readwrite	{'linkedin': 0.4298, 'billion': 0.3654, 'user': 0.2865, 'launch': 0.2865, 'hit': 0.2865}
top meta scientist claim ai win wipe humanity dangerous say former co researcher meta yann lecun spark row last weekend accuse prominent founder ai fear mongering	{'meta': 0.3784, 'ai': 0.2814, 'yann': 0.1892, 'wipe': 0.1892, 'win': 0.1892}
amazon shopify seller use ai save time well understand customer improve profit margin e commerce entrepreneur amazon shopify use ai seo write product description analyze review	{'shopify': 0.3697, 'amazon': 0.3697, 'use': 0.3143, 'ai': 0.2749, 'well': 0.1848}
chatgpt prompt present less stress business use five chatgpt prompt become present less stress work week boost business feel connected work	{'stress': 0.35, 'prompt': 0.35, 'present': 0.35, 'less': 0.35, 'chatgpt': 0.35}

3. Generate Similarity Matrix

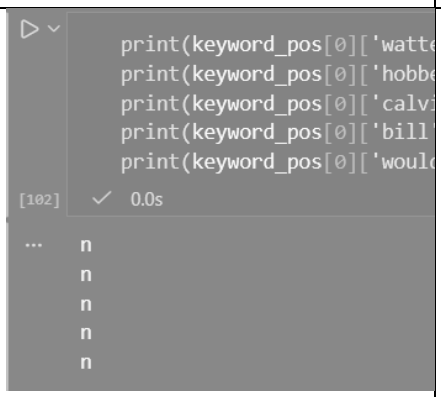
Pola penghitungan kesamaan makna terinspirasi dari term dari proses TF.IDF (*TF*) yang dijadikan fitur term sebanyak 193 kata. Maka untuk membuat itu matriks kesamaan, Term yang terbentuk pada TF dihitung kesamaan maknanya dengan 5 *keyword* dari masing-masing dokumen.

Disini *clean_pos_maps* bekerja untuk memberikan atribut fungsi kata yang ada ada di term dan 5 keyword per data.

	absurd	accuse	ai	amazon	...	year
[watterson, hobbes, calvin, bill, would]						
[set, geass, crunchyroll, collection,						
[xer, office, loneliness, gen, year]						
[uk, loophole, exploit, seychelles, secrecy]						
[friday, expect, black, year, wishlist]						
[wealth, create, control, ceo, founder]						
[linkedin, billion, user, launch, hit]						
[meta, ai, yann, wipe, win]						
[shopify, amazon, use, ai, well]						
[stress, prompt, present, less, chatgpt]						

	absurd	accuse	ai	amazon	...	year
watterson						
hobbes						
calvin						
bill						
would						

Pemanggilan Atribut Fungsi

Contoh pemanggilan atribut fungsi 5 keyword dokumen1 dengan *keyword_pos*:	Contoh pemanggilan atribut fungsi fitur *term* dengan *global_pos*:
	

Perhitungan Wu-Palmer First POS, Max POS, dan Max No. POS

Keyword *watterson* mempunyai fungsi kata noun “n”, tidak memiliki synset. Maka perhitungan langsung memberikan nilai 0. Jika kata ini sama, maka langsung diberikan nilai 1. Perhitungan dibawah ini hanya untuk kata yang

mempunyai synsets pada WordNet. Meskipun punya sysnets jika nilai yang dihasilkan None, maka nilai kemiripan=0.

$$sim_{wup}\begin{pmatrix} year,"n" \\ bill,"n" \end{pmatrix} = 2 \times \frac{depth(lcs(year,"a"\ bill,"n"))}{(depth(year,"a") + depth(bill,"n"))}$$

Untuk mencari depth atau kedalaman dari sebuat synset digunakan fungsi *min_depth()* dari WordNet. Berikut adalah hasilnya

Depth (bill,“n”) = 7

Depth (year,“n”) = 5

Ambil LCS (Lowest Common Subsumer) dengan fungsi berikut:

LCS adalah konsep paling umum pada kedua synsets

lcs = s1.lowest_common_hypernyms(s2)

Lcs = 1

Dengan Demikian,

$$sim_{wup}\begin{pmatrix} year,"n" \\ bill,"n" \end{pmatrix} = 2 \times \frac{1}{(5+7)} = \frac{2}{12} = 0{,}1666$$

Dalam percobaan penghitungan langsung dengan rumus dari NLTK yakni s1.wup_similariy(s2) hasilnya berbeda, yakni 0,2857. Nilai kesamaan Wu Palmer diperoleh menggunakan fungsi wup_similarity() pada pustaka NLTK WordNet. Meskipun rumus dasar Wu–Palmer diketahui, perhitungan kedalaman konsep dalam NLTK bergantung pada jalur hierarki internal yang tidak sepenuhnya diekspos, sehingga perhitungan manual dapat menghasilkan nilai yang berbeda.

