Pengelompokan Artikel Berbahasa Indonesia Berdasarkan Struktur Laten Menggunakan Pendekatan *Self Organizing Map*

Akhmad Zaini[1], M. Aziz Muslim[2], Wijono[3]

***Abstract*— Document grouping is a necessity among a large number of articles published on internet. Several attempts have been done to improve this grouping process, while majority of the efforts are based on word appearance. In order to improve its quality, the grouping of documents need to be based on topic similarity between documents, instead of the frequency of word appearance. The topic similarity could be known from its latency, since the similarity of the word interpretation are often used in the same context. In the unsupervised learning process, SOM is often used, in which this approach simplifies the mapping of multi-dimension data. This research result shows that implementation of the latent structure decreases characteristic dimension by 32% of the word appearance, hence makes this approach more time efficient than others. The latent structure, however, when implemented on SOM Algorithm, is capable to obtain good quality result compared to word appearance frequency approach. It is then proven by 5% precision improvement, recall improvement of 3%, and another 4% from F-measure. While the achievement is not quite significant, the quality improvement is able to put the dominance of grouping process, compared to the original classification defined by the content provider.**

***Intisari*— Pengelompokan dokumen merupakan salah satu kebutuhan di tengah-tengah banyaknya artikel yang terbit di internet. Telah banyak upaya yang dilakukan oleh para ahli dalam rangka pengelompokan dokumen ini, yang umumnya dilakukan berdasarkan kemunculan kata saja. Dalam rangka meningkatkan kualitas pengelompokan, diperlukan dasar yang tidak hanya bergantung pada frekuensi kemunculan kata saja, tetapi juga dari kedekatan topik antar dokumen. Kedekatan topik dapat diketahui melalui latensi, mengingat kata yang memiliki kemiripan makna sering kali digunakan pada konteks yang sama. Dalam hal pembelajaran tanpa supervisi, salah satu pendekatan yang sering digunakan adalah *Self Organizing Map* (SOM). Pendekatan ini biasanya digunakan untuk memetakan data berdimensi banyak menjadi lebih sederhana. Hasil menunjukkan, penggunaan struktur laten dapat mengurangi dimensi ciri sebesar 32% dari dimensi ciri kemunculan kata, sehingga berdampak pada efisiensi waktu. Meskipun demikian, struktur laten, ketika diterapkan pada algoritme SOM, mampu menghasilkan kualitas yang tidak kalah baiknya jika dibandingkan dengan menggunakan frekuensi kemunculan kata. Hal ini terbukti dengan adanya peningkatan presisi sebesar 5%, *recall* sebesar 3%, dan *F-Measure* sebesar 4%. Meskipun peningkatan yang ada tidak terlalu signifikan, tetapi jika dilakukan perbandingan dengan kategori asli dari penyedia konten, peningkatan kualitas tersebut mampu meletakkan dominasi pengelompokan sesuai dengan kategori aslinya.**

***Kata Kunci*— Pengelompokan artikel, *clustering*, latensi, *Self Organizing Map*, *unsupervised learning*.**

I. PENDAHULUAN

World Wide Web merupakan gudang artikel yang berisi jutaan artikel. Banyaknya artikel di web melahirkan kebutuhan baru bagi pengguna, seperti pengelompokan dan penelusuran artikel. Beberapa upaya telah dilakukan dalam rangka pengelompokan artikel, khususnya untuk artikel yang berbahasa Indonesia. Salah satu upaya yang pernah dilakukan adalah pengelompokan artikel kejadian (*event*) dengan menggunakan *single pass clustering*, dengan cara mengekstraksi ciri-ciri artikel berdasarkan kemunculan kata. Upaya tersebut menghasilkan rata-rata *recall* sebesar 76% dan presisi sebesar 87% [1]. Metode *Support Vector Machine* (SVM) juga pernah digunakan untuk mengelompokkan artikel berdasarkan kemunculan katanya dan diperoleh akurasi sebesar 91,67 % [2]. Kemudian, pendekatan *Self Organizing Map* (SOM) diterapkan pada percobaan lain untuk dapat mengelompokkan artikel berdasarkan kemunculan katanya juga [3].

Seluruh upaya di atas menggunakan kemunculan kata sebagai parameter ciri. Selain menggunakan kemunculan kata sebagai dasar pengelompokan artikel, ciri lain yang bisa digunakan adalah berdasarkan struktur urutan kata. *Suffix Tree Clustering* [4], *Document Index Graph* [5], dan ruang vektor *suffix tree* [6] merupakan upaya pengelompokan artikel yang menggunakan urutan kata sebagai dasar pengelompokan. Namun, ciri tersebut tidak dapat berdiri sendiri sebagai parameter. Ciri tersebut harus tetap dikombinasikan dengan kemunculan katanya. Pada percobaan lain juga telah dilakukan sebuah upaya untuk mengklasifikan sentimen suatu statemen berdasarkan frasa [7], tetapi hal ini belum memungkinkan untuk diterapkan pada dokumen yang mengandung banyak kalimat dan frasa di dalamnya.

Dari serangkaian upaya yang telah dilakukan, dapat disimpulkan bahwa kemunculan kata masih sangat dominan untuk digunakan sebagai ciri dalam pengelompokan dokumen. Meskipun demikian, kemunculan kata tidak selalu menjadi ukuran dalam menentukan relevansi artikel terkait dan penentuan hasil pencarian. Pada kasus penentuan artikel terkait, dua artikel dengan topik yang sama tidak selalu memiliki kesamaan kemunculan kata. Begitu juga pada kasus

[1] *Mahasiswa, Program Magister Universitas Brawijaya Malang, Jl. Veteran Malang, Jawa Timur, INDONESIA 65145 (tlp: 0341-551611; fax: 0341565420; e-mail: zaini1983@gmail.com)*

[2, 3] *Dosen, Teknik Elektro Universitas Brawijaya Malang, Jl. Veteran Malang, Jawa Timur, INDONESIA 65145 (tlp: 0341-551611; fax: 0341565420; e-mail: [2]muh_aziz@ub.ac.id, [3]wijono@ub.ac.id)*

pencarian, hasil pencarian yang relevan bagi pengguna terkadang sama sekali tidak mengandung kata kunci pencarian, tetapi memiliki kedekatan topik (hubungan tersembunyi) dengan kata kunci pencarian. Kesamaan maupun perbedaan konteks kalimat yang digunakan oleh kata kunci dengan kata lainnya menyebabkan hubungan tersebut muncul. Sebagai contoh, kata mobil sering digunakan pada konteks yang sama dengan kata kendaraan, atau mungkin sebaliknya. Kata bank terkadang digunakan pada beberapa konteks yang berbeda seperti bank soal dan bank instansi keuangan. Kesamaan maupun perbedaan penggunaan konteks, salah satunya dapat dilihat dari latensi.

