Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2023106468
Vol. 10, No. 3, Juni 2023, hlm. 641-652 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

KLASTERISASI BERITA BAHASA INDONESIA DENGAN MENGGUNAKAN *K-MEANS* DAN *WORD EMBEDDING*

Humasak Tommy Argo Simanjuntak*[1]
Prince Ephraim Prabowo Silaban[2], Joshua Koko Sarasi Manurung[3], Venny Handayani Sormin[4]

[1,2,3,4] Institut Teknologi Del, Kabupaten Toba
[1]humasak@del.ac.id [2]silabanprince@gmail.com [3]joshuamanurung777@gmail.com
[4]vennyhandayani00@gmail.com
*Penulis Korespondensi

(Naskah masuk: 05 Agustus 2022, diterima untuk diterbitkan: 21 Juni 2023)

Abstrak

Jumlah berita atau dokumen yang sangat melimpah merupakan sumber pengetahuan yang sangat berharga dan dapat digunakan untuk memperoleh wawasan dalam pengambilan keputusan. Namun, pertumbuhan jumlah berita dengan dimensi yang tinggi menjadi sebuah tantangan besar, yang menyebabkan sulitnya informasi pada berita dikategorikan secara efisien dan cepat. Kesulitan ini semakin kompleks dengan tidak adanya kelas atau label pada berita tersebut. Analisis konten dari berita yang belum memiliki kelas atau label dapat dilakukan dengan pendekatan *data mining*. Salah satu metode *data mining* yang dapat digunakan untuk mengelompokkan berita tanpa label, jumlah yang besar, dan sulit dilakukan secara manual adalah klastering. Klastering teks adalah salah satu metode penambangan data yang bertujuan untuk mengelompokkan dokumen berdasarkan kesamaan atau kemiripan di antara teks. Penelitian ini memberikan pendekatan baru dalam mengelompokkan berita Bahasa Indonesia dengan metode klastering, dimana ekstraksi fitur dilakukan melalui pendekatan *Neural Network* (*Word Embedding*) yang dapat menunjukkan kesamaan antar kata untuk mempertahankan semantik dan konteks dari kata yang ada pada berita. Sumber data yang digunakan adalah berita dari portal berita "Tempo" yang terdiri dari 520863 berita. Hasil penelitian menunjukkan bahwa jumlah klaster k = 4, dengan parameter *Word Embedding*: *min_count*=1 dan *embedding_size*=300 memberikan nilai *silhouette coefficient* terbaik sebesar 0.73. Hasil klasterisasi berita divisualisasikan dalam bentuk dimensi yang berbeda dan visualisasi *World Cloud* untuk menganalisis dan mengevaluasi metode yang diusulkan pada penelitian ini.

Kata kunci: Pengelompokan Berita, Klasterisasi Berita, *K-Means*, *Word2vec.*

CLUSTERING INDONESIA NEWS USING K-MEANS AND WORD EMBEDDING

Abstract

The enormous amount of news or documents is a precious source of knowledge and can be used to gain insight into decision-making. However, the growth in the number of news stories with high dimensions is a big challenge, making it difficult for information on the news to be categorized efficiently and quickly. This difficulty is further complicated by the absence of classes or labels on the news. Analysis of the content of news that does not yet have a class or label can be done with a data mining approach. The most used data mining method to group a tremendous amount of news without class labels is clustering. Text clustering is a data mining task that aims to group documents based on similarities. This study provides a new approach to classifying Indonesian news with the clustering method, where feature extraction is carried out through a Neural Network (Word Embedding) approach that can show similarities between words to maintain the semantics and context of the words in the news. The data source used is news from the news portal "Tempo," which consists of 5208063 news. The results showed that the number of clusters k = 4, with Word Embedding parameters: min_count=1 and embedding_size=300, produced the best silhouette coefficient value of 0.73. The results of news clustering were visualized in the form of different dimensions and World Cloud visualization to analyze and evaluate the proposed method.

Keywords: *News Grouping, News Clustering, K-Means, Word2vec*

1. PENDAHULUAN

Pesatnya penggunaan dan adopsi internet telah memacu pertumbuhan dan pertukaran informasi yang sangat pesat dibandingkan era sebelumnya sehingga mengakibatkan jumlah informasi terus meningkat secara eksponensial hingga mencapai lebih dari 550 triliun dokumen (EMCHA, WIDYAWAN, & ADJI,

2019). Perkembangan ini memungkinkan informasi diakses lebih mudah, namun jumlah berita yang besar menimbulkan banyak isu dan tantangan besar. Ketika seseorang membutuhkan dan mencari berita terkait topik tertentu, jumlah berita yang diterima bisa sangat banyak, beragam dan tidak dapat dikategorikan. Hal ini menjadi suatu permasalahan bagi pembaca dikarenakan pengguna harus mencari berita dengan cara umum yang membutuhkan waktu yang lama dan *resource* yang banyak. Selain itu, mesin pencarian seperti *search engine* pun akan mengalami kesulitan untuk menghasilkan berita yang saling berhubungan dan memiliki kesamaan topik. Oleh karena itu, pertumbuhan jumlah berita dengan dimensi yang tinggi menjadi sebuah tantangan besar, yang menyebabkan sulitnya informasi pada berita dikategorikan secara efisien dan cepat. Kesulitan ini semakin kompleks dengan tidak adanya kelas atau label pada berita tersebut. Dengan demikian terdapat suatu urgensi penyelesaian dari masalah yang disebutkan di atas. Hal ini dapat diatasi dengan pengelompokan berita ke dalam kelompok yang sama berdasarkan kesamaan kata, berita atau informasi yang ada di dalamnya.

Klasterisasi berita merupakan suatu metode dalam penambangan data (*data mining*) yang dapat membantu pembaca untuk mengumpulkan berita dari berbagai sumber dan menyajikannya dalam bentuk kelompok atau klaster. Berita yang berada dalam satu klaster adalah berita-berita yang yang memiliki topik yang sama atau dekat, dan berita yang berada pada klaster yang berbeda adalah berita dengan tingkat kesamaan topik yang sangat kecil. Tingkat kesamaan (*similarity*) berita dalam klaster yang sama dan tingkat perbedaan (*dissimilarity*) berita pada klaster yang berbeda merupakan metrik untuk mengukur kualitas dari klaster berita yang dihasilkan. Selain itu, jumlah berita yang sangat besar akan sangat mempengaruhi performa klasterisasi berita. Oleh karena itu, tahapan untuk melakukan klasterisasi berita mulai dari ekstraksi fitur sampai pembangunan model perlu dikembangkan agar dapat menghasilkan performansi klasterisasi yang baik.