Pada penelitian ini digunakan perhitungan langsung dengan library karena seluruh tahapan pemrosesan teks dan akses WordNet berada dalam satu ekosistem yang konsisten

Perhitungan wu-palmer untuk First POS, Max POS dan Max No POS semua sama. Yang membedakann adalah First POS langsung mengambil sysnet pertama dari atribut POS, Max POS mengambil kemiripan tertinggi dalam perhitungan Wu-Palmer antar sysnet dalam POS, dan Max No POS mengambil kemiripan tertinggi dari semua sysnet yang dimiliki sebuah kata tanpa memperhatikan POS.

Hasil perhitungan kemiripan menggunakan rumus *Sim = Mean-Std*

Keyword dok 1	**absurd**
watterson	0
hobbes	0,1667
calvin	0,1818
bill	0,1818
would	0

$$Mean = \frac{0+0\ 1667+0\ 1818+0\ 1818+0}{5} = 0\ 10606$$

$$Std = \sqrt{\frac{1}{N-1}\sum(x_i - x)^2}$$

$$\sum(x_i - x)^2 = (0 - 0\ 10606)^2 + (0\ 1667 - 0\ 10606)^2 + (0\ 1818 - 0\ 10606)^2 + (0\ 1818 - 0\ 10606)^2 + (0 - 0\ 10606)^2 = 0\ 03765$$

$$Std = \sqrt{\frac{1}{5-1}\ 0\ 03765} = \sqrt{0\ 0941} = 0\ 09702$$

Sim = 0,10606 – 0,0972 = 0,009044

Setelah semua dihitung maka menjadi Matriks Similarity untuk First POS:

	absurd	accuse	ai	amazon	among	analyze	anime	appear	artificial	aside	...
0	0.009044	0.008336	-0.003335	-0.040638	0.0	0.009044	0.011327	0.008336	0.009044	0.009044	...
1	0.007546	0.008460	-0.017216	0.008774	0.0	0.007546	0.002459	0.008460	0.007546	0.007546	...
2	0.068730	0.062744	0.059451	0.021765	0.0	0.068730	0.079145	0.062744	0.068730	0.068730	...
3	-0.015219	-0.008024	-0.001595	-0.011576	0.0	-0.015219	0.006185	-0.008024	-0.015219	-0.015219	...
4	0.057692	0.058610	0.058209	0.051321	0.0	0.057692	0.054472	0.058610	0.057692	0.057692	...
5	0.119934	0.119232	0.126038	0.066047	0.0	0.119934	0.155275	0.119232	0.119934	0.119934	...
6	0.051169	0.051661	0.057231	-0.040161	0.0	0.051169	0.062344	0.051661	0.051169	0.051169	...
7	-0.017307	-0.008389	-0.157981	0.005662	0.0	-0.017307	0.005271	-0.008389	-0.017307	-0.017307	...
8	0.047687	0.050655	-0.110152	-0.101814	0.0	0.047687	0.058743	0.050655	0.047687	0.047687	...
9	0.035401	0.040621	0.047084	0.041404	0.0	0.035401	0.044356	0.040621	0.035401	0.035401	...

10 rows × 193 columns

Penghapusan Zero Features Constant.

Dari perhitungan contoh ditemukan 19 term dengan nilai 0 untuk semua dokumen. Sehingga dari 193 term menjadi 174.

```
df_clean_contoh = del_cons_zero(df_contoh)
df_clean_contoh
```

✓ 0.0s

19

	absurd	accuse	ai	amazon	analyze	anime	appear	artificial	aside	away	...
0	0.009044	0.008336	-0.003335	-0.040638	0.009044	0.011327	0.008336	0.009044	0.009044	0.009044	...
1	0.007546	0.008460	-0.017216	0.008774	0.007546	0.002459	0.008460	0.007546	0.007546	0.007546	...
2	0.068730	0.062744	0.059451	0.021765	0.068730	0.079145	0.062744	0.068730	0.068730	0.068730	...
3	-0.015219	-0.008024	-0.001595	-0.011576	-0.015219	0.006185	-0.008024	-0.015219	-0.015219	-0.015219	...
4	0.057692	0.058610	0.058209	0.051321	0.057692	0.054472	0.058610	0.057692	0.057692	0.057692	...
5	0.119934	0.119232	0.126038	0.066047	0.119934	0.155275	0.119232	0.119934	0.119934	0.119934	...
6	0.051169	0.051661	0.057231	-0.040161	0.051169	0.062344	0.051661	0.051169	0.051169	0.051169	...
7	-0.017307	-0.008389	-0.157981	0.005662	-0.017307	0.005271	-0.008389	-0.017307	-0.017307	-0.017307	...
8	0.047687	0.050655	-0.110152	-0.101814	0.047687	0.058743	0.050655	0.047687	0.047687	0.047687	...
9	0.035401	0.040621	0.047084	0.041404	0.035401	0.044356	0.040621	0.035401	0.035401	0.035401	...

