KLASTERISASI DOKUMEN MENGGUNAKAN WEIGHTED K-MEANS BERDASARKAN RELEVANSI TOPIK

Muhammad Riduwan[1)], Chastine Fatichah[2)], dan Anny Yuniarti[3)]
[1, 2, 3)] Departemen Informatika, Institut Teknologi Sepuluh Nopember
Surabaya
e-mail: ridwan.yaa71@gmail.com[1)], chastine@if.its.ac.id[2)], anny@if.its.ac.id[3)]

ABSTRAK

Jumlah penelitian di dunia mengalami perkembangan yang pesat, setiap tahun berbagai peneliti dari penjuru dunia menghasilkan publikasi ilmiah seperti makalah, jurnal, buku dsb. Metode klasterisasi dapat digunakan untuk mengelompokkan kumpulan dokumen publikasi ilmiah ke dalam suatu kelompok tertentu berdasarkan relevansi antar topik. Klasterisasi pada dokumen memiliki karakteristik yang berbeda karena tingkat kemiripan antar dokumen dipengaruhi oleh kata-kata pembentuknya. Beberapa metode klasterisasi kurang memperhatikan nilai semantik dari kata. Sehingga klaster yang terbentuk kurang merepresentasikan isi topik dokumen. Klasterisasi dokumen teks masih memiliki kemungkinan adanya outlier karena pemilihan fitur teks yang tidak optimal. Oleh karena itu dibutuhkan pemrosesan data yang tepat serta metode yang mengoptimalkan hasil klaster. Penelitian ini mengusulkan metode klasterisasi dokumen menggunakan Weighted K-Means yang dipadukan dengan Maximum Common Subgraph. Weighted k-means digunakan untuk klasterisasi awal dokumen berdasarkan kata-kata yang diekstraksi. Pembentukan Weighted K-Means berdasarkan perhitungan Word2Vec dan TextRank dari kata-kata dalam dokumen. Maximum Common Subgraph merupakan tahap pembentukan graf yang digunakan dalam penggabungan klaster untuk menghasilkan klaster baru yang lebih optimal. Pembentukan graf dilakukan dengan perhitungan nilai Word2vec dan Co-occurrence dari klaster. Representasi topik dokumen tiap klaster dapat dihasilkan dari pemodelan topik Latent Dirichlet Allocation (LDA). Pengujian dilakukan dengan menggunakan dataset publikasi ilmiah dari Scopus. Hasil dari analisis Koherensi topik menunjukkan nilai koherensi usulan metode adalah 0,5375 pada dataset 1 yang bersifat heterogen dan 0,5642 pada dataset 2 yang bersifat homogen.

***Kata Kunci**: Co-occurence, Latent Dirichlet Allocation, Maximum Common Subgraph, Weighted k-means, Word2Vec*

ABSTRACT

The number of researches in the world has been increased, every year researchers from around the world produce scientific publications such as journals, proceeding, books, etc. Clustering methods can be used to cluster scientific publications based on relevance between topics. Clustering on document has different characteristics because the level of similarity between documents is influenced by the words. Some clustering methods less attention to the semantic value of the word. The cluster formed does not represent the topic of the document. Document clustering still has the possibility of outliers because the selection of text features is not optimal. Therefore, proper data processing and methods that optimize cluster results are needed. This research proposes a document clustering method using Weighted K-Means and Maximum Common Subgraph. Weighted k-means are used for initial clustering of documents based on extracted words. Weighted K-Means formed by Word2Vec and TextRank. Then maximum common subgraph is the graph formation stage used in combining clusters to produce a new optimal cluster. Graph is formed by Word2vec similarity and Co-occurrence of clusters. The topic cluster can be generated from the modeling of the Latent Dirichlet Allocation (LDA). Testing method use a dataset of scientific publication from Scopus. The results of the topic coherence analysis show the coherence value of the proposed method is 0.5375 in the heterogeneous dataset 1 and 0.5642 in the homogeneous dataset 2.

***Keywords**: Co-occurence, Latent Dirichlet Allocation, Maximum Common Subgraph, Weighted k-means, Word2Vec*

I. PENDAHULUAN

Jumlah penelitian di dunia mengalami perkembangan yang pesat setiap tahunnya. Peneliti baik dari kalangan akademisi maupun profesional dari penjuru dunia menghasilkan karya ilmiah penelitian yang dapat diwujudkan dalam bentuk jurnal ilmiah, makalah, konferensi, buku dll. Data karya ilmiah juga dapat dilihat di *World Wide Web* yang terimpan di dalam lembaga pengindeks seperti Google Scholar, Scopus, Thomsonreuters, DOAJ, dll. Kumpulan karya ilmiah tersebut merupakan sumber data yang berharga sehingga apabila dilakukan pengorganisasian yang lebih baik dan efisien dapat menghasilkan informasi yang berguna.

Metode klasterisasi dapat digunakan untuk mengelompokkan kumpulan dokumen karya ilmiah ke dalam suatu kelompok tertentu berdasarkan topik. Klasterisasi merupakan metode yang paling umum digunakan untuk mengelompokkan objek berdasarkan kemiripan dengan objek yang lain [1]. Terdapat beberapa pendekatan metode klasterisasi diantaranya *Partitioning Methods* [2] [3], *Hierarchical Methods* [4] [5], *Density Based Methods* [6] [7], *Grid Based Methods* [8], dan *Model Based Clustering Methods* [9]. Metode klasterisasi yang dipilih sangat bergantung pada permasalahan yang akan diselesaikan.

K-means merupakan metode klasterisasi yang menerapkan pendekatan *Partitioning Methods*. *K-means* dapat

diimplementasikan pada berbagai macam studi kasus, salah satunya adalah klasterisasi pada dokumen teks. Ide dasar klasterisasi dokumen teks mengelompokkan dokumen yang mempunyai kemiripan dengan dokumen lain. Untuk dapat menghitung tingkat kemiripan, dokumen tersebut harus diubah dalam bentuk vektor. Salah satu metode yang paling umum digunakan adalah dengan pendekatan *term frequency statistics*. Pendekatan ini digunakan pada beberapa penelitian seperti pada [10] menggunakan pendekatan *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk klasterisasi dokumen yang digunakan dengan *fuzzy K-means* and *hierarchical algorithm*. Pendekatan *term frequency* hanya memperhatikan nilai statistik dari kemunculan kata tanpa mengetahui pengaruh makna kata itu sendiri. Sehingga klaster yang terbentuk kurang merepresentasikan isi topik dokumen.