Beberapa penelitian telah dilakukan untuk menghasilkan klasterisasi berita. Namun, penelitian tersebut masih dilakukan pada berita dalam Bahasa Inggris seperti CNN, BBC and Aljazeera (FONSEKA, 2019), menggunakan prinsip klasterisasi dokumen dengan pendekatan algoritma AntClass (ONAN, 2017), menggunakan jumlah teks yang singkat seperti media sosial twitter dan reddit (CURISKIS, DRAKE, OSBORN, & KENNEDY, 2020) (LIM, KARUNASEKERA, & HARWOOD, 2017). Selain itu sebagian besar penelitian terkait dengan berita, banyak yang mengarah pada klasifikasi terkait dengan deteksi berita palsu (LI, GUO, WANG, & ZHENG 2021) dan *sentiment analysis* (KHARDE, & SONAWANE 2016). Namun, penelitian untuk mengklasterisasi berita Bahasa Indonesia masih sedikit dan menggunakan pendekatan tradisional. Salah satu pendekatan tradisional yang dilakukan untuk klasterisasi berita Bahasa Indonesia adalah menggunakan kombinasi *partitional clustering* dengan pembobotan kata (SLAMET, RAHMAN, RAMDHANI, & DARMALAKSANA, 2016). Algoritma *K-Means* adalah salah satu metode yang paling popular untuk kasus ini, yang melakukan pengelompokan data atau objek dengan sistem partisi (*partitional clustering*) dan memiliki tingkat komputasi yang cepat dan efisien. TF-IDF merupakan metode pembobotan kata yang umum digunakan untuk proses ekstraksi fitur dari berita dan dikombinasikan dengan *K-Means*. Salah satu keuntungan TF-IDF adalah metode yang sederhana. Ekstraksi fitur yang dilakukan dengan TF-IDF menghasilkan bobot vektor namun tidak memberikan kemiripan terkait dengan konteks atau semantik kata, sehingga belum menghasilkan kualitas klaster yang baik. Berdasarkan permasalahan tersebut, penelitian ini memberikan solusi dengan melakukan ekstraksi fitur dari berita dengan memperhitungkan konteks maupun semantik dari setiap kata pada berita. Pada penelitian ini, pengelompokan berita dilakukan dengan terlebih dahulu melakukan ekstraksi fitur dengan menggunakan *Word Embedding* yang mengimplementasikan konsep arsitektur *neural network*. Penerapan *Word Embedding* pada dokumen berita akan menghasilkan informasi terkait kesamaan semantik satu kata dengan kata lain. *Word Embedding* akan menemukan semantik kata berdasarkan kemiripan kata sehingga suatu kata dapat mewakili kata lain ketika sedang diterapkan pada sebuah model. Metode *Word Embedding* dengan *Word2Vec* akan memberikan vektor unik untuk setiap kata berdasarkan kata-kata yang muncul di sekitar kata tertentu, tidak seperti TF-IDF. *Word2Vec* memperhitungkan penempatan kata-kata dalam dokumen (sampai batas tertentu).

Oleh karena itu, penelitian ini memiliki kontribusi utama untuk menghasilkan sebuah framework atau pendekatan baru yang mengkombinasikan teknik *Word Embedding* dengan Word2Vec dan *partitional clustering* sebagai sebuah *unsupervised* model untuk memodelkan klasterisasi berita bahasa Indonesia.

Paper ini disusun dalam 5 Bab dengan detail sebagai berikut. Bagian 2 Studi Literatur berisi pembahasan dari penelitian yang sudah dilakukan terkait dengan *Unsupervised Learning* pada berita Bahasa Indonesia. Bagian 3 Gambaran Umum penelitian dan metode penelitian yang diusulkan, Bagian 4 Hasil dan Pembahasan penelitian yang dilakukan, dan Bagian 5 berisi Kesimpulan dan Saran dari penelitian.

2. STUDI LITERATUR

Dalam klasterisasi teks algotima *K-Means* merupakan salah satu algoritma yang umum digunakan. *K-Means* merupakan sebuah algoritma *clustering* yang mengelompokkan *n* buah objek ke dalam k klaster berdasarkan jaraknya dengan pusat klaster (JAIN & DUBES, 1988). Algoritma *K-Means* umum digunakan dalam kasus *unsupervised learning* karena algoritma *K-Means* mudah diimplementasikan dan kompleksitas waktunya linear. Seperti penelitian yang berjudul "*Clustering* Berita Berbahasa Indonesia" pada 4718 berita dari www.kompas.com (WIBISONO & KHODRA, 2006). Penelitian ini menggunakan metode TF-IDF untuk merepresentasikan koleksi berita sebagai sebuah vektor term yang akan menjadi masukan kepada algoritma *K-Means* untuk melakukan *clustering*. Metode yang sama juga digunakan pada penelitian yang berjudul "Klasterisasi Dokumen Web (Berita) Bahasa Indonesia Menggunakan Algoritma *K-Means*" (HUSNI, NEGARA, & SYARIEF, 2015). Berdasarkan kedua penelitian tersebut, hasil klasterisasi dengan menggunakan pembobotan vektor TF-IDF dan K-Means belum menghasilkan akurasi yang optimal. Salah satu teknik yang dapat diperbaiki adalah pada tahap prapemprosesan teks dan tahap ekstraksi fitur. Metode TF-IDF yang digunakan untuk pembobotan vektor merupakan metode yang masih tradisional, dimana pembobotan vektor memerlukan matriks *sparse* yang besar yang juga membutuhkan komputasi yang besar.

Penelitian lain dalam klasterisasi berita Bahasa Indonesia adalah penelitian dengan judul "*Clustering Articles in* Bahasa Indonesia *using Self-Organizing Map*" (GUNAWAN, AMALIA, & CHARISMA, 2017). Penelitian ini menggabungkan metode TF-IDF dan algoritma *Self Organizing Map* untuk klasterisasi berita. Sama seperti sebelumnya, klasterisasi berita Bahasa Indonesia dengan *Self Organizing Map* (SOM) dan TF-IDF belum menghasilkan akurasi yang baik. Hal ini dikarenakan kekurangan pada metode TF-IDF yang digunakan untuk mentransformasi berita menjadi vektor. Selain itu, TF-IDF juga tidak mendukung dokumen berita yang cenderung mirip dan dokumen berita yang memiliki *text explosion* atau jumlah teks yang terlalu banyak. Oleh sebab itu, maka perlu dilakukannya klasterisasi berita Bahasa Indonesia dengan metode yang lebih modern dan dapat menghasilkan klaster berita yang memiliki akurasi yang lebih sempurna. Pada penelitian ini, peneliti akan melakukan klasterisasi berita Bahasa Indonesia menggunakan algoritma *K-Means* untuk menghasilkan klaster dan *Word Embedding* sebagai ekstraksi fitur untuk menggantikan TF-IDF.

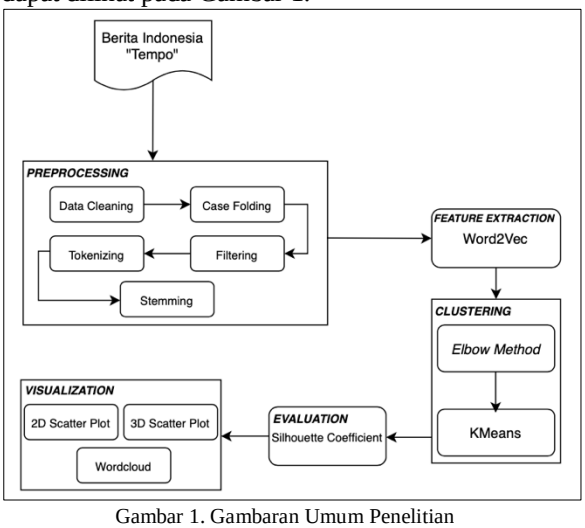
Word embedding adalah suatu metode untuk merepresentasikan kata dengan vektor bilangan real yang kontinu dengan panjang tetap. *Word embedding* memetakan kata dalam kosakata ke ruang vektor laten dimana kata-kata dengan konteks yang sama memiliki jarak yang berdekatan. Melalui *word embedding*, sebuah kata diubah menjadi vektor yang merangkum informasi sintaksis dan semantik kata tersebut (WANG, NULTY, & LILLIS, 2020).