10 rows × 174 columns

Beberapa hasil bernilai negatif, maka dilakukan min mas scaler.

Contoh pada *term absurd:* min = -0.01731, max = 0,119934

Perhitungan untuk term absurd Dokumen 1

$$x' = \frac{0{,}009044 - (-0{,}01731)}{0\ 119934 - (-0{,}01731)} = 0\ 192001$$

Hasil perhitungan min-max scaller untuk semua term

	absurd	accuse	ai	amazon	analyze	anime	appear	artificial	aside	away	...
0	0.192001	0.131052	0.544490	0.364442	0.192001	0.058031	0.131052	0.192001	0.192001	0.192001	...
1	0.181087	0.132024	0.495619	0.658804	0.181087	0.000000	0.132024	0.181087	0.181087	0.181087	...
2	0.626906	0.557375	0.765554	0.736196	0.626906	0.501817	0.557375	0.626906	0.626906	0.626906	...
3	0.015209	0.002861	0.550617	0.537573	0.015209	0.024380	0.002861	0.015209	0.015209	0.015209	...
4	0.546475	0.524986	0.761183	0.912270	0.546475	0.340365	0.524986	0.546475	0.546475	0.546475	...
5	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...
6	0.498950	0.470536	0.757739	0.367284	0.498950	0.391874	0.470536	0.498950	0.498950	0.498950	...
7	0.000000	0.000000	0.000000	0.640264	0.000000	0.018400	0.000000	0.000000	0.000000	0.000000	...
8	0.473576	0.462649	0.168400	0.000000	0.473576	0.368315	0.462649	0.473576	0.473576	0.473576	...
9	0.384055	0.384026	0.722013	0.853195	0.384055	0.274168	0.384026	0.384055	0.384055	0.384055	...

10 rows × 174 columns

4. Proses K-Means Clustering

Langkah-langkah K-Means:

- Menentukan jumlah k-*Cluster,* misal k = 2
- Inisisasi centroid awal. Sebagai contoh centroid awal memakai dokumen indeks ke-0 dan ke-2 ($C_1 = D_0$) dan ($C_2 = D_2$)
- Menghitung jarak nilai centroid awal dengan masing-masing dokumen. Dalam hal ini dicontohkan dengan Dokumen indeks ke-1 (D_1)

$$d(D_1\ C_1)$$

$$= \sqrt{(0\ 181087 - 0\ 192001)^2 + (0\ 132024 - 0\ 131052)^2 + \cdots + (0\ 1663 - 0\ 13943)^2}$$

$$= \sqrt{8\ 742660947} = 2\ 956799105$$

$d(D_1\ C_2)$

$= \sqrt{(0\ 181087 - 0\ 626906)^2 + (0\ 132024 - 0\ 557375)^2 + \cdots + (0\ 1663 - 0)^2}$

$= \sqrt{28\ 68624041} = 5\ 355953734$

- Hitung jarak minimal

  $\min\big(d(D_1\ C_1)\ d(D_1\ C_2)\big)$

  $= \min(2\ 956799105\ \ 5\ 355953734)$

  $=\ 2\ 956799105$

  Maka anggota cluster 1 adalah D_0 dan D_1

- Update Centroid Baru (N=2)

$C1(new)$

$$= \frac{1}{2}\begin{bmatrix} 0{,}192001 + \ 0{,}181087 \\ 0{,}131052 + \ 0{,}132024 \\ \cdots \end{bmatrix}$$

=[0.186544, 0.1315378, ..., 0.152872]