Dari sini dapat disimpulkan bahwa latensi berpotensi untuk dijadikan sebagai dasar dalam pengelompokan dokumen. Setelah diketahui ciri berdasarkan latensi, hal yang tidak kalah pentingnya adalah proses pengelompokan itu sendiri. Melalui pengujian yang telah dilakukan, disimpulkan bahwa SOM mampu melakukan *clustering* dengan akurasi yang lebih baik dibanding *K-Means*, *Single-Linkage*, maupun DBSCAN [8]. Percobaan lain juga telah membuktikan bahwa SOM memiliki sensitivitas terhadap *noise* yang lebih rendah jika dibandingkan dengan *Hierarchical Clustering*, serta pendekatan ini juga sesuai untuk digunakan pada *dataset* berukuran besar [9]. Melalui pengelompokan berdasarkan latensi diharapkan kebutuhan-kebutuhan terkait penelusuran dan penentuan artikel terkait dapat terpenuhi.

II. Pengelompokan Artikel Berdasarkan Struktur Laten

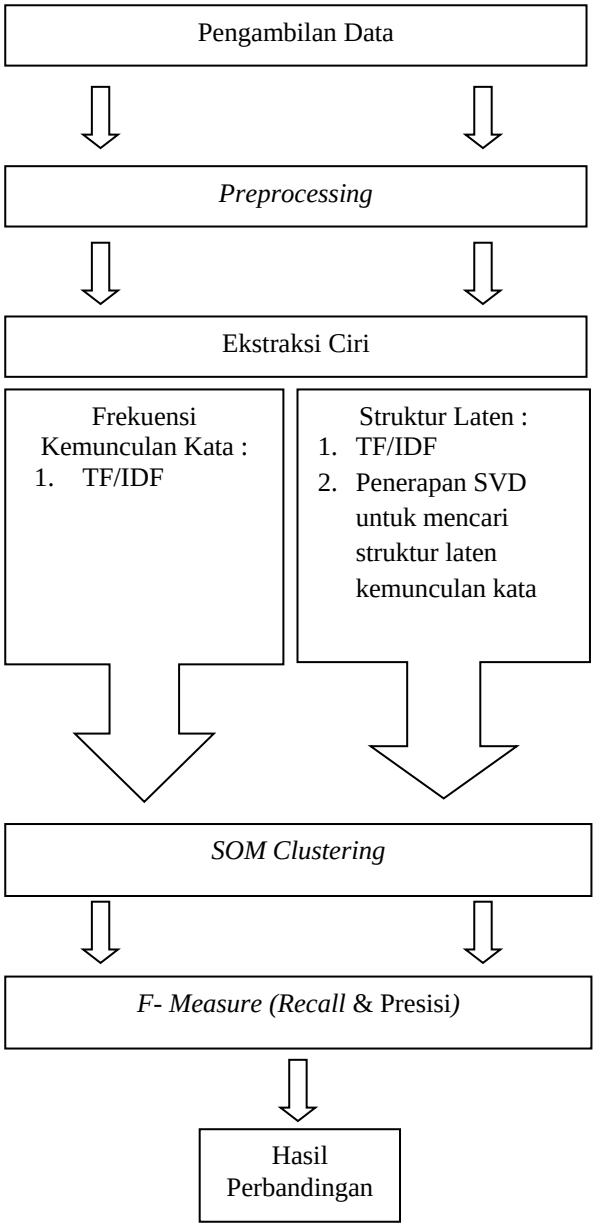
Tujuan akhir yang ingin dicapai dari penelitian ini adalah melihat sampai sejauh mana pengaruh penggunaan struktur laten jika dibandingkan dengan menggunakan frekuensi kemunculan kata ketika digunakan dalam proses pengelompokan dokumen. Perbandingan masing-masing ciri yang digunakan dapat dilihat dari hasil presisi dan *recall*, yaitu struktur laten mampu memberikan pengaruh yang positif terhadap pengelompokan atau justru sebaliknya. Gambaran umum proses investigasi ditunjukkan pada Gbr. 1.

Tahapan dimulai dengan proses pengambilan data dari internet. Dokumen dari internet umumnya bertipe *html*, sehingga format ini dibersihkan terlebih dahulu menjadi teks biasa. Selanjutnya, kumpulan dokumen dalam bentuk teks biasa masuk ke dalam tahap *preprocessing*. Pada tahap ini, dilakukan pembersihan terhadap ciri-ciri yang tidak diperlukan. Setelah melalui tahapan *preprocessing*, dokumen-dokumen dibangkitkan cirinya. Ciri-ciri dokumen disimpan dalam format matriks. Terdapat dua ciri yang dibangkitkan, yakni ciri kemunculan kata dan ciri struktur laten.

Masing-masing matriks ciri selanjutnya dikelompokkan secara *unsupervised* dengan menggunakan pendekatan SOM, sehingga muncul dua hasil pengelompokan, yakni pengelompokan berdasarkan kemunculan kata dan pengelompokan berdasarkan struktur laten. Kedua hasil inilah yang dibandingkan. Masing-masing diukur seberapa besar presisi, *recall*, dan *F-Measure* yang dihasilkan. Untuk memperkuat hasil evaluasi, dilakukan juga perbandingan antara hasil pengelompokan dengan kategori aslinya (kategori yang dibuat oleh penyedia konten). Dengan demikian, diharapkan diperoleh informasi-informasi baru yang mungkin belum terlihat melalui evaluasi presisi, *recall*, dan *F-Measure* saja.

III. Pengambilan Data

Dokumen yang diproses diambil dari situs http://detik.com, http://kompas.com, dan http://liputan6.com, khususnya untuk artikel-artikel yang terbit pada tanggal 1 September 2016 sampai dengan 30 September 2016. Data sengaja diambil dari beberapa penyedia agar dapat mewakili keberagaman artikel di internet. Proses pengambilan data dilakukan dengan bantuan salah satu pustaka perayapan web berbasis Python, yaitu Scrapy 1.1.0.



Gbr. 1 Diagram proses perbandingan pengelompokan dokumen berdasarkan frekuensi kemunculan kata dan struktur laten.

Pengambilan data dilakukan berdasarkan kategori yang telah didefinisikan oleh masing-masing situs, sehingga selalu diawali dari indeks artikel masing-masing kategori pada setiap situs. Rincian data yang diperoleh dari proses perayapan disajikan pada Tabel I.