Word2Vec adalah salah satu teknik *work embedding* yang populer digunakan melalui pendekatan *neural network*, yang dikembangkan oleh Tomas Mikolov pada tahun 2013 di Google (MIKOLOV, CHEN, CORRADO, & DEAN, 2013). Berdasarkan arsitektur *neural network*, *Word2Vec* memiliki 2 jenis yaitu "Skip-gram" dan "Continous Bag of Word" (CBOW). Arsitektur tersebut hanya terdiri atas 3 lapisan yaitu *Input*, *Hidden Layer*, dan *Output*. Beberapa penelitian sebelumnya sudah menggunakan Word2Vec untuk mentransformasikan kata menjadi vektor, diantaranya: Pembangunan ringkasan teks dalam Bahasa Indonesia dengan menggunakan *Neural Network* (RIKE, SUYANTO, & WISESTY, 2019) dan pembangunan ringkasan teks Bahasa Inggris dengan menggunakan algoritma *Clustering* (CAI, LIN, MA, & JIANG, 2019). Kedua penelitian ini menunjukkan bahwa model Word2Vec mampu melakukan ekstraksi fitur kata dengan mempertimbangkan semantiknya. Proses kalkulasi yang dilakukan tidak hanya mempertimbangkan kesamaan antar kata, tetapi juga kesamaan pada seluruh kalimat.

Berdasarkan penelitian terdahulu tersebut, *Word2Vec* mampu mempertahankan arti semantik dari kata-kata yang berbeda dalam sebuah dokumen, sehingga informasi konteks tidak hilang. Hal ini memberikan pengaruh terhadap performansi proses pembelajaran pada teks Bahasa Indonesia secara *unsupervised*. Pendekatan Word2Vec juga menghasilkan ukuran vektor *embedding* yang sangat kecil sehingga komputasi lebih cepat.

3. METODE PENELITIAN

Beberapa tahapan penelitian yang diusulkan untuk melakukan klasterisasi berita Bahasa Indonesia dapat dilihat pada Gambar 1.



Gambar 1. Gambaran Umum Penelitian

Adapun proses yang dilakukan pada gambaran umum sistem yang ditampilkan pada gambar 1 di atas adalah sebagai berikut.

1. *Text Preprocessing*
   Pada tahap *text preprocessing* dilakukan *data cleaning*, *filtering*, *tokenizing*, *case folding*, *stemming*.
2. *Feature Extraction*
   Data yang sudah diproses pada tahap *text preprocessing* direpresentasikan menjadi bentuk vektor dengan menggunakan metode *Word2Vec*.
3. *Clustering*
   Algoritma K-Means digunakan pada tahapan *Clustering*, yang membagi atau mengelompokkan data yang tidak berlabel ke dalam beberapa klaster berdasarkan analisis kesamaan (*similarity*) maupun ketidaksamaan (*dissimilarity*) yang ada pada dataset untuk kemudian memperoleh pola keterhubungan antar data.
4. *Evaluation*
   Pada tahap ini dilakukan evaluasi terhadap hasil klastering dengan menggunakan metode *Silhoutte Coefficient*.
5. *Visualization and Analysis*
   Pada tahap ini hasil klaster yang didapatkan akan divisualisasikan dalam bentuk 2D dan 3D, serta melakukan analisis hasil klaster.

3.1 *Text Preprocessing*

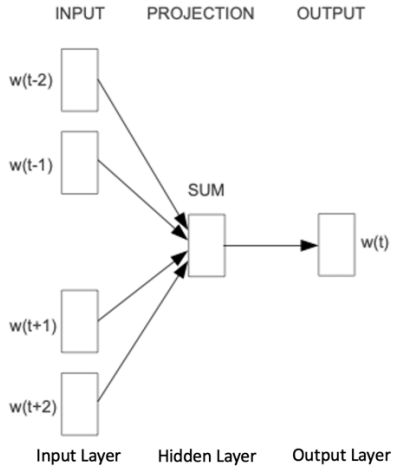
Tahap ini bertujuan untuk melakukan pra pemrosesan pada data berita sehingga mengurangi kata-kata yang tidak perlu dan mempermudah proses selanjutnya. Tahapan *text preprocessing* yang dilakukan yaitu, *data cleaning, filtering, tokenizing, case folding*, *stemming*.

a) *Data cleaning* dilakukan untuk menghapus atau menghilangkan data yang duplikat. Pada proses ini *library* python yang digunakan adalah *pandas library* (McKINNEY, 2021).
b) *Case Folding*
   Proses untuk menyeragamkan bentuk karakter maupun kata pada keseluruhan teks yang terdapat dalam berita, baik menjadi huruf kecil atau huruf kapital. *Case folding* akan menghasilkan kata pada berita dalam suatu bentuk yang standar. *Library* yang digunakan pada proses *case folding* adalah *library Natural Language Toolkit* (NLTK, 2021).
c) *Tokenizing*
   Tokenizing adalah proses untuk membagi teks pada berita menjadi token, yang dapat berupa kata, frasa, maupun simbol. Proses *Tokenizing* menggukanan *library Natural Language Toolkit* (NLTK, 2021).
d) *Filtering*
   Filtering adalah pengecekan yang dilakukan untuk mencari dan menemukan *stopword* yang terdapat di dalam teks. Jika pada teks ditemukan kata yang merupakan *stopword*, maka kata tersebut dihapus. Proses *filtering* dilakukan dengan menggunakan *library Natural Language Toolkit* (NLTK, 2021).
e) *Stemming*
   Stemming bertujuan untuk mengekstrak kata dasar (*root word*) dari suatu kata. Pada *Stemming* berita berbahasa Indonesia, *library* yang digunakan adalah *library* Sastrawi, yang dimana *library* Sastrawi merupakan *library stemmer* yang digunakan untuk mengatasi masalah perubahan kata pada teks bahasa Indonesia menjadi kata dasar (ROSID, FITRANI, ASTUTIK, & MULLOH, 2020).

3.2 Ekstraksi Fitur dengan *Word2Vec*

Word2Vec adalah metode ekstraksi fitur yang menggunakan konsep *word embedding* untuk merepresentasikan kata menjadi vektor. *Word2Vec* bekerja dengan mengambil korpus teks sebagai *input* dan kemudian merepresentasikan kata yang ada pada korpus menjadi bentuk vektor sebagai *output*. Vektor kata yang dihasilkan digunakan untuk mengukur jarak kedekatan antar vektor kata dengan vektor kata yang lain. *Word2Vec* menggunakan ide *neural network* untuk melatih model dan merepresentasikan setiap kata menjadi vektor (LI, et al., 2018). Arsitektur *neural network Word2Vec* terdiri atas 3 layer, yaitu *Input Layer*, *Hidden Layer*, dan *Output Layer*.

Word2Vec mempunyai dua jenis arsitektur pemodelan yang dapat digunakan untuk merepresentasikan kata menjadi bentuk vektor. Arsitektur tersebut yaitu *continuous bag-of-word* (*CBOW*) dan *Skip-Gram*. Pada penelitian ini arsitektur *Word2Vec* yang digunakan adalah *CBOW*. *CBOW* merupakan model *word embedding* yang memprediksi target kata ketika diberikan konteks (*input*) yang ada di sekitarnya. *CBOW* menggunakan teknik prediksi kata saat ini yang digunakan sebagai target dari konteks (sebagai sebuah *inputan*) (PUTRI, 2020). Arsitektur *CBOW* dilihat pada Gambar 2.