- Iterasi sampai kondisi konvergen

Hasil cluster dengan k=2

corpus	category	cluster
Finally, new work from "Calvin & Hobbes's" Bill Watter:	Art	0
Code Geass Is Getting a Truly Wild Blu-ray Collection. In	Art	0
A Gen Xer who worked from home for 7 years wants to	Relationships	1
Criminals and sanctions-busters exploiting UK secrecy	Relationships	0
What To Expect From This Year's Black Friday Deals. Bla	Beauty	1
Entrepreneurship 361: 6 Skills From Billion-Dollar Entre	Beauty	1
LinkedIn hits 1 billion users launches AI chatbot coach	Artificial Intelli	1
A top Meta scientist's claims AI won't wipe out humani	Artificial Intelli	0
Amazon and Shopify sellers are using AI to save time, b	Motivation	1
5 ChatGPT Prompts To Be More Present And Less Stres:	Motivation	1

Elbow Method dengan SSE

$C_1 = D_0, D_1, D_3, D_7$

$C_2 = D_2, D_5, D_6, D_8, D_9$

Hitung Centroid Cluster

C1 dengan N=4,

$$= \frac{1}{4}\begin{bmatrix} 0{,}192001 + \ 0{,}181087 + 0\ 015208 + 0 \\ 0{,}131052 + \ 0{,}132024 + 0\ 002861 + 0 \\ \dots \end{bmatrix}$$

C1 new = [0.09707, 0.06648, 0.39768, ..., 0,16472]

C2 dengan N=6,

$$= \frac{1}{6}\begin{bmatrix} 0{,}626906 + \ 0\ 54647 + 1 + 0\ 4989 + 0\ 47357 + 0\ 38405 \\ 0\ 557374 + \ 0\ 52498 + 1 + \ 0\ 4705 + 0\ 46264 + \ 0\ 3840 \\ \dots \end{bmatrix}$$

C2 new = [0.58832, 0,56659, 0,69581, ..., 0,44059]

SSE C1

$D_0–C_1 = (0{,}192001 - 0{,}09707)^2 + (0{,}131052 - 0.06648)^2 + ... + (0{,}13943 - 0{,}16472)^2$

$D_0–C_1 = [0.009011043 + 0.004168994+, ..., +0{,}000639278] = 4{,}095115$

$D_1–C_1 = 2{,}848817993$

$D_3–C_1 = 2{,}765058696$

$D_7–C_1 = 3{,}371006991$

SSE C1 = 4,095115+2,848817993+ 2,765058696 + 3,371006991 = 13,08

SSE C2 = 58,85565096

SSE K=2 = 13,08 + 58,85565096 = 71,93565049

Demikian penghitungan SSE untuk K=2, untuk K >= 2 sebagai berikut,

```
k=2 | SSE=71.94
k=3 | SSE=36.26
k=4 | SSE=26.72
k=5 | SSE=21.00
k=6 | SSE=15.54
k=7 | SSE=11.35
k=8 | SSE=7.29
k=9 | SSE=3.56
```

Evaluasi Silhouette dan DBI

Silhouette Coefficient:

C1 = [D0, D1, D3, D7], A =4

C2 = [D2, D4, D5, D6, D8, D9], A=6

Menghitung *a(i),*

$$a_i = \frac{dist(D_0\ D_1) + dist(D_0\ D_3) + dist(D_0\ D_7)}{|4| - 1}$$

Hitung jarak anggota i ke anggota yang lain dalam 1 cluster,

$dist(D_0\ D_1)$

$$= \sqrt{(0\ 192001 - 0\ 181087)^2 + (0\ 13105 - 0\ 13202)^2 + \cdots + (0\ 13944 - 0\ 166309)^2}$$

$$= \sqrt{8\ 742660947} = 2\ 95679$$

Maka seterusnya,

$dist(D_0\ D_3) = 2\ 969614$

$dist(D_0\ D_7) = 3\ 449520$

$$a_{(D_0)} = \frac{2\ 95679 + 2\ 969614 + 3\ 449520}{|4| - 1} = 3\ 12531$$

Maka seterusnya,

$a_{(D_1)} = 2{,}854987771$

$a_{(D_3)} = 2{,}833961029$

$a_{(D_7)} = 2{,}954892088$

Menghitung *b(i),*

b(i) = min($d_{i,c}$)

Karena jumlah cluster 2, maka nilai minial jarak hanya 1 yaitu dari C1 ke C2. Hitung jarak anggota cluster i ke anggota cluster lain,

$dist(D_0\ D_2)$

$$= \sqrt{(0\ 192001 - 0\ 626906)^2 + (0\ 13105 - 0\ 55737)^2 + \cdots + (0\ 13944 - 0)^2}$$

$$= \sqrt{34\ 02848104} = 5\ 83339$$

Maka seterusnya,

$dist(D_0\ D_4) = 6{,}435738801$

$dist(D_0\ D_5) = 10{,}9102476$

$dist(D_0\ D_6) = 4{,}863894475$

$dist(D_0\ D_8) = 5{,}116499033$

$dist(D_0\ D_9) = 5{,}251795158$

$b(D_0) = \frac{5\ 83339+6\ 435739+\ 10\ 910248+\ 4\ 8638945+\ 5\ 116499+5\ 251795}{6} =$