Pembentukan vektor dokumen dapat dilakukan dengan pendekatan yang berbasis *Natural Language Processing* (NLP). *Word2Vec* merupakan salah satu pendekatan *word embedding* yang berbasis NLP yang digunakan untuk memetakan kata kedalam vektor multidimensi. *Word2vec* dibuat tim peneliti yang dipimpin oleh Mikolov di google [11] dan sudah dikembangkan dalam beberapa penelitian seperti pada [12] [13]. Penggunaan *Word2Vec* dalam klasterisasi dapat menghasilkan klaster yang mempunyai kedekatan secara semantik. Hongzhi *et al* [4] menggunakan *Weighted k-means* dengan pendekatan *Word2Vec* untuk pembentukan bobot dari vektor kata, hasilnya dapat menurunkan jarak dari pusat klaster dalam kasus dokumen teks.

Disetiap dokumen maupun klaster memiliki topik yang terkandung didalamnya. Pada dasarnya topik tersebut merupakan bagian dari beberapa kata penyusun dokumen. Perkembangan analisis teks pada pemodelan topik sendiri pada dasanya bersumber pada matriks TF-IDF, selanjutnya dikembangkan lagi menjadi beberapa metode diantaranya *Latent Semantic Analysis* (LSA), *Probabilistic Latent Semantic Analysis* (PLSA), dan *Latent Dirichlet Allocation* (LDA). LDA [14] adalah salah satu algoritma pemodelan topik yang banyak digunakan di beberapa penelitian dan merupakan bentuk pengembangan dari beberapa algoritma pemodelan topik sebelumnya.

Pengelompokan dokumen berdasarkan topik masih memiliki kemungkinan adanya *outlier* karena pemilihan fitur teks yang tidak optimal. Penggunaan metode penggabungan klaster dapat menangani beberapa permasalahan terkait klasterisasi yang kurang optimal. Pada dasarnya metode penggabungan klaster bertujuan untuk mencari similaritas antar klaster yang terbentuk. *Maximun Common Subgraph* (MCS) dapat digunakan sebagai representasi klaster untuk pengukuran similaritas dengan menghitung relasi kemunculan bersama (*Co-occurrence*) sebagai bobot [15], namun hal tersebut kurang menghasilkan informasi semantik yang merepresentasikan kata. Penghitungan relasi dengan *Word2Vec* dapat digunakan, namun kurang memberikan konteks similaritas klaster yang tepat. Oleh karena itu pembentukan graf dapat dikombinasikan dengan pendekatan *Co-occurrence* dan *Word2Vec* dengan memanfaatkan kelebihan kedua pendekatan tersebut.

Penelitian ini mengusulkan metode klasterisasi dokumen menggunakan *Weighted K-Means* yang dipadukan dengan *Maximum Common Subgraph* sesuai dengan relevansi dari topik dokumen. *Weighted k-means* digunakan untuk klasterisasi awal dokumen berdasarkan kata-kata yang diekstraksi. Pembentukan *Weighted K-Means* berdasarkan perhitungan *Word2Vec* dan *TextRank* dari kata-kata dalam dokumen. *Maximum Common Subgraph* merupakan tahap pembentukan graf yang digunakan dalam penggabungan klaster untuk menghasilkan klaster baru yang lebih optimal. Pembentukan graf dilakukan dengan perhitungan nilai *Word2vec* dan *Co-occurrence* dari klaster. Representasi topik dokumen tiap klaster dapat dihasilkan dari pemodelan topik *Latent Dirichlet Allocation* (LDA) [16]. Pengujian dalam makalah ini menggunakan perhitungan *Coherence Measure* [17]. Pengujian dilakukan untuk menghitung nilai koherensi dokumen yang merepresentasikan keterkaitan antar topik klaster.