TABEL I
RINCIAN SUMBER DATA

Kategori	Situs	Jumlah
Sepakbola	http://detik.com	367
Sepakbola	http://kompas.com	99
Sepakbola	http://liputan6.com	75
Total Artikel Sepakbola		**541**
Otomotif	http://detik.com	223
Otomotif	http://kompas.com	125
Otomotif	http://liputan6.com	208
Total Artikel Otomotif		**556**
Teknologi	http://detik.com	194
Teknologi	http://kompas.com	126
Teknologi	http://liputan6.com	192
Total Artikel Teknologi		**512**
Ekonomi	http://detik.com	438
Ekonomi	http://kompas.com	140
Ekonomi	http://liputan6.com	280
Total Artikel Ekonomi		**858**
Kesehatan	http://detik.com	166
Kesehatan	http://kompas.com	127
Kesehatan	http://liputan6.com	240
Total Artikel Kesehatan		**533**
Total Keseluruhan Artikel		**3.000 artikel**

IV. *PREPROCESSING*

Tahapan ini dilakukan untuk mempermudah proses ekstraksi ciri dengan membuang ciri-ciri yang tidak diperlukan [10]. Langkah-langkah tahapan ini adalah sebagai berikut.

1. *Tokenization* bertujuan mempermudah eksplorasi kata pada artikel yang diproses dengan cara membuang tanda-tanda pungtuasi, seperti tanda kurung, petik, dan lain sebagainya.
2. *Stopword Removal*. Kata yang terlalu sering muncul pada seluruh artikel biasanya tidak terlalu banyak berperan dalam mengidentifikasi makna suatu artikel, sehingga tidak diperlukan dalam proses pengelompokan. Contoh kata yang sering muncul pada seluruh artikel adalah kata depan, kata keterangan, kata bantu, dan lain sebagainya. Kata-kata ini sebaiknya tidak dilibatkan dalam proses pengelompokan.
3. *Stemming*, yaitu teknik untuk mengonversi kata-kata pada artikel menjadi kata dasar [11]. Satu kata dasar biasanya digunakan pada beberapa kata dengan perbedaan imbuhan. Dengan mengembalikan seluruh kata menjadi kata dasar, jumlah kata yang harus diproses dapat direduksi, sehingga proses berjalan lebih efisien.

V. EKSTRAKSI CIRI

Untuk dapat mengelompokkan dokumen, kumpulan dokumen tersebut harus dikonversi terlebih dahulu menjadi sebuah ruang vektor. Ruang vektor ini berupa matriks dua dimensi dan sering disebut dengan *Vector Space Model* (VSM) [12]. Tabel II merupakan contoh VSM, dengan baris menyatakan dokumen sedangkan kolom menyatakan ciri yang terkandung. Oleh karena itu, jumlah baris VSM sama dengan jumlah data yang diobservasi.

TABEL II
CONTOH MATRIKS CIRI (RUANG VEKTOR)

	Ciri 1	Ciri 2	Ciri 3	CiriN
Dok1	w1,1	w2,1	w3,1	wn,1
Dok2	w1,2	w2,2	w3,2	wn,2
DokN	w1,n	w2,n	w3,n	wn,n

Terdapat dua jenis ciri yang digunakan untuk kemudian dibandingkan hasilnya. Yang pertama adalah ciri frekuensi kemunculan kata dan yang kedua adalah ciri struktur laten. Mengingat besarnya dimensi ciri, maka frekuensi kata yang digunakan sebagai dasar pembuatan matriks ciri adalah kata-kata yang memiliki kemunculan lebih dari sepuluh kali. Kata-kata yang kemunculannya berada di bawah batas yang ditentukan dianggap belum bisa mewakili makna suatu dokumen.

A. *Term Frequency/Inverse Document Fequency (TF/IDF).*

TF/IDF merupakan mekanisme pembobotan suatu ciri yang berdasar pada frekuensi kemunculan kata. Secara sederhana, bobot TF/IDF memberikan nilai yang tinggi untuk kata yang sering muncul di satu dokumen tetapi jarang muncul di dokumen lainnya, serta nilai yang rendah untuk kata yang berfrekuensi tinggi pada suatu dokumen tetapi juga sering muncul di dokumen lain [12]. Persamaan bobot TF/IDF dituliskan pada persamaan berikut.

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \quad (1)$$

dengan $tf - idf_{t,d}$ merupakan bobot TF/IDF $kata_t$ pada $dokumen_d$, $tf_{t,d}$ merupakan frekuensi $kata_t$ pada $dokumen_d$, dan idf_t merupakan nilai idf dari $kata_t$, dengan

$$idf_{t,d} = log \frac{N}{df_t}. \quad (2)$$

N merupakan jumlah dokumen yang diproses dan df_t adalah jumlah dokumen yang mengandung $kata_t$.

B. *Ciri Struktur Laten*

Misalnya terdapat sembilan dokumen singkat dengan konten sebagai berikut.

1. 'Romeo dan juliet'.
2. 'Romeo terbunuh oleh belati'.
3. 'Juliet terbunuh oleh racun'.
4. 'Racun dan belati merupakan benda mematikan'.
5. 'Pak roni seorang montir mobil'.
6. 'Pak roni bekerja di sebuah dealer'.
7. *'Dealer* A menjual mobil ferrari'.
8. 'Kendaraan yang dijual di *dealer* A sangat bagus'.
9. 'Mobil ferrari merupakan kendaraan yang nyaman'.

Jika pada seluruh dokumen ini dikenakan operasi *preprocessing* dan diekstrak ciri kemunculan katanya (dikonversi menjadi matriks TF/IDF), maka akan dihasilkan

sebuah matriks dengan dimensi 9 x 17. Masing-masing baris pada matriks tersebut mewakili satu objek dokumen, sedangkan kolom mewakili ciri kemunculan kata, sehingga objek dokumen dapat dianggap sebagai vektor baris pada matriks TF/IDF.

Hubungan antara satu vektor dengan vektor lainnya dapat diketahui dengan cara menghitung koefisien korelasi. Jika suatu vektor memiliki hubungan dengan vektor lain, maka koefisien korelasi akan bernilai mendekati 1. Sebaliknya, koefisien korelasi akan bernilai mendekati -1 jika tidak ada keterkaitan antara dua buah vektor. Dengan kata lain, dokumen juga dapat dianggap demikian, yaitu dua dokumen dapat dianggap memiliki hubungan kedekatan makna ketika nilai koefisien korelasi mendekati 1 serta sebaliknya.

Pada matriks TF/IDF dokumen 1 dan dokumen 2, koefisien korelasi bernilai 0,31. Hal ini wajar mengingat pada dokumen 1 dan setidaknya ada kemiripan kemunculan kata khususnya pada kata 'Romeo'. Namun jika dilihat korelasi antara dokumen 5 dengan dokumen 8, korelasi antara keduanya bernilai -0.20, seolah-olah antara keduanya tidak memiliki hubungan keterkaitan. Padahal jika dicermati dokumen 5 dan dokumen 8 sama-sama membahas topik otomotif.