Gambar 2. Arsitektur *CBOW* (MIKOLOV, CHEN, CORRADO, & DEAN, 2013)

Proses *training* pada arsitektur *CBOW* memiliki kompleksitas yang proporsional dengan:

$$O = E \times T \times Q \quad (1)$$

dimana:

$$Q = N \times D + D \times log_2(V) \quad (2)$$

dengan:
E: jumlah *training epochs*
T: jumlah kata pada *training set*
N: jumlah kata sebelumnya pada *input layer*
D: representasi kata (*word*)
V: ukuran dari *vocabulary*

3.3 Klasterisasi dengan *K-Means*

K-Means adalah salah satu algoritma *clustering* dengan sistem partisi, dimana *K-Means* mengelompokkan setiap objek data ke dalam beberapa klaster, dimana satu klaster mempunyai tingkat kemiripan yang tinggi, dan karakteristik yang berbeda dengan dengan objek pada klaster lain (SLAMET, RAHMAN, RAMDHANI, & DARMALAKSANA, 2016). Algoritma *K-Means* mengambil jumlah klaster k sebagai *input* dan menghasilkan titik *centroid* akhir sebagai *output*. Algoritma *K-Means* memilih titik awal *centroid* secara acak, dan jumlah iterasi untuk mencapai *centroid* klaster dipengaruhi oleh kandidat *centroid* klaster awal (sangat bergantung pada inisialisasi awal). Sehingga ditemukan pada konstruksi algoritma *K-Means* dengan menentukan *centroid* klaster dilihat dari densitas data awal yang tinggi untuk mendapatkan performansi yang lebih tinggi (MANIK, et al., 2018). Pada penelitian ini eksperimen dilakukan dengan menggunakan *K-Means++*, dimana algoritma ini memiliki kelebihan yang tidak tergantung pada inisialisasi awal dari titik *centroid*. Secara umum jika diberikan observasi (x_1, x_2, x_3), *K-Means* bertujuan untuk melakukan partisi sejumlah n observasi ke dalam k klaster (k <= n), S = {S1, S2, …, S3}, sehingga K-Means mencoba meminimalkan jarak antara objek dalam satu klaster. Secara formal, memiliki persamaan sebagai berikut.

$$\arg\min_s \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 = \arg\min_s \sum_{i=1}^{k} |S_i| \, Var \, S_i \quad (3)$$

Berikut ini adalah urutan proses yang dilakukan oleh algoritma K-Means:

1. Memilih secara random titik k sebagai *initial centroid* dari k klaster.
2. ***For*** setiap *point I* yang tersisa ***do***
3. Alokasikan *I* ke klaster dengan *centroid* terdekat
4. *Update centroid* klaster (yang memuat *i)* atau Hitung *centroid* klaster
5. Kembali ke *point* 3
6. ***End For***

3.4 Evaluasi

Evaluasi klaster dilakukan untuk mengetahui seberapa baik data dikelompokkan berdasarkan model klasterisasi yang sudah ditetapkan. Pada penelitian ini, peneliti menggunakan *silhouette coefficient* sebagai metode evaluasi klaster. Metode ini memberikan representasi singkat tentang seberapa baik objek pada klasternya. *Silhoutte coefficient* memiliki persamaan sebagai berikut.

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (4)$$

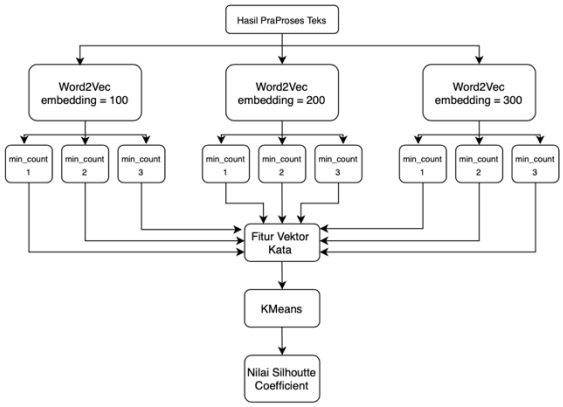
3.5 Skenario Eksperimen

Eksperimen penelitian menggunakan 520863 artikel berita Tempo yang terdiri atas atribut *content*, *tag*, *date time*, dan *title* seperti yang ditunjukkan pada Tabel 1. Fitur yang digunakan pada eksperimen ini adalah fitur *content* (isi dari artikel berita). Rata-rata jumlah kata yang terdapat pada fitur *content* adalah 265 kata dan jumlah maksimum adalah 7083 kata.

Tabel 1. Struktur Dataset Artikel Tempo

Atribut	Tipe Data	Keterangan
Title	*String*	Judul berita
Date time	*Numerik*	Tanggal dan waktu berita
Tag	*String*	metadata yang membantu untuk menjelaskan suatu hal dan memungkinkan ditemukan ketika melakukan pencarian didalam berita.
Content	*String*	Isi dari berita

Eksperimen dilakukan dengan mempertimbangkan beberapa nilai parameter yang digunakan pada *word embedding* dan jumlah klaster pada K-Means. Parameter *word embedding* yang digunakan adalah *embedding size* dan *min_count*. *Embedding size* adalah jumlah dimensi dari *embedding*, dimana nilai *default* adalah 100. *Min-count* adalah minimum jumlah kata yang dipertimbangkan ketika proses *training* model dilakukan. Kata-kata yang jumlah kemunculannya kurang dari jumlah *min_count* akan diabaikan. Nilai *default* untuk min_*count* adalah 5. Selanjutnya, peneliti akan mengamati hasil (vektor) dari seluruh kombinasi nilai parameter *embedding size* dan *min_count* yang akan digunakan sebagai *inputan K-Means.* Peneliti mengamati hasil yang paling optimal yang memberikan representasi data yang paling baik. Selanjutnya hasilnya akan dievaluasi secara kuantitatif menggunakan metode *Silhoutte Coefficient*. Skenario dari eksperimen yang dilakukan dapat dilihat pada Gambar 3.



Gambar 3. Skenario Eksperimen

Adapun nilai parameter yang digunakan yaitu *embedding size* 100, 200, 300 dan *min count* 1, 2, 3.

4. HASIL DAN PEMBAHASAN

4.1 Hasil *Data Preprocessing*

a) *Data Cleaning*

Penerapan *data cleaning* menghasilkan perubahan pada dataset berita dari segi jumlah. Pada dataset jumlah baris data awal adalah 520863. Pada dataset ditemukan *missing value* sebanyak 2541 pada atribut *tags*, maka ketika dilakukannya *data cleaning* untuk menghapus *missing value*, jumlah baris data untuk atribut *content*, *datetime*, dan *title* tetap sebanyak 520863 data dan 518321 untuk atribut *tags*.

b) *Case Folding*

Untuk menyamakan bentuk kata, maka setelah *Case folding* diimplementasikan, semua kata yang terdapat pada dataset berita berubah menjadi huruf kecil.

Tabel 2. Hasil Penerapan *Case Folding*

Sebelum *Case Folding*	**Sesudah *Case Folding***
Menurut Rano, sejak dimulainya sistem ganjil genap di Bandung, masih ada beberapa pengendara yang belum mengetahui penerapannya	menurut rano, sejak dimulainya sistem ganjil genap di Bandung, masih ada beberapa pengendara yang belum mengetahui penerapannya

c) *Tokenizing*

Hasil implementasi *tokenizing* adalah semua kata pada teks berita diubah ke dalam bentuk token.

Tabel 3. Hasil Penerapan *Tokenizing*

Sebelum *Tokenizing*	**Sesudah *Tokenizing***
menurut rano, sejak dimulainya sistem ganjil genap di Bandung, masih ada beberapa pengendara yang belum mengetahui penerapannya	[menurut, rano, sejak, dimulainya, sistem, ganjil, genap, di, bandung, masih, ada, beberapa, pengendara, yang, belum, mengetahui, penerapan]

d) *Filtering*

Hasil implementasi *filtering* adalah semua kata-kata yang kurang penting pada data teks berita dihapus seperti *stopword* dan *special character*.