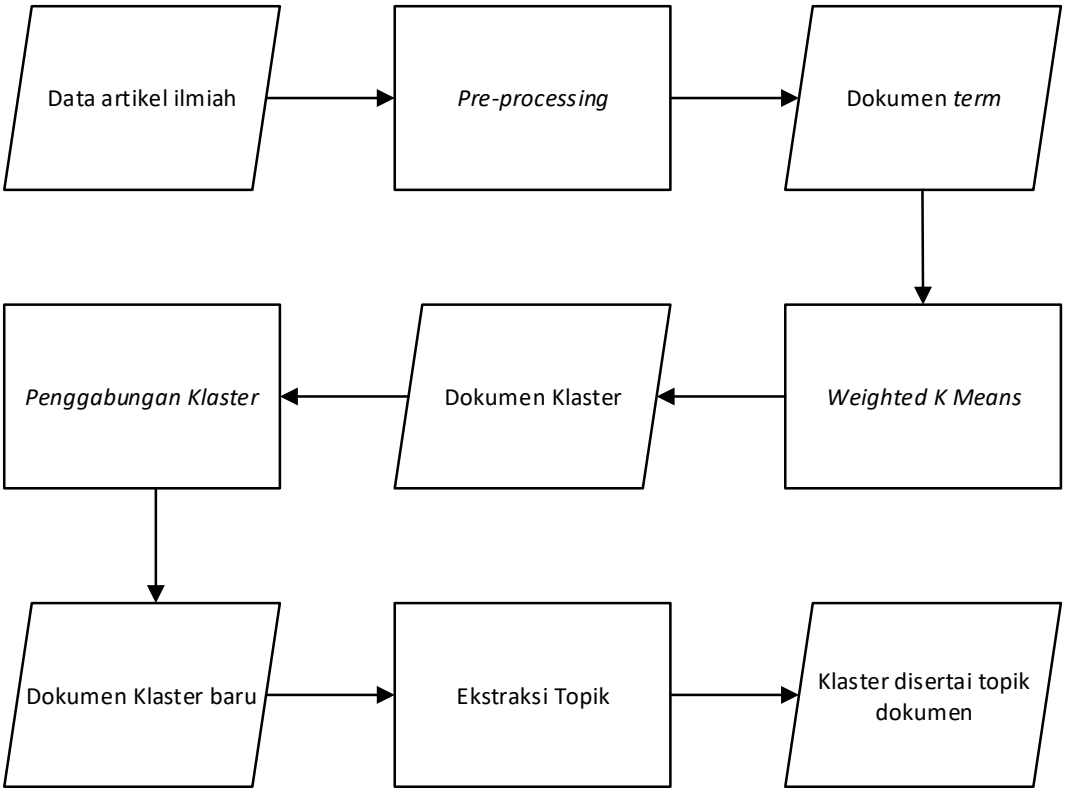
II. METODOLOGI PENELITIAN

Penelitian ini bertujuan mengklasterisasi dokumen berdasarkan topik dari dokumen tersebut. Terdapat dua proses utama yaitu klasterisasi yaitu menggunakan *Weighted K-means* dan penggabungan klaster berdasarkan *Maximum Common Subgraph*. Diagram dari metode yang digunakan dapat dilihat pada Gambar 1.

A. Pre-Processing

Tahap *pre-processing* digunakan untuk memproses dataset untuk menjadi data yang lebih bersih yang akan digunakan pada tahap klasterisasi. Tahap *pre-processing* terdiri dari beberapa proses sebagai berikut.

1. Konversi ke huruf kecil
   Data dari dokumen teks akan dikonversikan ke huruf kecil. Hal ini bertujuan untuk menyelaraskan *string* ke dalam bentuk standar.
2. Penghilangan *tag html*
   Beberapa data publikasi yang tersimpan di database masih memiliki *tag html* sehingga harus dihapuskan agar tidak menjadi *noise*.
3. Penghilangan tanda baca
   Pada proses ini sistem akan menghilangkan tanda baca yang biasa digunakan dalam kalimat agar tidak dinyatakan sebagai kata.



Gambar 1. Diagram dari metode yang digunakan

4. Tokenisasi
   Data yang berupa kalimat, paragraf atau dokumen akan dibagi menjadi token atau kumpulan kata. Token didapat dari pemisahan kata berdasarkan karakter spasi.
5. Penghilangan *stopwords*
   Stopwords adalah kata yang umum digunakan dengan jumlah besar dan dianggap tidak memiliki makna tersendiri. Dalam penelitian ini kata yang termasuk *stopword* akan didasarkan pada kumpulan stopword dalam bahasa inggris.

Hasil keluaran dari tahap *pre-processing* berupa dokumen *term*, yaitu kumpulan *term* atau kata-kata yang terdapat dalam satu dokumen yang akan digunakan pada tahap klasterisasi.

B. Klasterisasi menggunakan Weighted K-Means

Pada tahap ini dokumen *term* akan dikelompokkan pada satu klaster berdasarkan kemiripan kata. Setiap dokumen D terdiri dari beberapa kata $w_1, w_2, \dots, w_{i-1}, w_i$. Masing masing kata tersebut dapat dinyatakan sebagai vektor kata $V_1, V_2, \dots, V_{i-1}, V_i$. Perhitungan nilai vektor kata dilakukan dengan menggunakan algoritma *Word2Vec*.

Selain itu dalam setiap dokumen terdapat kata yang dapat dianggap sebagai kata kunci. Nilai vektor dari kata kunci dapat dinyatakan sebagai V_T. Perhitungan nilai kata kunci dapat menggunakan algoritma *keyword extraction* seperti *TextRank* [18]. *Keyword* atau kata kunci merupakan kata yang menjadi pusat perhatian dari dokumen. Nilai kata kunci didapat dari perhitungan algoritma *TextRank* berdasarkan probabilitas hubungan suatu kata dengan kata lain. Nilai *TextRank* dari kata tertinggi itu yang akan dijadikan kata kunci suatu dokumen.

Weighted K-Means memiliki nilai *weight* atau bobot dalam setiap kata. Bobot tersebut didapat dari perhitungan relevansi antara kata kunci V_T dan kata V_i. Perhitungan bobot dapat dicari dengan menggunakan persamaan *Euclidean Distance* yang dinotasikan pada persamaan (1).

$$C_i = 1 - \frac{\left|\left|V_i - V_T\right|\right|^2 / n}{\left|\left|V_T\right|\right|^2 / n} = 1 - \frac{||V_i - V_T||^2}{||V_T||^2} \quad (1)$$

n adalah dimensi dari vektor, ketika $V_i = V_T, C_i = 1$, maka *weight* yang relevan adalah yang terbesar, ketika $V_i = (0, \dots, 0), C_i = 0$, maka *weight* adalah terkecil. Oleh karena itu, *weight* yang relevan dapat mengekspresikan relativitas antara kata kunci dan topik tertentu.

Untuk setiap dokumen D akan mempunyai V_D yang merupakan vektor multidimensi untuk mewakili dokumen yang dinotasikan pada persamaan (2).

$$V_D = (C_1 V_1, C_2 V_2, \dots, C_{i-1} V_{i-1}, C_1 V_1) \quad (2)$$

Secara keseluruhan, persamaan tersebut meningkatkan kapasitas mengekspresikan topik yang akan membawa pengaruh yang baik untuk klasterisasi topik. Secara umum langkah-langkah agoritma *Weighted K-means* adalah sebagai berikut:

1. Hitung *weight* yang relevan disetiap kata dengan persamaan (1).
2. Dapatkan nilai vektor dokumen berdasarkan persamaan (2).
3. Normalisasi nilai vektor dokumen agar setiap dokumen memiliki panjang vektor yang sama.
4. Pilih objek *K* secara acak dari kumpulan data dan setiap objek mewakili pusat klaster awal atau *mean* dari suatu topik.
5. Hitung jarak antara dokumen dan setiap pusat klaster dengan menggunakan *cosine distance*.
6. Hitung ulang rata-rata setiap klaster sebagai pusat klaster baru.
7. Jika semua pusat klaster tidak berubah, fungsi objektif telah *converged* maka algoritma telah berakhir, jika tidak modifikasi pusat klaster lalu kemudian ulangi langkah 5 dan langkah 6.