Pada umumnya, pembaca akan menganggap dokumen 5 memiliki keterkaitan dengan dokumen 8, salah satunya karena pada dokumen 5 terdapat komposisi kata yang juga terdapat pada dokumen 7, yaitu kata 'mobil'. Selanjutnya, pada dokumen 7 terdapat komposisi kata yang juga ada pada dokumen 8, yaitu kata '*dealer*'. Secara sederhana, terjadinya proses induksi antara dokumen 5 dan dokumen 8 terlihat pada Gbr. 2. Dengan adanya proses induksi kemunculan kata, seharusnya dokumen 5 berpotensi memiliki hubungan dengan dokumen 8 atau memiliki nilai korelasi > 0.

Gbr. 2 Proses induksi korelasi dari dokumen 5 menuju dokumen 8.

Induksi korelasi yang terjadi antara dokumen 5 dengan dokumen 8 dapat dianggap sebagai latensi yang terjadi antara dokumen 5 dengan dokumen 8. Untuk memunculkan latensi tersebut, diperlukan sebuah pendekatan yang mampu meningkatkan korelasi antara dokumen 5 dengan dokumen 8. Salah satu pendekatan dalam aljabar linier yang dapat digunakan untuk meningkatkan potensi korelasi antar vektor dalam suatu matriks adalah *Singular Value Decomposition* (SVD) [13]. SVD memungkinkan dapat dilihatnya potensi korelasi antar variabel sekaligus memperkecil korelasi antar variabel yang tidak berpotensi memiliki hubungan korelasi [13].

SVD merupakan salah satu teknik pemfaktoran matriks A menjadi tiga buah matriks, yaitu matriks ortogonal U, matriks diagonal S, dan *transpose* dari matriks ortogonal V [13]. SVD dapat ditunjukkan melalui persamaan berikut.

$$A_{m\times n} = U_{m\times k} S_{k\times k} V_{k\times n} \quad (3)$$

dengan $U^T U = I, V^T V = I$, kolom-kolom U merupakan vektor *eigen* yang ortonormal dari AA^T, kolom-kolom V merupakan vektor *eigen* yang ortonormal dari $A^T A$, dan S merupakan matriks diagonal yang mengandung akar pangkat dari nilai *eigen* U atau V dalam posisi yang terurut. Hasil SVD juga dapat ditulis dengan interpretasi yang berbeda. Jika himpunan vektor yang menyusun U adalah $\vec{u}_1, \vec{u}_2 \ldots \vec{u}_n$, kemudian entri diagonal utama S adalah $\sigma_1, \sigma_2 \ldots \sigma_n$, dan himpunan vektor penyusun V adalah $\vec{v}_1, \vec{v}_2 \ldots \vec{v}_n$, maka teorema SVD dapat ditulis sebagai (4).

$$A = \sum_{i=1}^{k} \sigma_i \vec{u}_i \vec{v}_i^T \quad (4)$$

Nilai-nilai σ_1 diurutkan dari yang terbesar sampai dengan yang terkecil. Apabila beberapa nilai σ_i yang besar diambil dan nilai σ_i yang mendekati nol dibuang, maka diperoleh aproksimasi dari matriks *A*. Terdapat beberapa tahapan yang bisa dilakukan untuk mencari SVD, yaitu sebagai berikut.

1. Mencari nilai U.
   a. Mencari nilai AA^T.
   b. Mencari vektor *eigen* dan nilai *eigen* dari AA^T.
   c. Mengurutkan vektor *eigen* berdasarkan nilai *eigen* dari yang terbesar ke yang terkecil.
   d. Melakukan proses ortonormalisasi Gram-Schmidt terhadap vektor-vektor *eigen* yang telah terurut.
2. Mencari nilai V.
   a. Mencari nilai $A^T A$
   b. Mencari vektor eigen dan nilai eigen dari $A^T A$
   c. Mengurutkan vektor eigen berdasarkan nilai eigen dari yang terbesar ke yang terkecil
   d. Melakukan proses ortonormalisasi gram-schmidt terhadap vektor-vektor eigen yang telah terurut.
   e. Melakukan proses *transpose* terhadap matriks ortonormal vektor-vektor eigen.
3. Mencari nilai S, yaitu membuat matriks diagonal S dengan cara mengakarpangkatkan nilai-nilai *eigen* dari U dari yang terbesar ke yang terkecil.

Bila pada matriks TF/IDF dengan dimensi 9 x 17 dikenakan operasi SVD, maka akan terbentuk tiga buah matriks yang masing-masing adalah matriks U dengan dimensi 9 x 9, matriks diagonal S dengan dimensi 9 x 9, dan matriks V dengan dimensi 9 x 17. Selanjutnya, jika hanya diambil dua kolom dari matriks U, maka akan terbentuk matriks baru yang memiliki dimensi yang lebih sederhana dari matriks TF/IDF. Jika masing-masing matriks TF/IDF dan matriks U dihitung korelasi antar dokumennya, maka perubahan korelasi terlihat seperti pada Gbr. 3. Gbr. 3 menunjukkan terjadinya penguatan korelasi antara dokumen 5 dengan dokumen 8, yang semula bernilai -0,2 meningkat menjadi 1,0, karena memang sebenarnya kedua dokumen tersebut berpotensi memiliki kedekatan makna.

C. Rasio Energi

Pada tahap akhir pembentukan matriks struktur laten, kita hanya mengambil sebagian matriks U yang merupakan hasil operasi *SVD*. Pemotongan matriks tersebut sebaiknya tidak dilakukan secara sembarangan. Hal ini dilakukan untuk menjaga kualitas kandungan informasi yang terdapat pada matriks struktur laten. Kandungan informasi yang terdapat pada matriks U dapat dianggap sebagai rasio energi yang tersisa [14].