Tabel 4. Hasil Penerapan *Filtering*

Sebelum *Filtering*	**Sesudah *Filtering***
[menurut, rano, sejak, dimulainya, sistem, ganjil, genap, di, bandung, masih, ada, beberapa, pengendara, yang, belum, mengetahui, penerapan]	[menurut, rano, sejak, dimulainya, sistem, ganjil, genap, bandung, masih, ada, beberapa, pengendara, belum, mengetahui, penerapan]

e) *Stemming*

Hasil implementasi *stemming* adalah semua kata pada teks berita telah berubah menjadi kata dasar. *Stemming* dilakukan dengan menggunakan *library* Sastrawi.

Tabel 5. Hasil Penerapan *Stemming*

Sebelum *Stemming*	**Sesudah *Stemming***
[menurut, rano, sejak, dimulainya, sistem, ganjil, genap, di, bandung, masih, ada, beberapa, pengendara, yang, belum, mengetahui, penerapan]	[menurut, rano, sejak, mulai, sistem, ganjil, genap, bandung, masih, ada, berapa, kendara, belum, ketahui, terap]

4.2 Ekstraksi Fitur

Pada ekstraksi fitur dilakukan beberapa skenario eksperimen dengan beberapa nilai parameter *embedding size* dan *min count* yang berbeda. Parameter yang memiliki nilai *silhouette coefficient* tertinggi digunakan sebagai parameter untuk ekstraksi fitur.

Ekstraksi fitur menghasilkan representasi kata dari dataset berita yang digunakan. Berikut ini adalah contoh salah satu representasi vektor dari kata "virus" setelah melalui tahap ekstraksi fitur dengan menggunakan *Word2Vec*.

```
[ ]  model.wv['virus']

array([ 1.2627857 , -0.9191024 , -1.374116  , -1.3159034 ,  1.4672178 ,
        0.52911866,  0.2783673 , -1.0310416 ,  0.88108684, -0.9891492 ,
        0.8762697 , -1.5180212 ,  0.01489236, -0.9755481 ,  1.2691535 ,
        1.6767533 , -1.4363345 ,  0.91646975,  0.15044343,  0.9960944 ,
        0.3476531 , -0.64635456, -0.56815547,  1.0399821 ,  0.3810048 ,
       -2.8459573 , -0.6728131 ,  2.1308188 , -0.05652368,  1.9499674 ,
       -1.0787761 , -1.3698075 , -0.8291219 , -0.11668108,  0.48059732,
        1.2310326 ,  1.8532585 ,  1.5034171 , -0.5580325 , -0.4615626 ,
        0.7870531 ,  0.04957006, -1.1112318 , -0.49992377,  1.5680217 ,
       -1.4498391 ,  1.563933  , -0.6927016 , -1.5090649 ,  1.2916104 ,
       -1.1769277 , -1.0208627 , -0.10345536,  0.7720318 ,  0.43739307,
       -1.0579474 , -0.6933134 , -2.327713  , -0.1504926 ,  0.22239494,
        1.7164793 ,  0.56791025, -0.86105216,  3.5048964 ,  0.35155407,
        0.8456959 ,  0.8056187 , -0.07194825,  1.1473627 ,  1.6134605 ,
        0.0608822 , -0.7324726 , -1.1462193 , -0.6494072 , -1.8269331 ,
        0.9746852 , -0.8963956 ,  0.7490295 , -0.5021839 , -0.82983214,
        1.3002942 , -0.12349906, -1.380177  ,  0.2752893 , -1.5392913 ,
       -0.8704376 ,  1.056402  ,  1.9445553 , -1.0922724 , -0.6040193 ,
       -0.25022584, -1.9471872 ,  1.4947852 , -0.6188385 ,  1.7413374 ,
        0.05768671, -0.8445164 , -2.7932594 , -0.07813434, -1.6842742 ,
       -0.49214762,  2.682     ,  0.14354971, -0.11332215, -0.9947444 ,
        0.31674352, -0.50319123, -1.4590269 , -1.899006  ,  0.30736926,
       -0.13092099,  1.1099216 , -0.9860226 ,  1.1005741 , -0.9034377 ,
        0.12920375, -1.1688124 ,  0.70627713,  0.9376494 ,  0.5990917 ,
        1.655556  , -0.608371  ,  0.12584308,  1.4937811 ,  0.20610404,
       -0.45939216,  1.9506489 ,  0.6578413 ,  0.7959883 ,  0.16632949,
```

Gambar 4. Representasi Vektor kata "virus"

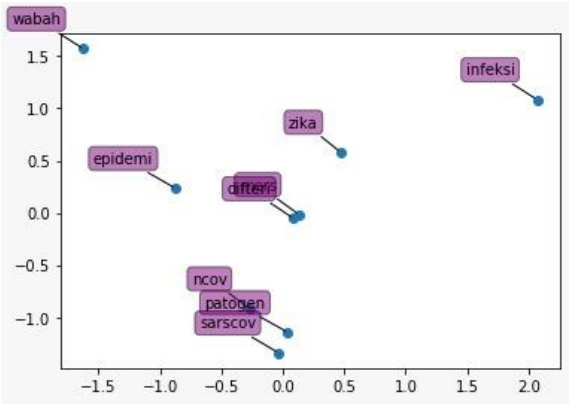
Setiap kata yang telah dimodelkan dengan *Word2Vec* juga memiliki bobot *similarity* dengan kata yang lain. Gambar 5 adalah kata-kata yang memiliki *similarity* dengan kata "virus".

```
[ ]  model.wv.most_similar("virus")

[('mers', 0.6341812610626221),
 ('wabah', 0.6177667379379272),
 ('ncov', 0.6010279059410095),
 ('virusvirus', 0.5998170375823975),
 ('infeksi', 0.5869833827018738),
 ('zika', 0.5786353349685669),
 ('merscov', 0.5757984519004822),
 ('patogen', 0.568328857421875),
 ('covid', 0.5610857605934143),
 ('kolera', 0.5585897564888)]
```

Gambar 5. Representasi Similarity Vektor kata "virus"

Jika nilai *similiarity* semakin mendekati 1, maka kata tersebut memiliki kesamaan yang tinggi secara semantik, hal ini dapat dibuktikan melalui Gambar 6.

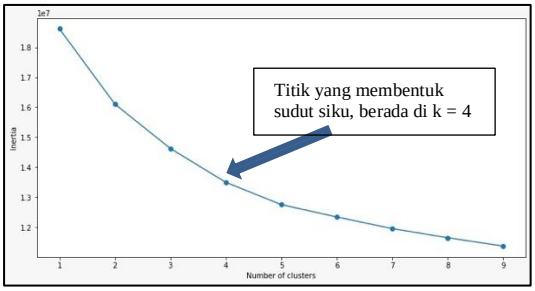


Gambar 6. Representasi kedekatan kata "virus" berdasarkan nilai *similarity*

Berdasarkan Gambar 6 tersebut, tiap *term* atau kata akan diplot berdasarkan hasil vektor yang sudah direduksi menggunakan PCA. Setiap *term* dipetakan ke beberapa klaster kata dengan hasil vektor yang saling berdekatan. Jumlah klaster yang terbentuk sesuai dengan jumlah masukan kata dan hasil vektor tiap kata.