C. Pembentukan Graf Klaster

Hasil klasterisasi harus diubah ke dalam bentuk graf. Gambar 2 menunjukkan struktur graf pada umumnya yang terdiri dari *vertex* dan *edge*. Graf Klaster G_i terdiri dari *vertex* atau *node* v_i yang mewakili kumpulan kata dalam klaster, serta *edge* e_i yang mewakili relasi antar kata tersebut. Untuk mendapatkan *vertex* dari klaster pelu dilakukan *unigram extraction* yaitu ekstraksi kata tunggal pembentuk klaster dengan *minimum support* tertentu. Kata yang diekstrasi adalah kata yang sudah dilakukan proses *stemming*. *Stemming* adalah proses pemetaan dan penguraian bentuk dari suatu kata menjadi bentuk kata dasarnya. Algoritma *stemming* yang digunakan adalah *PorterStemmer*. Sebuah klaster K_i terdiri dari kumpulan dokumen *term* D_t. Nilai dari *unigram extraction* adalah sekumpulan kata w_t dimana $w_t \in D_t | w_t > \min\_support$.

Selanjutnya perhitungan *edge* dengan mencari relasi antar kata. Relasi antar kata dapat dicari dengan menghitung similaritas antar kata pada perhitungan *Word2Vec*. Sebuah *node* v_i dan *node* v_j memiliki nilai *Word2Vec similarity* $W_{sim(i,j)}$. Nilai *similarity* akan berada di rentang angka -1 sampai 1. Nilai negatif memiliki arti bahwa kata tersebut memiliki arti berlawanan dengan kata yang lain. Kedua *node* tersebut akan memiliki relasi apabila memiliki nilai diatas *threshold* yang ditentukan.

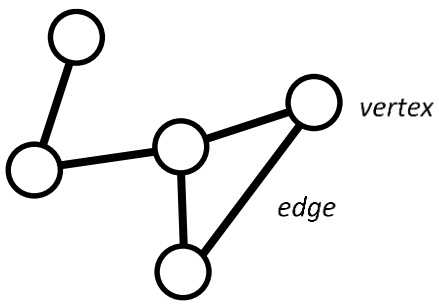
Alternatif lain dapat menggunakan perhitungan *Co-occurrence* [19] yaitu perhitungan jumlah kemunculan kata bersama dengan nilai *window* atau rentang kata tertentu. Nilai *Co-occurrence* dari dua *node* dapat dinyatakan dengan *Word Co-occurrence Matrix*. Sebuah node v_i memiliki frekuensi kata f_i dan *N* adalah jumlah kata dalam teks. Nilai *Co-occurence* dari *node* v_i dapat dinyatakan dengan persamaan (3). Selanjutnya untuk menghitung nilai *co-occurence* kata dari *node* v_i dan *node* v_j dapat dinyatakan dengan persamaan (4) dengan $f_{i,j}$ adalah frekuensi kemunculan kata dari *node* v_i dan *node* v_j.

$$W_{co(i)} = \frac{f_i}{N} \tag{3}$$

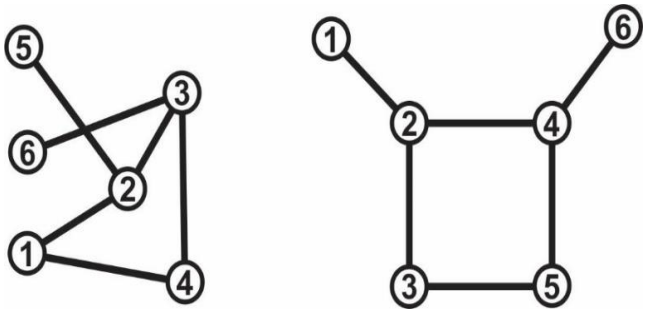
$$W_{co(i,j)} = \frac{f_{i,j}}{W_{co(i)} + W_{co(j)} - f_{i,j}} \tag{4}$$

Selanjutkan akan dikombinasikan dua pendekatan perhitungan similaritas antara node v_i dan v_j menggunakan persamaan (5) dengan koefisien α yang memiliki rentang nilai 0 sampai 1.

$$W_{edge(i,j)} = \alpha \times W_{sim(i,j)} + (1 - \alpha) \times W_{co(i,j)} \tag{5}$$



Gambar 2. Struktur Graf



Gambar 3. Contoh Graf *Isomorphic*

Graf yang klaster disimpan kedalam bentuk *graph distance metric*. Pembentukan graf dilakukan pada semua klaster. Selanjutnya akan dilakukan perhitungan similaritas antar graf pada tahap penggabungan klaster.

D. Penggabungan Klaster menggunakan Maximum Common Subgraph (MCS)

Setiap klaster akan memiliki model graf sesuai dengan relasi antara kata. Terdapat kemungkinan adanya kemiripan struktur dalam graf. Oleh karena itu perlu dilakukan *similarity check* dengan menggunakan konsep *Maximum Common Subgraph* (MCS) [20]. Sebuah graf dinyatakan *isomorphic* jika terdapat korespondensi satu-satu ke pemetaan *node* antar kedua graf. MCS menghitung nilai *common subgraph* maksimal dari graf *isomorphic* yang terbentuk dari graf G_1 dan G_2. Contoh dari graf *isomorphic* dapat dilihat pada Gambar 3.

Masukkan dari MCS adalah graf representasi teks G_1 dan G_2. Hasil dari MCS adalah *sub-graph optimal* G'. Alur metode MCS dapat dijelaskan sebagai berikut:

1. Cari *node* yang sama antara G_1 dan G_2, tambahkan ke G'.
2. Ambil 2 *node* yang berbeda pada G'. Jika kedua *node* bersebelahan pada G_1 dan G_2, maka edge yang menghubungkan kedua *node* tersebut ditambahkan ke G'. Bobot *edge* terkecil antara *edge* G_1 dan G_2 menjadi bobot *edge* di G'.
3. Ulangi langkah 2 sampai tidak ada lagi *edge* yang bisa ditambahkan.