$6\ 401928114$

Maka seterusnya,

$b(D_1) = 5{,}855753376$

$b(D_3)$ = 6,951912515

$b(D_7)$ = 6,66955172

$$s(D_0) = \frac{\left(b(D_0) - a(D_0)\right)}{max\left(b(D_0)\ a(D_0)\right)} = \frac{(6\ 401928 - 3\ 125311)}{6\ 401928}$$

$$= 0\ 511817171$$

Maka seterusnya nilai silhouette yang dihasilkan untuk k=2,

		a(i)	b(i)	s(i)
C1	D0	3,12531	6,40193	0,51182
	D1	2,85499	5,85575	0,51245
	D3	2,83396	6,95191	0,59235
	D7	2,95489	6,66955	0,55696
C2	D2	3,95283	6,01837	0,34321
	D4	3,83594	6,23127	0,3844
	D5	6,85874	10,9407	0,3731
	D6	4,59799	5,38311	0,14585
	D8	3,81344	5,28647	0,27864
	D9	4,1168	4,95878	0,1698
NILAI SILHOUETTE				0,38686

Davies-Bouldin Index:

C1 = [D0, D1, D3, D7], A =4

C2 = [D2, D4, D5, D6, D8, D9], A=6

Menghitung Centroid Cluster (Contoh Centroid C1),

C1 dengan N=4,

$$= \frac{1}{4}\begin{bmatrix} 0{,}192001 + & 0{,}181087 + 0\ 015208 + 0 \\ 0{,}131052 + & 0{,}132024 + 0\ 002861 + 0 \\ & \ldots \end{bmatrix}$$

C1 new = [0.09707, 0.06648, 0.39768, ..., 0,16472]

C2 dengan N=6,

$$= \frac{1}{6}\begin{bmatrix} 0{,}626906 + \ 0\ 54647 + 1 + 0\ 4989 + 0\ 47357 + 0\ 38405 \\ 0\ 557374 + \ 0\ 52498 + 1 + \ 0\ 4705 + 0\ 46264 + \ 0\ 3840 \\ \ldots \end{bmatrix}$$

C2 new = [0.58832, 0,56659, 0,69581, ..., 0,44059]

Menghitung jarak anggota Cluster dengan Centroidnya (Contoh C1),

$dist(D_0\ C_1)$

$$= \sqrt{(0\ 192001 - 0\ 097074)^2 + (0\ 13105 - 0\ 06648)^2 + \cdots + (0\ 13944 - 0\ 16472)^2}$$

$$= \sqrt{4\ 09511585} = 2\ 02364$$

Maka seterusnya,

$dist(D_1\ C_1)$= 1,68784

$dist(D_3\ C_1)$= 1,66285

$dist(D_7\ C_1)$= 1,83603

$$S_1 = \frac{2\ 02364 + \ 1\ 68784 + \ 1\ 66285 + \ 1\ 83603}{4} = \ 1\ 80259$$

$$S_2 = \frac{2\ 1323 + \ 1\ 9148 + \ \ 5\ 4526 + \ 3\ 1935 + 2\ 0587 + 2\ 54459}{6} = \ 2\ 88275$$

Menghitung jarak antar centroid $M_{ij,}$

$dist(C_1\ C_2) = M_{c1\ c2}$

$$= \sqrt{(0\ 097074 - 0\ 5883)^2 + (0\ 06648 - 0\ 56659)^2 + \cdots + (0\ 16472 - 0\ 44059)^2}$$

$$= \sqrt{33\ 15816382} = \ 5\ 7583$$

$$R_{1\,2} = R_{2\,1} = \frac{S_1 + S_2}{M_{c1\,c2}} = \frac{1\,80259 + 2\,88275}{5\,7583} = 0\,81366$$

$$DBI = \frac{R_{1\,2} + R_{2\,1}}{2} = 0\,81366$$

Demikian DBI untuk k=2 yaitu 0,81366

7. Mean Similarity dan Standar Deviasi

Perhitungan manual lengkap excel bisa dilihat disini:

TF_IDF, K-Means, SSE, Silhouette Coeffisient, Davies-Bouldin Indeks, Mean Similarity dan Standar Deviation

https://shorturl.at/GAYgc