4.3 *K-Means*

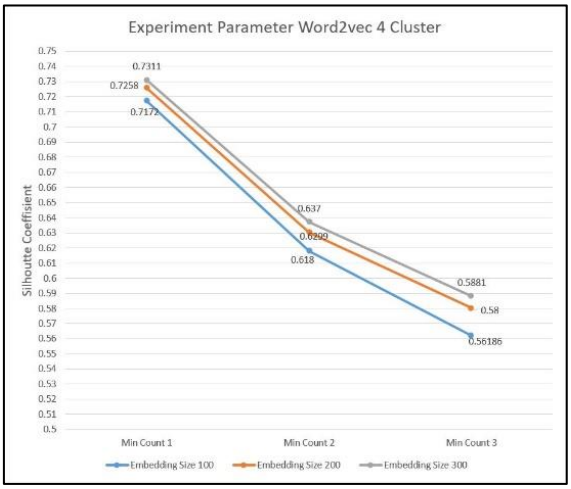
Setelah ekstraksi fitur selesai dilakukan, dilanjutkan dengan klasterisasi menggunakan algoritma *K-Means*. Pada klasterisasi, data masukan adalah hasil ekstraksi fitur dan juga nilai k, sebagai jumlah klaster. Nilai k yang digunakan ditentukan melalui *elbow method*, yang bertujuan untuk mendapatkan nilai k yang optimal untuk klaster yang akan dibuat. Hasil *elbow method* dapat dilihat melalui grafik pada Gambar 7



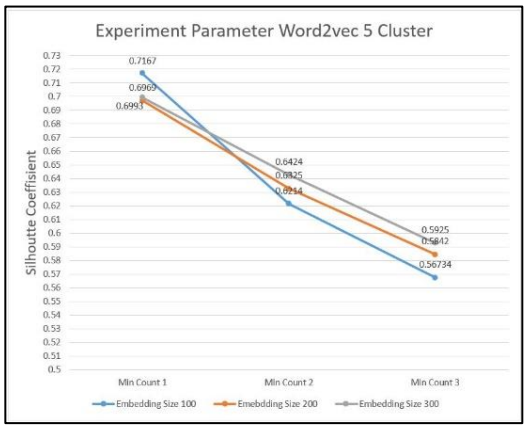
Gambar 7. Penentuan jumlah klaster dengan *elbow method*

Berdasarkan Gambar 7 di atas dapat dilihat bahwa grafik yang menunjukkan jumlah klaster optimal adalah sudut yang membentuk siku terdapat pada jumlah klaster = 4 (empat). Setelah nilai k yang digunakan diperoleh, maka selanjutnya dilakukan klasterisasi. Pada eksperimen ini, selain nilai k=4, peneliti juga mencoba nilai k= 5, 6, 7 untuk menunjukkan perbandingan nilai evaluasi pada beberapa jumlah klaster yang berbeda.

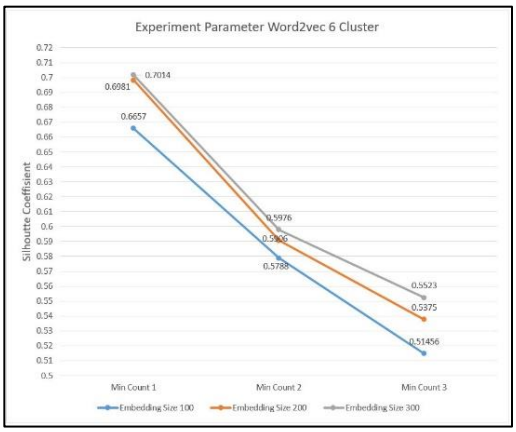
Berdasarkan ekperimen yang dilakukan dengan menggunakan beberapa kombinasi nilai parameter, maka hasil dari eksperimen tersebut dapat dilihat melalui Gambar 8 -11.



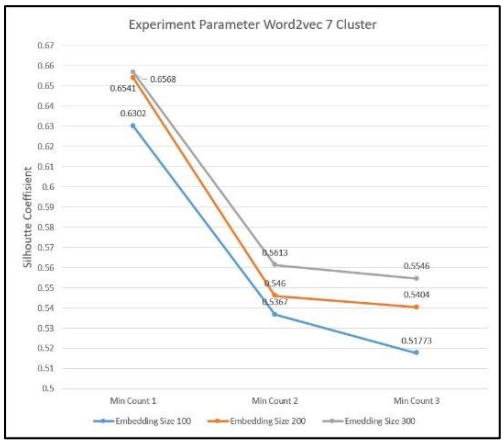
Gambar 8. Hasil Eksperimen Parameter *Word2Vec 4* Klaster



Gambar 9. Hasil Eksperimen Parameter *Word2Vec* 5 Klaster



Gambar 10. Hasil Eksperimen Parameter *Word2Vec* 6 Klaster



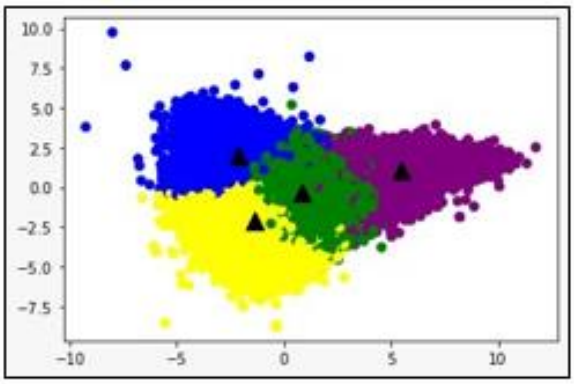
Gambar 11. Hasil Eksperimen Parameter *Word2Vec* 7 Klaster

Melalui grafik yang ditunjukkan oleh Gambar 8 - 11 dapat dilihat bahwa *embedding size* 300 dengan *min count* 1 memiliki nilai *silhouette coefficient* tertinggi. Pada grafik juga ditunjukkan bahwa jumlah klaster yang paling optimal untuk mengelompokkan berita teks Bahasa Indonesia adalah 4, dengan nilai *silhouette coefficient* = 0.7311. Oleh karena itu, struktur klaster yang dihasilkan sudah memiliki struktur yang kuat (*0.71 < SC < 1.00*). Hasil klasterisasi berita yang diperoleh dengan menggunakan 4 klaster adalah klaster pertama berisi 139307 artikel berita, klaster kedua berisi 64792 artikel berita, pada klaster ketiga berisi 1744001 dan kluster 4 berisi 142762.

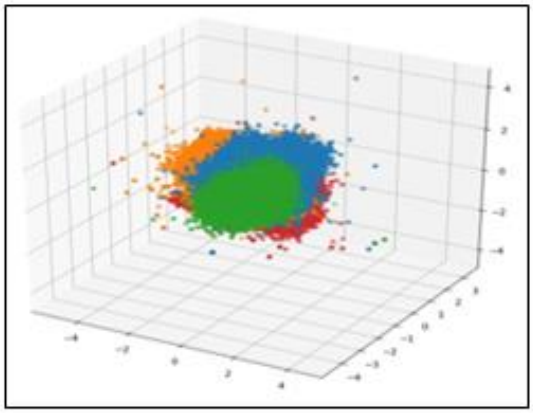
Grafik pada gambar 8-11 juga menunjukkan bahwa parameter *min_count* dan *embedding size* Word2Vec mempengaruhi kualitas klaster yang dibentuk. Semakin tinggi nilai *min_count*, maka kualitas klaster yang dihasilkan semakin rendah. Eksperimen menunjukkan bahwa frekuensi jumlah minimal kata = 1 adalah yang terbaik, atau dengan kata lain semua kata yang terdapat pada berita (setelah *pre-processing* teks) dipertimbangkan untuk di klaster, karena memiliki tingkat kepentingan yang sama. Kemudian, nilai *embedding size* yang lebih besar juga memberikan kualitas klaster yang lebih baik. Hal ini menunjukkan bahwa nilai dimensi vektor yang mewakili setiap token atau kata pada teks berita sebaiknya besar, mempertimbangkan juga jumlah data teks yang digunakan.

4.3.1 Visualisasi Klaster

Hasil klasterisasi yang diperoleh divisualisasikan secara 2D dan 3D sehingga dapat menunjukkan gambaran sebaran objek pada keempat klaster tersebut.