Langkah selanjutnya adalah mencari similaritas antar graf $S(G_1, G_2)$ yang dirumuskan oleh persamaan (6).

$$S(G_1, G_2) = \beta \frac{N_{G\prime}}{N_{\max(G_1,G_2)}} + (1-\beta) \frac{E_{G\prime}}{E_{\max(G_1,G_2)}} \quad (6)$$

$N_{G\prime}$ merupakan *node* pada G', $N_{\max(G_1,G_2)}$ merupakan nilai maksimal total *node* pada G_1 dan G_2, $E_{G\prime}$ merupakan jumlah *edge* pada G', dan $E_{\max(G_1,G_2)}$ merupakan nilai maksimal total *edge* pada G_1 dan G_2. Koefisien β merupakan nilai antara 0 dan 1 yang mewakili tingkat kepentingan *node* terhadap *edge* pada *subgraph*. Jika dua buah graf klaster G_1 dan G_2 memiliki jarak yang lebih rendah daripada *threshold* t, maka graf klaster G_1 dan G_2 dapat digabung menjadi satu klaster. Dengan demikian keluaran dari tahap ini adalah kumpulan klaster dokumen yang baru berdasarkan perhitungan bobot MCS pada setiap elemen dari kata.

E. Ekstraksi Topik Klaster

Pada tahap ini adalah mengekstrak topik yang ada untuk menggabarkan isi topik dari dokumen klaster. Implementasi dari tahap ini menggunakan pemodelan *Latent Dirichlet Allocation* (LDA) [14]. LDA merupakan bagian dari *Bayesian Hierarchical Models* yang merupakan kumpulan data teks yang dimodelkan sebagai model campuran dari berbagai topik. Hasil keluaran dari tahap ini adalah kumpulan *term* yang merupakan topik dari setiap klaster dengan nilai bobot berdasarkan perhitungan dari LDA.

III. UJI COBA DAN PEMBAHASAN

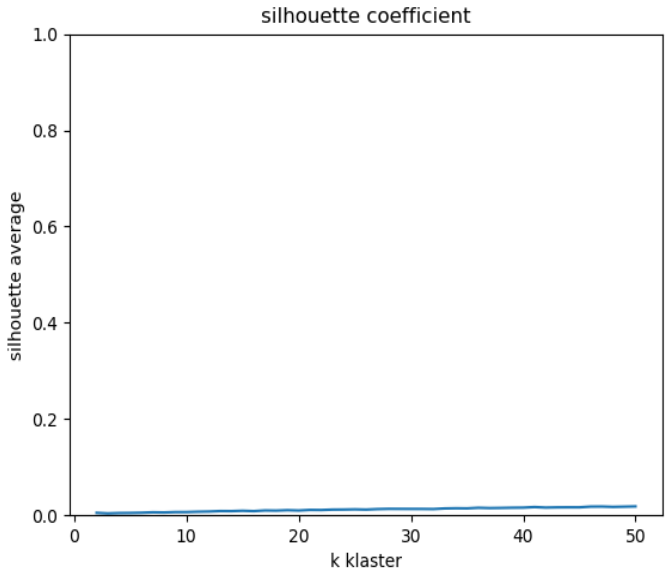
Usulan metode dievaluasi menggunakan perhitungan koherensi topik, yaitu mengukur interpretabilitas topik yang dihasilkan. Suatu nilai klaster koheren secara topikal jika antara topik dokumen di dalam klaster memiliki keterkaitan kontekstual yang tinggi. Perhitungan koherensi topik didasarkan pada penelitian dari Roder *et al* [17]. Proses uji coba dilakukan dengan menyiapkan dataset dan skenario dari uji coba.

Dataset yang digunakan dalam penelitian ini berasal dari publikasi ilmiah yang terindeks Scopus dengan *author* dari dosen Institut Teknologi Sepuluh Nopember. Dataset dapat diperoleh dengan menggunakan API Scopus yang kemudian diolah untuk disimpan ke dalam basis data SQL Server. Dokumentasi dari API Scopus dapat dilihat pada alamat *https://dev.elsevier.com/scopus.html*. Terdapat banyak *field* atau kolom dari data yang didapat, untuk proses klasterisasi hanya akan menggunakan data *title* dan *abstract* dari dokumen artikel ilmiah.

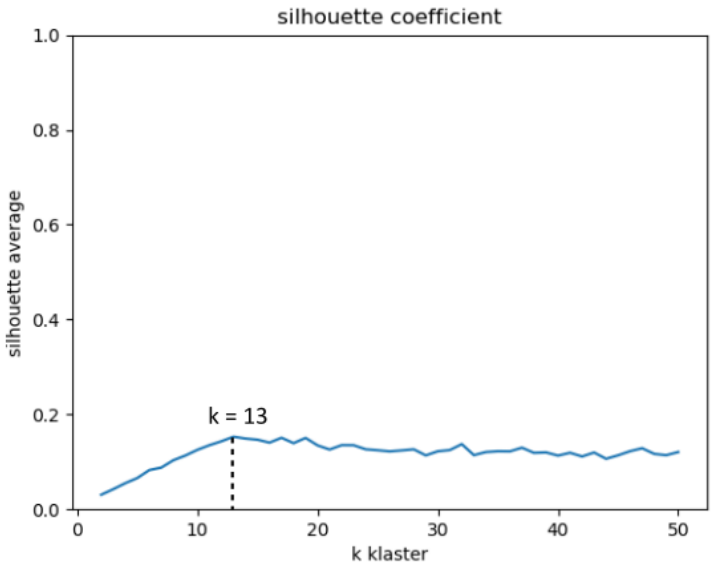
A. Analisis Dataset

Analisis dataset bertujuan untuk menentukan jumlah *k* klaster dan memilih beberapa jenis data yang akan digunakan dalam uji coba. *Silhouette Coefficient* dapat digunakan untuk menganalisis data dan menentukan *k* optimal [21]. Nilai *Silhouette* berasal dari perhitungan kemiripan suatu objek dengan kelompoknya sendiri atau *cohesion*, kemudian dibandingkan dengan kelompok lain atau *separation*. Data yang didapat dari Scopus berjumlah 5.242 data publikasi, data ini selanjutnya disebut dataset keseluruhan. Perhitungan rata-rata *Silhouette Coefficient* dilakukan pada dataset keseluruhan dengan menggunakan nilai *k* antara 2 sampai 50. Hasil dari perhitungan dapat dilihat pada Gambar 4.