Korelasi antar dokumen pada matriks TF/IDF

	1	2	3	4	5	6	7	8	9
1	1.0	0.3	0.3	-0.2	-0.1	-0.2	-0.2	-0.2	-0.2
2	0.3	1.0	0.2	0.1	-0.2	-0.2	-0.2	-0.3	-0.2
3	0.3	0.2	1.0	0.1	-0.2	-0.2	-0.2	-0.3	-0.2
4	-0.2	0.1	0.1	1.0	-0.2	-0.3	-0.3	-0.3	-0.3
5	-0.1	-0.2	-0.2	-0.2	1.0	0.3	-0.2	-0.2	-0.2
6	-0.2	-0.2	-0.2	-0.3	0.3	1.0	0.1	0.0	-0.2
7	-0.2	-0.2	-0.2	-0.3	-0.2	0.1	1.0	0.0	0.2
8	-0.2	-0.3	-0.3	-0.3	-0.2	0.0	0.0	1.0	0.1
9	-0.2	-0.2	-0.2	-0.3	-0.2	-0.2	0.2	0.1	1.0

Korelasi antar dokumen pada matriks struktur laten

	1	2	3	4	5	6	7	8	9
1	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0
2	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0
3	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0
4	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0
5	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0
6	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0
7	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0
8	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0
9	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0

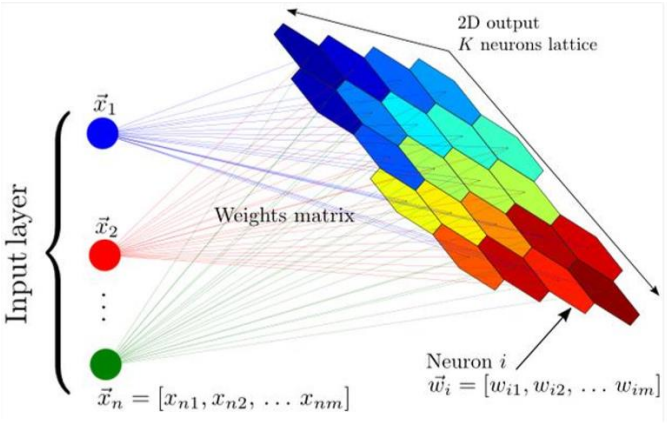
Gbr. 3 Visualisasi perubahan korelasi sebelum dan sesudah operasi SVD.

Pada hasil operasi SVD, rasio energi merupakan rasio antara total kuadrat nilai *singular* (diagonal matriks S) yang diambil dibagi dengan total kuadrat dari seluruh nilai *singular* [14]. Jika diambil dua kolom matriks U, berarti diambil dua nilai *singular* dari matriks S. Disarankan untuk mempertahankan rasio energi kurang lebih 90% agar kualitas nilai informasi pada matriks struktur laten tetap bertahan [14]. Penentuan rasio energi yang dipertahankan tentunya berdampak pula pada waktu pengelompokan. Hal ini terjadi mengingat semakin kecil rasio energi, maka semakin kecil pula dimensi matriks struktur laten, sehingga semakin kecil pula waktu pengelompokan yang diperlukan.

VI. PENGELOMPOKAN DENGAN *SELF ORGANIZING MAP* (SOM)

Tahap selanjutnya setelah pembentukan matriks ciri adalah mengelompokkan masing-masing dokumen ke dalam kelompok-kelompok yang homogen berdasarkan ciri yang dimiliki. Proses pengelompokan ini termasuk dalam kategori *unsupervised learning*, yaitu tidak ada label yang digunakan sebagai dasar pengelompokan. Label kategori asli yang diambil pada saat proses pengambilan data hanya digunakan sebagai faktor pembanding, hasil pengelompokan sudah mendekati kategori yang telah didefinisikan oleh penyedia konten atau belum.

Untuk keperluan pengelompokan tersebut, digunakan pendekatan SOM. Prinsip dasar SOM adalah mencoba untuk mentransformasikan masukan observasi yang berdimensi banyak menjadi bentuk yang lebih sederhana, biasanya berdimensi sedikit, bisa 1, 2 atau 3 [15], sehingga karakteristik objek yang diobservasi lebih mudah dipahami. SOM memiliki keunggulan dalam hal kemudahan implementasi dan mampu menyelesaikan permasalahan kompleks yang bersifat nonlinier [16].



Gbr. 4 Struktur pemetaan SOM [17].

A. *Struktur SOM*

Secara sederhana struktur SOM terdiri atas dua lapisan, yakni lapisan masukan dan lapisan keluaran (neuron). Setiap *node* pada lapisan masukan terhubung ke seluruh *node* pada neuron, tetapi setiap neuron tidak terhubung satu sama lain. Begitu juga dengan setiap *node* pada lapisan masukan, tidak terhubung antara satu dengan lainnya [17]. Ilustrasi struktur SOM ditunjukkan pada Gbr. 4.

B. *Proses Pemetaan SOM*

Terdapat dua jenis matriks yang mewakili karakteristik dokumen observasi, yakni matriks kemunculan kata, TF/IDF, dan matriks struktur laten. Masing-masing matriks dibagi menjadi dua bagian, yaitu sebagai data latih dan sebagai data uji (*testing*).

1) Pembuatan Topologi: Topologi merupakan keluaran yang diharapkan pada proses *training*. Masing-masing data latih, baik itu TF/IDF maupun latensi, berperan sebagai masukan pada proses ini. Topologi yang dihasilkan mewakili karakteristik data latih. Selanjutnya, topologi ini disebut sebagai *node* keluaran. Pada penelitian ini, jumlah *node* keluaran yang dihasilkan adalah sejumlah kelompok topik berita. Misalnya S adalah himpunan *node* keluaran yang diinginkan, sedangkan I adalah matriks data latih yang berupa himpunan vektor *node* masukan $v \in I$. Untuk masing-masing posisi, *node* $n_i \in S$ harus ditata terlebih dahulu sesuai dengan peta keluaran yang diinginkan. Begitu juga bobot pada masing-masing n_i, harus diberikan dengan kaidah $dim(n_{i_{bobot}}) = dim(I)$.

2) Pengorganisasian: Agar SOM dapat beroperasi, terdapat beberapa parameter yang harus diatur terlebih dahulu. Secara umum, keseluruhan proses dilakukan selama beberapa kali daur hidup pembelajaran. Pada beberapa referensi, parameter ini sering disebut dengan *epoch*. Selanjutnya, perilaku adaptasi S terhadap I dipengaruhi oleh dua hal, yakni laju (μ) pembelajaran dan radius ketetanggaan (σ) n_i terhadap *node* pemenang ($n_{pemenang}$). Berikut ini merupakan rincian proses pengorganisasian.