Gambar 12. Visualisasi Klaster 2 Dimensi



Gambar 13. Visualisasi Klaster 3 Dimensi

Berdasarkan visualisasi klaster yang disajikan dalam bentuk 2D dan 3D tersebut, dapat dilihat

bahwa setiap objek yang memiliki kesamaan (warna yang sama) telah berkumpul dalam suatu klaster, begitu juga dengan objek-objek lain yang memiliki warna yang berbeda.

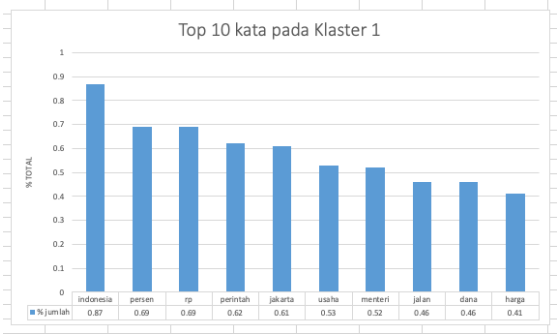
4.3.2 Analisis Hasil Klaster

Pada bagian ini, setiap klaster akan diamati dan diverifikasi untuk menentukan topik berita yang terdapat pada setiap klaster. Analisis hasil klaster dilakukan dengan menggunakan Visualisasi *Word Cloud.*

1. Analisis Hasil Klaster 1

Gambar 14. *Word Cloud Cluster* 1

Berdasarkan Gambar 14 di atas terlihat bahwa kata-kata yang dominan pada klaster 1 adalah kata-kata seperti *uang*, *sri*, *mulyani*, *triliun*, *juta*, *bank*, *mandiri*, *miliar*, *suku bunga*, *investasi*, *ekonomi*, *rp*, *persen dan dana*. Secara detail, persentase frekuensi dari *top* 10 kata-kata yang paling sering muncul dari keseluruhan data pada klaster 1 ditampilkan pada Gambar 15.



Gambar 15. *Top* 10 kata pada Klaster 1

Berdasarkan analisis kata yang terdapat pada klaster yang dihasilkan, maka dapat disimpulkan bahwa klaster 1 dominan berisi informasi terkait **ekonomi**.

2. Analisis Hasil Klaster 2

Berdasarkan Gambar 16 di atas terlihat bahwa kata-kata yang dominan pada klaster 2 adalah kata-kata seperti presiden, pasang, calon, partai, gerindra, golkar, mahkamah, konstitusi, anggota, dpr, wakil, rakyat dan gubernur.

Gambar 16. *Word Cloud Cluster* 2

Secara detail, persentase frekuensi dari *top* 10 kata-kata yang paling sering muncul dari keseluruhan data pada klaster 2 ditampilkan pada Gambar 17.

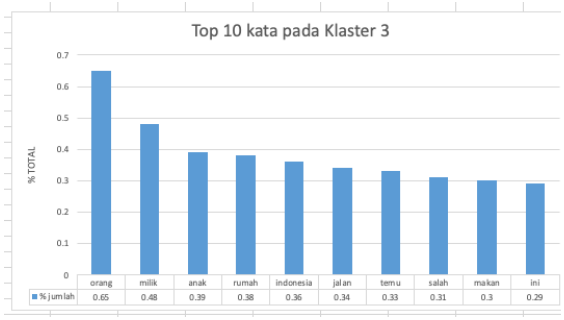
Gambar 17. *Top* 10 kata pada Klaster 2

Berdasarkan analisis yang dihasilkan, dapat disimpulkan bahwa klaster 2 dominan berisi informasi terkait **politik** di Indonesia.

3. Analisis Hasil Klaster 3

Gambar 18. *Word Cloud Cluster* 3

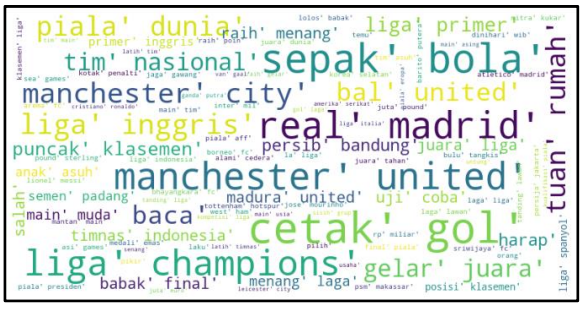
Berdasarkan Gambar 18 di atas terlihat bahwa kata-kata yang dominan pada klaster 3 adalah kata-kata seperti *jakarta*, *pemerintah*, *masyarakat*, *media*, *sosial*, *bencana*, *jalan*, *dan tol*. Secara detail, persentase frekuensi dari *top* 10 kata-kata yang paling sering muncul dari keseluruhan data pada klaster 3 ditampilkan pada Gambar 19.



Gambar 19. *Top* 10 kata pada Klaster 3

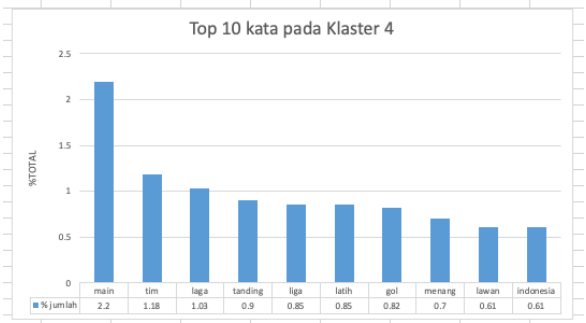
Berdasarkan analisis yang dihasilkan, dapat disimpulkan bahwa klaster 3 dominan berisi informasi terkait **sosial**.

4. Analisis Hasil Klaster 4



Gambar 20. *Word Cloud Cluster* 4

Berdasarkan Gambar 20 di atas terlihat bahwa kata-kata yang dominan pada klaster 4 adalah kata-kata seperti *manchester, madrid, liga, inggris, totenham, messi, timnas, sepak, bola, pemain, taji, piala dan klasemen*. Persentase frekuensi dari *top* 10 kata yang paling sering muncul dari keseluruhan data pada klaser 4 ditampilkan pada Gambar 21.



Gambar 21. *Top 10 kata pada Cluster* 4

Berdasarkan analisis yang dihasilkan, dapat disimpulkan bahwa klaster 4 dominan berisi informasi terkait **olahraga**.

4.4 Evaluasi Hasil Klaster

Eksperimen pada penelitian ini menggunakan *Word2Vec* dengan pengaturan sejumlah parameter seperti *embedding size* dan *min_count* yang berbeda. Hasil eksperimen menunjukkan kualitas dan kekuatan klaster terbaik yang dihasilkan pada *silhouette coefficient* tertinggi **0.73** dengan parameter *embedding size* 300 dan *min_count* 1. Hasil ini menunjukkan bahwa klaster yang diperoleh sudah menunjukkan struktur yang kuat baik dalam satu klaster maupun antar klaster. Hal ini diverifikasi secara visual dengan *Word Cloud* yang menunjukkan bahwa setiap klaster yang terbentuk memiliki topik tertentu yang spesifik dan kata-kata dalam satu klaster yang terkait secara semantik dengan topik pada klaster tersebut.

5. KESIMPULAN DAN SARAN

Penggunaan model *Word2Vec* dengan *embedding size* 100 dan *min_count* 1 sebagai parameter yang digunakan dalam model *Word2Vec* memberikan nilai akurasi terbaik. Hal ini disebabkan karena *embedding size* 100 dan *min_count* 1 sesuai untuk menghasilkan fitur dari data yang digunakan. Ukuran *embedding size* dan *min_count* disesuaikan dengan ukuran data yang digunakan sehingga kata yang direpresentasikan sebagai vektor memberikan semantik dan menunjukkan makna teks berita.