Pada Gambar 4 masih belum terlihat nilai *Silhouette Coefficient* yang signifikan diantara klaster 2 sampai 50. Hal tersebut dikarenakan banyaknya *outlier* pada dataset serta adanya *curse of dimensionality* karena besarnya dimensi data yang digunakan. Oleh karena itu data tersebut perlu disaring berdasarkan nilai *Silhouette Coefficient* disetiap dataset. Dari dataset keseluruhan, diambil beberapa data yang mempunyai nilai *Silhouette* diatas 0,1. Hasinya adalah data yang didapat berjumlah 265 data publikasi, data ini kemudian akan disebut sebagai dataset 1. Selanjutnya dihitung kembali rata-rata *Silhouette Coefficient* disetiap klaster untuk mencari nilai *k* optimal pada dataset 1. Perhitungan dilakukan pada *k* antara 2 sampai 50, hasilnya nilai *k* optimal yang didapat adalah 13. Grafik nilai rata-rata *Silhouette Coefficient* dari dataset 1 dapat dilihat pada Gambar 5.



Gambar 4. Hasil perhitungan rata-rata *Silhouette Coefficient* pada dataset keseluruhan



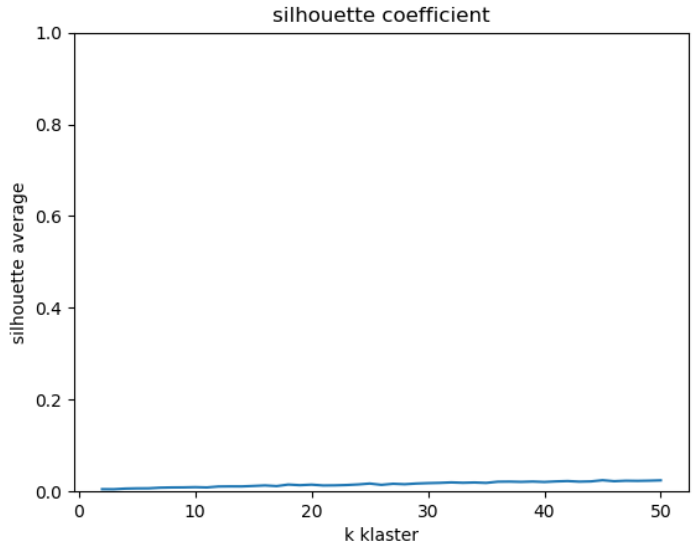
Gambar 5. Hasil perhitungan rata-rata Silhouette Coefficient *pada dataset 1*

Dataset 1 berisikan data dari semua bidang penelitian di dalam Scopus, data publikasi yang didapat cenderung beragam atau dapat disebut sebagai data heterogen. Oleh karena itu akan dibandingkan dengan dataset yang mewakili satu bidang untuk mendapatkan dataset homogen. Pada data Scopus terdapat pilihan bidang, pada pengujian ini dataset yang dipilih adalah dataset yang berasal dari bidang *Computer Science* dari data publikasi Scopus yang berjumlah 1.804 data. Selanjutnya dilakukan perhitungan *Silhouette Coefficient* pada dataset *Computer Science* dengan nilai klaster antara 2 sampai 50. Hasil perhitungan dapat dilihat pada Gambar 6.

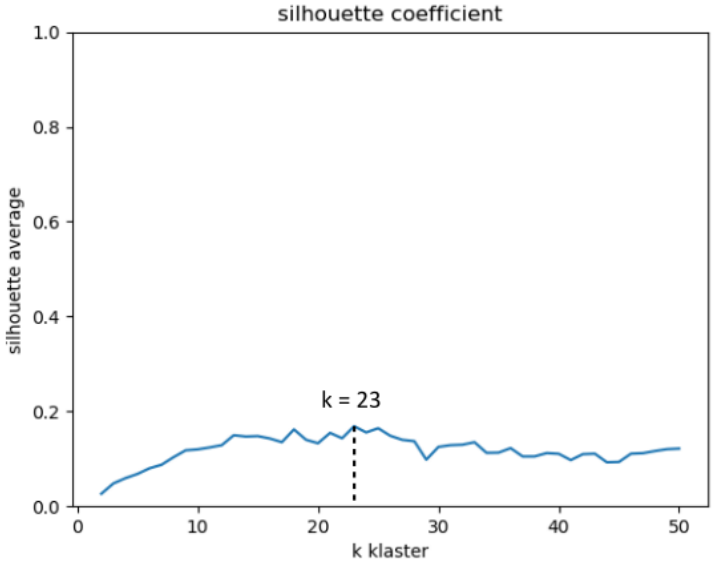
Hasil perhitungan dari dataset *Computer Science* masih belum terlihat nilai *Silhouette Coefficient* yang signifikan. Oleh karena itu dataset akan disaring dengan mencari nilai *Silhouette Coefficient* disetiap data, dengan *threshold* adalah 0,1. Hasil yang didapat adalah dataset sejumlah 155 data publikasi, selanjutnya data tersebut akan disebut sebagai dataset 2. Setelah itu pada dataset 2 dicari nilai *k* optimal, hasilnya dapat dilihat pada Gambar 7 dengan nilai *k* yang didapat adalah 23.