1. Penentuan *node* pemenang.

   Node pemenang $n_{pemenang} \in S$ ditentukan oleh jarak antara I terhadap n_i. $n_{pemenang}$ dipilih dari n_i yang memiliki jarak terendah terhadap I. $n_{pemenang}$ harus memenuhi kondisi berikut.

   $$\forall n_i \in S : diff\left(n_{pemenang_{bobot}}, v\right) \leq diff(n_{i_{bobot}}, v). \quad (5)$$

   Oleh sebab itu, fungsi *best matching unit*, $bmu(v)$, yang dapat menentukan $n_{pemenang}$ pada vektor masukan v dapat didefinisikan sebagai (6).

   $$bmu(v) = \arg\min diff\left(v, n_{i_{bobot}}\right), i = 0, 1, 2, \dots, |S| - 1 \quad (6)$$

   dengan $diff(x, y)$ merupakan selisih antara x dan y, atau jarak antara x terhadap y. Jarak Euclide merupakan fungsi jarak yang umum digunakan pada algoritma SOM.

   $$diff(x, y) = \sqrt{\sum_i (x_i - y_i)^2}. \quad (7)$$

2. Adaptasi Bobot.

   Penentuan *node* pemenang disertai dengan adaptasi dari seluruh bobot S menurut persamaan berikut.

   $$n_{i_{bobot}}(t+1) = n_{i_{bobot}}(t) + \mu(t).h\left(\left|n_{pemenang} - n_i\right|, t\right).(v_i - n_{i_{bobot}}(t)). \quad (8)$$

   Persamaan (8) menunjukkan bahwa penyesuaian bobot dipengaruhi oleh laju pembelajaran μ dan fungsi ketetanggaan h. Laju pembelajaran merupakan bilangan real yang bernilai $0 < \mu \leq 1$ dan selalu meluruh setiap siklus pembelajaran, sehingga $\mu(t+1) < \mu(t)$. Agar nilai μ dapat selalu meluruh setiap pergantian siklus, maka nilai μ dapat dibangkitkan dengan fungsi peluruhan eksponensial, seperti pada (9).

   $$\mu(t) = \mu_0 . k^{\left(\frac{t}{t_{max}}\right).\left(\log\left(\frac{\mu_f}{\mu_0}\right).\frac{1}{\log(k)}\right)} \quad (9)$$

   dengan μ_0 merupakan nilai inisial μ dan μ_f merupakan nilai μ akhir yang diinginkan. Untuk peningkatan performa algoritma, Haykin menyederhanakan persamaan (9) menjadi seperti berikut.

   $$\mu(t) = \mu_0 . e^{-\frac{t}{t_{max}}}.$$

   Fungsi ketetanggaan menentukan jumlah *node* yang terpengaruh oleh proses penyesuaian yang dialami oleh *node* pemenang. Secara sederhana, bukan hanya *node* pemenang yang mengalami proses penyesuaian, tetapi *node-node* yang berada di sekitar *node* pemenang juga ikut mengalami penyesuaian. Semakin jauh jarak sebuah *node* dengan *node* pemenang, maka semakin kecil pula peluang *node* tersebut untuk beradaptasi. Salah satu fungsi ketetanggaan yang umum digunakan adalah fungsi Gaussian :

   $$h(d) = e^{-\frac{d^2}{2.\sigma^2}}. \quad (10)$$

   Dari persamaan di atas, tampak bahwa fungsi ketetanggaan tidak hanya bergantung pada jarak d saja, tetapi juga bergantung pada radius σ. Di sisi lain, ukuran radius σ seharusnya selalu mengecil setiap siklus pembelajaran, sehingga notasi σ pada siklus t dapat diganti dengan $\sigma(t)$ dan fungsi ketetanggaan pada jarak d dan siklus t dapat dinotasikan dengan $h(d, t)$.

   $$h(d, t) = e^{-\frac{d^2}{2.\sigma(t)^2}}. \quad (11)$$

   Nilai dari radius $\sigma(t)$ dapat diperlakukan sama dengan nilai dari μ yang juga selalu meluruh setiap pergantian siklus pembelajaran, sehingga dapat dituliskan

   $$\sigma(t) = \sigma_0 . e^{\left(-\frac{t.\log\sigma_0}{t_{max}}\right)} \quad (12)$$

C. Proses Pengujian SOM

Node keluaran yang dihasilkan mewakili kelompok berita berdasarkan topik, sehingga jumlahnya sama dengan kategori berita. *Node* ini selanjutnya akan diujikan dengan data *testing*. Proses pengujian yang dilakukan cukup sederhana, yakni dengan menghitung jarak Euclide antara masing-masing data *testing* dengan seluruh *node* keluaran. Hasil kluster ditentukan berdasarkan jarak terdekat antara data *testing* dengan *node* keluaran dengan persamaan sebagai berikut.

$$C_n = S\left[\min\left(diff\left(I_n, \sum S\right)\right)\right] \quad (13)$$

dengan I_n adalah data *testing* ke n, C_n merupakan *cluster* untuk data *testing* ke n, dan S seluruh himpunan *node* keluaran.

VII. METODE EVALUASI PENGELOMPOKAN

Proses evaluasi didasarkan pada data kategori yang sebenarnya. Hasil pengelompokan yang dilakukan oleh proses *clustering* dibandingkan dengan hasil pengelompokan yang didasarkan pada kategori artikel sebenarnya (*Ground Truth*). Selanjutnya istilah *Ground Truth* disebut dengan GT. Perlu diingat bahwa pengelompokan yang dilakukan dalam penelitian adalah merupakan salah satu bentuk dari *unsupervised learning*, sehingga dalam menentukan kualitas pengelompokan tidak cukup hanya didasarkan pada label yang terkandung pada GT saja. Tabel III merupakan salah satu contoh perbandingan antara *clustering* dengan GT.

TABEL III
CONTOH PERBANDINGAN GT DENGAN *LABEL CLUSTER*

Dokumen	**Label GT**	***Label Cluster***
1	Bola	Ekonomi
2	Bola	Ekonomi
3	Ekonomi	Bola
4	Ekonomi	Bola

Jika dilihat dari aspek labelnya, jelas sekali hasil yang ditunjukkan merupakan pengelompokan yang sama sekali

tidak akurat. Namun, jika dilihat dari aspek *unsupervised learning*, hasil di atas merupakan proses *clustering* yang sepenuhnya akurat, karena telah mampu mengelompokkan subjek yang sejenis dalam kelompok yang sama, terlepas dari *label cluster* yang dihasilkan.

Untuk dapat mencapai evaluasi dari aspek *unsupervised learning*, diperlukan evaluasi yang tidak hanya didasarkan pada label saja, tetapi juga diukur dari pasangan antar subjek dokumen (*pairwise*) [12]. Seluruh subjek yang diteliti dipasangkan satu sama lain sehingga membentuk N(N-1)/2 pasangan, dengan ketentuan sebagai berikut.

1. Jika terdapat pasangan dokumen yang ketika di GT memiliki label yang sama dan di *cluster* juga memiliki label yang sama, maka pasangan tersebut dihitung sebagai *True Positif* (TP).
2. Jika terdapat pasangan dokumen yang ketika di GT memiliki label yang berbeda dan di *cluster* juga memiliki label berbeda, maka pasangan tersebut dihitung sebagai *True Negatif* (TN).
3. Jika terdapat pasangan dokumen yang ketika di GT memiliki label yang berbeda, tetapi di *cluster* memiliki label yang sama, maka pasangan tersebut dihitung sebagai *False Positif* (FP).
4. Jika terdapat pasangan dokumen yang ketika di GT memiliki label yang sama, tetapi di *cluster* memiliki label berbeda, maka pasangan tersebut dihitung sebagai *False Negatif* (FN).