Pemilihan jumlah k klaster yang sesuai juga mempengaruhi algoritma *K-Means*. Hasil terbaik dari algoritma *K-Means* diperoleh pada jumlah klaster 4 dengan nilai *silhouette coeficient* 0.73. Perbandingan nilai *silhouette coefficient* yang dihasilkan oleh jumlah k klaster yang berbeda menunjukkan bahwa penentuan nilai k klaster mempengaruhi hasil klasterisasi. Terlalu banyak nilai k yang digunakan dapat menghasilkan klaster yang kurang optimal karena penyebaran objek yang tidak stabil pada setiap klaster. Hasil eksperimen menunjukkan kombinasi dari *word embedding* dengan *partitional clustering* sebagai sebuah *unsupervised* model berhasil mengklasterisasi berita Bahasa Indonesia. Ekstraksi fitur dengan *word embedding* mempengaruhi hasil klasterisasi berita Bahasa Indonesia.

Pada penelitian selanjutnya, dibutuhkan penerapan suatu metode dalam menentukan nilai setiap parameter *Word2Vec*. Upaya lain yang dilakukan dalam penelitian mendatang ialah dengan menggunakan teks berita selain Tempo dan jumlah data yang lebih banyak, serta struktur *neural network* pada *word embedding* yang lebih kompleks untuk melakukan ekstraksi fitur.

DAFTAR PUSTAKA

CAI, Z., LIN, N., MA, C. & JIANG, S., 2019. Indonesian Automatic Text Summarization Based on A New Clustering Method in Sentence Level. *In Proceedings of the 2019 International Conference on Big Data Engineering*, pp.30-35. New York:Association for Computing Machinery.

CURISKIS, S. A., DRAKE, B., OSBORN, T. R. & KENNEDY, P. J., 2020. An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Information Processing & Management*, 57(2), p.102034.

EMCHA, A. C., WIDYAWAN & ADJI, T. B., 2019. Quotation Extraction from Indonesian Online News. *In 2019 International Conference on Information and Communications Technology (ICOIACT)*, pp. 408-412. Yogyakarta:IEEE.

FONSEKA, W. P. I., 2019. Automated News Clustering Using an Unsupervised Learning Model. *Master Project Final Report*. University of Colombo School of Computing.

GUNAWAN, D., AMALIA & CHARISMA, I., 2017. Clustering Articles in Bahasa Indonesia using Self-Organizing Map. *In 2017 International Conference on Electrical Engineering and Informatics (ICELTICs)*, pp. 239-244. Banda Aceh:IEEE.

HUDIN, M., FAUZI, M. & ADINUGROHO, S., 2018. Implementasi *Metode* Text Mining dan K-Means Clustering untuk Pengelompokan Dokumen Skripsi (Studi Kasus: Universitas Brawijaya). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11), pp. 5518-5524. Malang:Filkom Universitas Brawijawa.

HUSNI, NEGARA, Y. D. P. & SYARIEF, M., 2015. Clusterisasi Dokumen Web (Berita) Bahasa Indonesia Menggunakan Algoritma K-Means. *Jurnal SimanteC*, 4(3), pp. 159-166. Bangkalan:Fakultas Teknik Universitas Trunojoyo Madura.

JAIN, A. K. & DUBES, R. C., 1988. Algorithm for CLustering Data. New Jersey:Prentice-Hall, Inc.

KHARDE, V. A. & SONAWANE, S. S., 2016. *Sentiment* Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications*, 139(11), pp. 5-15. New York:Foundation of Computer Science(FCS).

LI, C., LU, Y., WU, J., ZHANG, Y., Xia, Z., WANG, T., YU, D., CHEN, X., LIU. P. & GUO, J., 2018. LDA Meets Word2Vec: A Novel Model for Academic Abstract Clustering, *In Companion Proceedings of the web conference 2018*, pp. 1699-1706. Lyon:Association for Computing Machinery.

LI, D., GUO, H., WANG, Z. & ZHENG, Z., 2021. *Unsupervised* Fake News Detection Based on Autoencoder. *IEEE Access, 9*, pp. 29356-29365. IEEE.

LIM, K. H., KARUNASEKERA, S. & HARWOOD, A., 2017. ClusTop: A Clustering-based Topic Modelling Algorithm for Twitter using Word Networks. *IEEE International Conference on Big Data (BIGDATA)*, pp. 2009-2018. Boston:IEEE.

MANIK, L. P., SYAFIANDINI, A. F., MUSTIKA, H. F., ABKA, A. F. & RIANTO, Y., 2018. Evaluating the Morphological and Capitalization Features for Word Embedding-Based POS Tagger in Bahasa Indonesia, *In 2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 49-53. Tangerang:IEEE.

McKINNEY, W., 2021. *pandas.* [daring] Tersedia di: https://pandas.pydata.org [Diakses 2021].

MIKOLOV, T., CHEN, K., CORRADO, G. & DEAN, J., 2013. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

NLTK, T., 2021. *Natural Language Toolkit.* [daring] Tersedia di: https://www.nltk.org [Diakses 2021].

ONAN, A., 2017. A K-medoids Based Clustering Scheme with An Application to Document Clustering. *In 2017 International Conference on Computer Science and Engineering (UMBK)*, pp. 354-359. Antalya:IEEE.

ONAN, A., BULUT, H. & KORUKOGLU, S., 2017. An Improved Ant Algorithm with LDA-based Representation for Text Document Clustering. *Journal of Information Science*, 43(2), pp.275-292. SAGE.

PRABOWO, Y. D., MARSELINO, T. L. & SURYAWIGUNA, M., 2019. Pembentukan Vector Space Model Bahasa Indonesia Menggunakan Metode Word to Vector. *Jurnal Buana Informatika*, 10(1), pp. 29-40. Yogyakarta:Universitas Atma Jaya.

PUTRI, S. K., 2020. Pre-Train Word Vector Bahasa Indonesia Generation Dengan Menggunakan Word2vec untuk Bidang Komputer dan Teknologi Informasi (Skripsi Sarjana). Universitas Sumatera Utara.

RIKE, A., SUYANTO, S. & WISESTY, U. N., 2019. Indonesian Abstractive Text Summarization Using Bidirectional Gated Recurrent Unit Gated Recurrent Unit. *Procedia Computer Science*, 157, pp. 581-588. Elsevier B.V.

ROSID, M. A., FITRANI, A. S., ASTUTIK, I. R. I. & MULLOH, N. I., 2020. Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi. *IOP Conference Series: Materials Science and Engineering*, 874(1), p.012017. IOP Publishing.

SLAMET, C., RAHMAN, A., RAMDHANI, M. A. & DARMALAKSANA, W., 2016. Clustering the Verses of the Holy Qur'an using K-Means Algorithm. *Asian Journal of Information Technology*, 15(24), pp. 5159-5162.

WANG, C., NULTY, P. & LILLIS, D., 2020. A Comparative Study on Word Embeddings in Deep Learning for Text Classification. *In Proceedings of the 4th International Conference on Natural Language Processing and*

Information Retrieval, pp. 37–46. New York:Association for Computing Machinery.

WIBISONO, Y. & KHODRA, M. L., 2006. Clustering *Berita Bahasa Indonesia*. s.l., s.n., pp. 495-496.

WIDYASTUTI, N. N., BIJAKSANA, A. & SARDI, I. L., 2018. Analisis Word2vec untuk Perhitungan Kesamaan Semantik Antar Kata. *e-Proceeding of Engineering*, 5(3), pp. 7603-7612. Universitas Telkom.