B. Skenario Pengujian

Terdapat dua skenario uji coba yang akan dilakukan, skenario pertama yaitu membandingkan usulan metode dengan metode *K-means* untuk mengetahui dampak dari penggunakan *weight* pada usulan metode. Pengujian akan menggunakan dataset 1 dan dataset 2 disetiap metode yang digunakan. Skenario selanjutnya adalah untuk



Gambar 6 Hasil perhitungan rata-rata *Silhouette Coefficient* pada dataset *Computer Science*



Gambar 7 Hasil perhitungan rata-rata *Silhouette Coefficient* pada dataset 2

TABEL I
SKENARIO PENGUJIAN

Skenario	Dataset	Metode
1	1 dan 2	Usulan Metode (*Weighted K-Means*)
	1 dan 2	*K-Means*
2	1	Usulan Metode (*Word2vec* dan *Co-occurence*)
	1	*Word2Vec*
	1	*Co-occurence*

TABEL II
HASIL PENGUJIAN SKENARIO 1

Percobaan	Dataset	K-Means		Usulan Metode	
		Jumlah klaster	Rata-rata Koherensi Topik	Jumlah Klaster	Rata-rata Koherensi Topik
1	1	13	0,4994	11	**0,5375**
2	2	23	0,4558	14	**0,5642**

TABEL III
HASIL PENGUJIAN SKENARIO 2

Percobaan	Dataset	Metode	Jumlah Klaster	Rata-rata Koherensi Topik
1	1	Usulan Metode (Word2vec dan Co-occurence)	11	**0,5254**
2	1	Word2Vec	11	0,5157
3	1	Co-occurence	11	0,5414

mengetahui pengaruh penggunaan metode pada pembentukan relasi graf. Metode *Word2Vec* dan *Co-occurrence* akan dibandingkan dengan usulan metode yaitu penggabungan antara *Word2Vec* dengan *Co-occurrence*. Skenario dari pengujian dapat dilihat pada Tabel I.

C. Hasil Pengujian Skenario 1

Pada usulan metode penggunaan algoritma *Weighted K-means* menambahkan bobot kata berdasarkan perhitungan vektor dari masing-masing kata dengan *keyword* yang didapat. Hal tersebut akan berdampak pada hasil klasterisasi yang dihasilkan. Pada skenario 1 terdapat dua jenis dataset yang digunakan yaitu dataset 1 yang berisikan 265 data publikasi ilmiah serta dataset 2 sejumlah 155 data publikasi ilmiah. Selanjutnya, perhitungan koherensi dilakukan pada usulan metode dibandingkan dengan *K-means* dengan menggunakan dataset 1 dan dataset 2.

Tabel II menunjukkan hasil pengujian dari skenario 1, pada tabel tersebut dapat terlihat perbedaan jumlah klaster yang digunakan. Perbedaan jumlah klaster terjadi karena pada usulan metode terdapat proses penggabungan klaster dimana jumlah klaster akan lebih efisien dibandingkan dengan kalster awal. Pada saat percobaan 1 menggunakan dataset 1 jumlah klaster yang digunakan pada *K-Means* adalah 13 sedangkan pada usulan metode adalah 11. Pada *K-means* menghasilkan nilai rata-rata koherensi topik sebesar 0,4994, sedangkan usulan metode menghasilkan nilai sebesar 0,5254. Kemudian pada percobaan 2 dengan menggunakan dataset 2, jumlah klaster *K-Means* adalah 23 dan jumlah klaster usulan metode adalah 14. Nilai rata-rata koherensi *K-means* sebesar 0,4558 sedangkan pada usulan metode sebesar 0,5642.

Hasil pengujian pada skenario 1 menunjukkan adanya peningkatan yang terjadi pada penggunaan usulan metode dibandingkan dengan *K-means*. Peningkatan tersebut cukup signifikan, hal tersebut dapat dilihat bahwa pada metode *K-means* nilai rata-rata koherensi yang didapat berada dibawah angka 0,5. Sedangkan pada usulan metode nilai rata-rata koherensi berada diatas 0,5 dengan selisih 0,0381 pada dataset 1 dan 0,1084 pada dataset 2. Perhitungan koherensi dilakukan berdasarkan ekstraksi topik yang menggunakan algoritma LDA. Topik yang didapat merupakan perhitungan semantik dari setiap kata di dalam dokumen klaster. Pada usulan metode, nilai semantik dari dokumen sangat diperhatikan. Sedangkan pada *K-means*, perhitungan relasi antar dokumen hanya didasarkan pendekatan statistik. Sehingga hasil klaster pada usulan metode akan memiliki nilai interpretabilitas topik yang lebih tinggi dibandingkan dengan *K-means*.

D. Hasil Pengujian Skenario 2

Pembentukan relasi graf untuk perhitungan *Maximum Common Subgraph* pada proses penggabungan klaster dapat dilakukan dengan beberapa metode, yaitu menggunakan *Word2Vec* atau *Co-occurrence*. Usulan metode menggunakan penggabungan antara *Word2Vec* dengan *Co-occurrence*. Skenario 2 menguji dampak dari penggabungan metode tersebut dibandingkan dengan metode lain. Dataset yang digunakan adalah dataset 1. Hasil dari pengujian dapat dilihat pada Tabel III.

Terdapat 3 percobaan yang dilakukan pada skenario 2. Percobaan 1 mengggunakan dataset 1, pembentukan graf menggunakan penggabungan *Word2Vec* dan *Co-occurrence* dengan jumlah klaster adalah 11. Percobaan 2 mengggunakan dataset 1, pembentukan graf menggunakan similaritas *Word2Vec* dengan jumlah klaster adalah 11. Percobaan 3 mengggunakan dataset 1, pembentukan graf menggunakan perhitungan *Co-occurrence* dengan jumlah klaster adalah 11. Pada percobaan 1, nilai rata-rata koherensi yang didapat lebih besar dari pada percobaan 2 dan percobaan 3. Hal tersebut dikarenakan pada percobaan 1 menggunakan usulan metode dengan menggabungkan kelebihan dari *Word2Vec* dan *Co-occurrence* secara bersama-sama. *Word2Vec* dapat menggabarkan relasi kata berdasarkan kemiripan secara semantik. Sedangkan *Co-occurrence* dapat menggabarkan kedekatan kata berdasarkan kemunculan bersama. Peningkatan yang terjadi pada penggunaan usulan metode tidak begitu signifikan. Ketiga percobaan menunjukkan nilai rata-rata koherensi topik yang berada diatas 0,5 dengan selisih kurang dari 0,2. Hal tersebut dikarenakan pada pembentukan relasi kata graf sangat dipengaruhi pada jumlah data yang digunakan. Sehingga semakin besar jumlah data yang digunakan akan menyebabkan jumlah *vertex* dan *edge* bertambah, sehingga penggunaan variasi metode akan menunjukkan hasil yang berbeda. Namun pada penggunaan data yang besar membutuhkan jumlah memori yang besar juga.