Masing-masing nilai TP, TN, FP, dan FN digunakan sebagai dasar untuk menghitung presisi dan *recall* dengan persamaan sebagai berikut.

$$P = \frac{TP}{TP + FP} \tag{14}$$

dengan P merupakan nilai presisi yang berkisar antara 0 sampai dengan 1.

$$R = \frac{TP}{TP + FN} \tag{15}$$

dengan R merupakan nilai *recall* yang berkisar antara 0 sampai dengan 1. Setelah nilai presisi dan recall diketahui, selanjutnya nilai *F-Measure* dapat dicari dengan (15).

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R}. \tag{15}$$

Nilai *F-Measure* dapat memberikan bobot yang kuat terhadap nilai *FN* dibanding *FP* dengan cara mengatur nilai $\beta > 1$.

VIII. EVALUASI

A. Matriks Ciri

Matriks ciri yang terbentuk disimpan ke dalam format *.csv*. Untuk masing-masing jenis matriks, terdapat dua bagian yaitu *testing* dan *training*. Data *training* diambil dari 1000 sampel dokumen secara acak, sedangkan sisanya digunakan untuk *testing*. Pembagian data pada matriks TF/IDF yang merupakan representasi ruang vektor frekuensi kemunculan kata disajikan pada Tabel IV.

TABEL IV
RINCIAN MATRIKS TF/IDF

No.	Nama Berkas	Bagian	Jumlah data	Dimensi
1	all-train.csv	*Training*	1.000	1.000 x 3.486
2	all-test.csv	*Testing*	2.000	2.000 x 3.486

Untuk menghasilkan matriks yang menggambarkan struktur laten diperlukan tambahan waktu. Tambahan waktu yang diperlukan bergantung pada besarnya rasio energi yang dipertahankan. Semakin besar rasio energi yang dipertahankan, semakin besar pula waktu yang dibutuhkan. Begitu pula sebaliknya. Tabel V menunjukkan kebutuhan waktu yang diperlukan berdasarkan rasio energi yang dipertahankan.

TABEL V
RINCIAN MATRIKS STRUKTUR LATEN

Rasio Energi	Bagian	Dimensi	Waktu Tambahan
95%	*Training*	1.000 x 1.500	59,08 detik
	Testing	2.000 x 1.500	
94%	*Training*	1.000 x 1.470	57,81 detik
	Testing	2.000 x 1.470	
93%	*Training*	1.000 x 1.380	55,19 detik
	Testing	2.000 x 1.380	
90%	*Training*	1.000 x 1.200	44,6 detik
	Testing	2.000 x 1.200	
89%	*Training*	1.000 x 1.110	38,91 detik
	Testing	200 x 1.110	
87%	*Training*	1.000 x 1.020	38,29 detik
	Testing	2.000 x 1.020	
84%	*Training*	1.000 x 900	31,41 detik
	Testing	2.000 x 900	

Rasio energi yang dipertahankan terlalu besar (mendekati 100%) kurang baik karena matriks kurang bisa memunculkan latensi yang terkandung. Sebaliknya, rasio energi yang terlalu kecil juga kurang baik, karena matriks yang dihasilkan kehilangan terlalu banyak informasi yang mewakili karakteristik dokumen. Maka, pada penelitian ini diambil latensi dengan rasio energi yang berada di antara 80% sampai dengan 95%.

B. Evaluasi Hasil Pengelompokan

Setelah diperoleh matriks ciri, baik itu berdasarkan frekuensi kemunculan kata maupun struktur laten dengan berbagai tingkat rasio energi yang berbeda-beda, tahap selanjutnya adalah proses pengelompokan dokumen berdasarkan masing-masing jenis matriks ciri.

Tabel VI menunjukkan adanya peningkatan kualitas ketika digunakan beberapa jenis matriks struktur laten, jika dibandingkan ketika digunakan frekuensi kemunculan kata. Baris yang diarsir merupakan matriks struktur laten yang menghasilkan kombinasi presisi dan *recall* terbaik. Meskipun untuk membangkitkan matriks struktur laten diperlukan waktu tambahan, penggunaan matriks tersebut ternyata dapat meningkatkan efisiensi waktu pada saat proses pengelompokan.

TABEL VI
EVALUASI HASIL PENGELOMPOKAN

Matriks	Rasio Energi	Presisi	*Recall*	*F-Measure*
TFIDF	100%	0,68	0,72	0,7
Struktur Laten	95%	0,72	0,74	0,73
Struktur Laten	94%	0,72	0,75	0,74
Struktur Laten	93%	0,72	0,74	0,73
Struktur Laten	90%	0,72	0,74	0,73
Struktur Laten	89%	0,73	0,75	0,74
Struktur Laten	87%	0,72	0,74	0,73
Struktur Laten	84%	0,72	0,74	0,73

TABEL VII
PENINGKATAN EFISIENSI STRUKTUR LATEN

Matriks	Rasio Energi	Dimensi Ciri	Waktu *Training*
TFIDF	100%	3.486	≈ 1,48 detik
Struktur Laten	95%	1.500	≈ 0,82 detik
Struktur Laten	94%	1.470	≈ 0,80 detik
Struktur Laten	93%	1.380	≈ 0,76 detik
Struktur Laten	90%	1.200	≈ 0,73 detik
Struktur Laten	89%	1.110	≈ 0,69 detik
Struktur Laten	87%	1.020	≈ 0,63 detik
Struktur Laten	84%	900	≈ 0,62 detik

TABEL VIII
EVALUASI HASIL PENGELOMPOKAN

Kategori Asli	Kategori Versi TF/IDF	Kategori Versi Struktur Laten dengan rasio 89%
Olahraga (337)	329 (olahraga) 19 (otomotif) 3 (teknologi) 1 (kesehatan)	329 (olahraga) 8 (otomotif) 3 (teknologi) 1 (kesehatan)
Ekonomi (568)	187 (ekonomi) 40 (otomotif) 16 (teknologi) 1 (kesehatan)	268 (ekonomi) 20 (teknologi) 7 (otomotif) 2 (kesehatan) 1 (olahraga)
Otomotif (394)	340 (ekonomi) 313 (otomotif) 20 (teknologi) 2 (kesehatan) 1 (olahraga)	360 (otomotif) 259 (ekonomi) 15 (teknologi) 2 (kesehatan) 1 (olahraga)
Teknologi (352)	306 (teknologi) 23 (ekonomi) 19 (otomotif) 4 (kesehatan)	307 (teknologi) 29 (ekonomi) 18 (otomotif) 5 (kesehatan)
Kesehatan (349)	341 (kesehatan) 18 (ekonomi) 7 (teknologi) 7 (olahraga) 3 (otomotif)	339 (kesehatan) 12 (ekonomi) 7 (teknologi) 6 (olahraga) 1 (otomotif)

Selanjutnya, Tabel VII menunjukkan terjadinya peningkatan waktu *training* untuk jenis matriks struktur laten. Hal ini disebabkan dimensi ciri yang digunakan lebih kecil dari dimensi ciri yang digunakan oleh TF/IDF, sehingga proses *training* dapat berjalan lebih cepat. Meskipun demikian, berkurangnya dimensi ciri terhadap dimensi aslinya tidak berdampak buruk terhadap kualitas pengelompokan.