IV. KESIMPULAN

Dalam makalah ini metode *Weighted K-means* dan *Maximum Common Subgraph* untuk klasterisasi artikel ilmiah telah diusulkan. *Weighted K-means* digunakan untuk menambah bobot kata pada proses klasterisasi dan menggunakan perhitungan *Word2Vec* dan *TextRank* untuk menentukan nilai vektor dari kata. Sedangkan *Maximum Common Subgraph* digunakan untuk membentuk graf yang digunakan pada penggabungan klaster.

Pengujian dilakukan dengan menghitung nilai koherensi topik yang dibandingkan antara usulan metode dengan beberapa metode melalui beberapa skenario percobaan. Pada skenario 1 klasterisasi *Weighted K-Means* dari usulan metode dibandingkan dengan algoritma *K-Means*. Hasil nilai rata-rata koherensi topik dari usulan metode memiliki nilai lebih baik daripada *K-means*, baik pada dataset 1 yang bersifat heterogen maupun dataset 2 yang bersifat homogen. Pada skenario 2 pengujian dilakukan untuk mengetahui dampak penggunaan algoritma relasi pembentukan graf. Usulan metode menggunakan penggabungan *Word2Vec* dan *Co-occurrence* menggunakan persamaan (5) dibandingkan dengan algoritma *Word2Vec* dan algoritma *Co-occurrence*. Hasilnya menunjukkan bahwa penggabungan metode memiliki nilai rata-rata koherensi yang lebih baik.

Secara umum penggunaan *Weighted K-Means* dan penggabungan Klaster *Maximum Common Subgraph* dari usulan metode dapat meningkatkan nilai koherensi topik pada klasterisasi dokumen artikel ilmiah. Akan tetapi penggunaan metode penggabungan klaster menggunakan graf dapat menyebabkan penggunaan memori yang tinggi tergantung pada besarnya graf yang digunakan.

V. SARAN

Untuk penelitian selajutnya pada proses penggabungan klaster dapat menggunakan pendekatan statistik atau pendekatan graf lain yang lebih ramah dalam penggunaan memori tanpa mengurangi nilai dari klaster.

DAFTAR PUSTAKA

[1] P.-N. Tan, M. Steinbach, A. Karpatne and V. Kumar, Introduction to Data Mining, Pearson Education Inc, 2006.

[2] N. Heidari, Z. Moslehi, A. Mirzaei and M. Safayani, "Bayesian distance metric learning for discriminative fuzzy c-means clustering," *Neurocomputing*, vol. 319, pp. 21-33, 2018.

[3] D. Purwitasari, C. Fatichah, I. Arieshanti and N. Hayatin, "K-medoids algorithm on Indonesian Twitter feeds for clustering trending issue as important terms in news summarization," in *2015 International Conference on Information & Communication Technology and Systems (ICTS)*, Surabaya, 2015.

[4] W. Wei, J. Liang, X. Guo, P. Song and Y. Sun, "Hierarchical division clustering framework for categorical data," *Neurocomputing*, Vols. 118-134, p. 341, 2019.

[5] B. Lorbeer, A. Kosareva, B. Deva, D. Softić, P. Ruppel and A. Küpper, "Variations on the Clustering Algorithm BIRCH," *Big Data Research*, vol. 11, pp. 44-53, 2018.

[6] M. Kumar and M. Reddy, "A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method," *Pattern Recognition*, vol. 58, pp. 39-48, 2016.

[7] L. Carlos and A. Rodrigo, "Density-based clustering methods for unsupervised separation of partial discharge sources," *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 224-230, 2019.

[8] A. Artu and C. Özdoğan, "Parallel WaveCluster: A linear scaling parallel clustering algorithm implementation with application to very large datasets," *Journal of Parallel and Distributed Computing*, vol. 71, no. 7, pp. 955-962, 2011.

[9] N. Mulani, A. Pawar, P. Mulay and A. Dani, "Variant of COBWEB Clustering for Privacy Preservation in Cloud DB Querying," *Procedia Computer Science*, vol. 50, pp. 363-368, 2015.

[10] P. Bafna, D. Pramod and A. Vaidya, "Document Clustering: TF-IDF approach," in *International Conference on Electrical, Electronics, and Optimization Techniques*, Chennai, 2016.

[11] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, Arizona, 2013.

[12] L. Zhang, J. Li and C. Wang, "Automatic synonym extraction using Word2Vec and spectral clustering," in *2017 36th Chinese Control Conference (CCC)*, Dalian, 2017.

[13] V. Vargas-Calderón and J. E.Camargo, "Characterization of citizens using word2vec and latent topic analysis in a large set of tweets," *cities*, vol. 92, pp. 187-196, 2019.

[14] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research 3*, pp. 993-1022, 2003.

[15] A. Nurilham, D. Purwitasari and C. Fatichah, "Ekstraksi Frasa Kunci pada Penggabungan Klaster berdasarkan Maximum-Common-Subgraph," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 7, no. 3, 2018.

[16] K. Bastani, H. Namavari and J. Shaffer, "Latent Dirichlet allocation (LDA) for topic modeling of the CFPB consumer complaints," *Expert Systems with Applications*, vol. 127, pp. 256-271, 2019.

[17] M. Roder, A. Both and A. Hinneburg, "Exploring the Space of Topic Coherence Measures," in *WSDM '15 Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, Shanghai, 2015.

[18] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, 2004.

[19] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, p. 203–208, 1996.

[20] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph," *Pattern Recognition Letters*, vol. 19, no. 3-4, pp. 255-259, 1998.

[21] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65, 1987.