Untuk dapat mengetahui lebih jauh dampak penggunaan struktur laten terhadap proses pengelompokan, dilakukan perbandingan antara kategori asli dari penyedia dengan kategori hasil pengelompokan, baik itu berdasarkan frekuensi kata maupun struktur laten.

Tabel VIII menunjukkan penggunaan matriks TF/IDF tidak cukup bagus dalam mengidentifikasi dominasi kategori artikel. Ini terlihat pada artikel dengan kategori otomotif, yaitu penggunaan matriks TF/IDF justru menunjukkan dominasi artikel bertopik ekonomi pada kelompok berita otomotif. Hal yang berbeda ditunjukkan pada hasil pengelompokan berdasarkan struktur laten. Dapat dilihat bahwa dominasi artikel untuk topik otomotif juga tetap didominasi oleh artikel bertopik otomotif pula.

IX. KESIMPULAN

Beberapa hal dapat disimpulkan berdasarkan serangkaian evaluasi yang telah dilakukan. Penggunaan struktur laten dapat mengurangi dimensi ciri secara signifikan. Hal ini dapat dilihat pada matriks struktur laten, bahwa dengan rasio energi sebesar 89% dimensi ciri yang terbentuk hanya sebesar 1.110 ciri atau sekitar 32% dari dimensi ciri TF/IDF. Dengan berkurangnya dimensi ciri yang digunakan, penggunaan struktur laten berdampak positif pada waktu yang diperlukan untuk proses pengelompokan. Selain berpotensi meningkatkan efisiensi waktu, struktur laten juga dapat meningkatkan kualitas pengelompokan sebesar 4%. Meskipun secara persentase tidak terlalu besar, tetapi matriks struktur laten mampu mengidentifikasi dominasi kategori artikel secara lebih tepat jika dibandingkan menggunakan TF/IDF.

Meskipun dapat meningkatkan efisiensi waktu dan kualitas hasil pengelompokan, penggunaan struktur laten membutuhkan waktu tambahan pada tahap *preprocessing*, yaitu untuk proses pembentukan matriks ciri. Penambahan artikel baru mengakibatkan perubahan pada matriks struktur laten, sehingga menyebabkan matriks ini tidak cocok digunakan untuk proses pengelompokan yang bersifat incremental, sehingga peluang yang dapat dilakukan dalam rangka pengembangan penelitian ini, salah satunya adalah masih diperlukan adanya kajian yang mendalam mengenai penggunaan struktur laten untuk pengelompokan yang bersifat inkremental, sehingga penambahan artikel baru tidak memerlukan pembangkitan ulang matriks ciri struktur laten.

REFERENSI

[1] A. Z. Arifin dan N. A. Setiono, "Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering," *Proceeding of Seminar on Intelligent Technology and Its Applications (SITIA)*, Surabaya, 2002.

[2] D. Y. Liliana, A. Hardianto dan M. Ridok, "Indonesian News Classification using Support Vector Machine," *World Academy of Science, Engineering and Technology*, vol. 57, pp. 767-770, 9 2011.

[3] Ambarwti dan E. Winarko, "Pengelompokan Berita Indonesia Berdasarkan Histogram Kata Menggunakan Self-Organizing Map," *IJCCS*, vol. VIII, no. 1, p. 101, 2014.

[4] Z. Oren, E. Oren, M. Omid dan R. M. Karp, "Fast and Intutive Clustering of Web Document," *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Newport Beach, CA, 1997.

[5] M. S. K. Khaled M. Hammouda, "Efficient Phrase-Based Document Indexing for Web Document Clustering," *IEEE Transactions On Knowledge and Data Enginering*, vol. 16, no. 10, pp. 1279-1296, 2004.

[6] H. Chim dan X. Deng, "Efficient Phrase-Based Document Similarity for Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 9, pp. 1217-1229, 2008.

[7] H. A. Putranto, O. Setyawati dan Wijono, "Pengaruh Phrase Detection dengan POS-Tagger terhadap Akurasi Klasifikasi Sentimen menggunakan SVM," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 5, no. 4, pp. 252-259, 2016.

[8] M. Bakhshi, M.-R. Feizi-Derakhshi dan E. Zafarani, "Review and Comparison between Clustering Algorithms with Duplicate Entities Detection Purpos," *International Journal of Computer Science & Emerging Technologies* , vol. 3, no. 3, 2012.

[9] O. A. Abbas, "Comparisons Between Data Clustering Algorithm," *Fourth International Conference on Engineering and Technology Research*, vol. 5, no. 3, 2008.

[10] F. Gorunescu, *Data Mining : Concepts, Models and Techniques*, vol. 12, Berlin: Springer, 2011.

[11] P. B. S. Wiguna dan B. S. Hantono, "Peningkatan Algoritma Porter Stemmer Bahasa Indonesia berdasarkan Metode Morfologi dengan Mengaplikasikan 2 Tingkat Morfologi dan Aturan Kombinasi Awalan dan Akhiran," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 2, no. 2, 2013.

[12] C. D. Manning, P. Raghavan dan H. Schütze, *An Introduction to Information Retrieval*, Cambridge, England: Cambridge UniversityPress , 2009.

[13] K. Baker, "Singular value decomposition tutorial", Columbus, Ohio: The Ohio State University, 2005.

[14] J. Leskovec , A. Rajaraman dan J. D. Ullman , *Mining of Massive Datasets*, Cambridge, England: Cambridge University Press, 2014.

[15] D. T. Larose, *Discovering Knowledge Data : An Introduction to Data Mining*, New Jersey: John Willey & Sons, Inc., 2005.

[16] M. A. Muslim, M. Ishikawa dan T. Furukawa, "Task Segmentatioin in a Mobile Robot by mnSOM and Clustering With Spatio-Temporal Contiguity," *International Conference on Neural Information Processing*, Berlin, Heidelberg, 2007.

[17] K. Pang, "Self-Organizing Maps," 2003. [Online]. Available: https://www.cs.hmc.edu/~kpang/nn/som.html. [Diakses 14 Mei 2015].