JURNAL TEKNOLOGI TECHNOSCIENTIA ISSN: 1979-8415
Vol. 8 No. 2 Februari 2016

# EFISIENSI PHRASE SUFFIX TREE DENGAN SINGLE PASS CLUSTERING UNTUK PENGELOMPOKAN DOKUMEN WEB BERBAHASA INDONESIA

**Desmin Tuwohingide[1], Mika Parwita[2] , Agus Zainal Arifin[3] , Diana Purwitasari [4]**

[1,2,3,4] Teknik Informatika, , Institut Teknologi Sepuluh Nopember;
[1] Jurusan Teknik Komputer dan Komunikasi, Politeknik Negeri Nusa Utara

Masuk: 4 Oktober 2015, revisi masuk: 13 Nopember 2015, diterima:6 Januari 2016

**ABSTRACT**

The number of indonesian documents which available on internet is growing very rapidly. Automatic documents clustering shown to improving the relevant documents search results of many found documents. *Suffix tree* is one of documents clustering method that developed, because it is proven to increase precision. In this paper, we propose a new method to clustering indonesian web documents based on phrase efficiency in the choice process of *base cluster* with the combination of documents frequency and term frequency calculation on the phrase with a *single pass clustering algorithm* (SPC). Every phrase that is considered as the base cluster will be vectored then calculate of the *term frequency* and *document frequency*. Furthermore, the documents will be calculate their similarity based on the *tf-idf weighted* using the *cosine similarity* and documents clustering is done by using a *single pass clustering algorithm*. The proposed method is tested on 6 dataset with number of different document 10, 20, 30, 40, 50 and 60 documents. The experiment result show that the proposed method succeeded clustering indonesian web documents by reducing the leaf node with no derivative and produces the *F-measure* an average of 0.78 while STC traditional produces the *F-measure* an average of 0.55.This result prove that the efficiency of phrase by phrase choice on internal nodes and leaf nodes that have derivative, and a combination of term frequency and document frequency calculation on the *base cluster*, gives a significant impact on the process of clustering documents.

**KeywordS:** Documents Clustering, *Single-Pass Clustering*, *Suffix Tree*.

**INTISARI**

Jumlah dokumen berbahasa Indonesia yang tersedia di internet tumbuh dengan sangat pesat. Pengelompokan dokumen secara otomatis terbukti meningkatkan hasil pencarian dokumen yang relevan dari sekian banyak dokumen yang ditemukan. Salah satu metode yang berkembang dalam pengelompokan dokumen adalah *suffix tree* karena terbukti meningkatkan *precision*. Penelitian ini mengusulkan metode baru untuk mengelompokan dokumen web berbahasa Indonesia berdasarkan efisiensi *phrase* pada proses pemilihan *base cluster* dengan kombinasi perhitungan *document frequency* dan *term frequency* pada *phrase suffix tree* dengan algoritma *Single Pass Clustering* (SPC). Setiap *phrase* yang dianggap sebagai *base cluster* akan divektorkan kemudian dilakukan perhitungan *document frequency* dan *term frequency*. Selanjutnya, Setiap dokumen akan dihitung kemiripannya berdasarkan pembobotan tf-idf menggunakan *cosine similarity* dan pengelompokan dokumen dilakukan dengan menggunakan algoritma SPC. Pengujian dilakukan pada 6 dataset dengan jumlah dokumen yang berbeda yaitu 10, 20, 30, 40, 50 dan 60 dokumen. Hasil pengujian menunjukkan metode yang diusulkan berhasil mengelompokkan dokumen web berbahasa Indonesia dengan mereduksi *leaf node* tanpa anak dan menghasilkan nilai *F-measure* rata-rata 0,78 sedangkan STC tradisional menghasilkan *F-measure* rata-rata 0,55. Hal ini menunjukkan bahwa efisiensi *phrase* melalui pemilihan *phrase* pada *internal node* dan *leaf node* yang memiliki anak serta kombinasi perhitungan *term frequency*, dan *document frequency* pada *base cluster* memberi dampak yang signifikan pada proses pengelompokan dokumen.
**Kata Kunci:** Pengelompokan dokumen , *Single-Pass Clustering*, *Suffix Tree*

[1]tdesmin@ymail.com

## PENDAHULUAN

Banyaknya informasi yang dipublikasikan melalui internet memberi dampak penyebaran informasi yang cepat dari berbagi sumber. Namun, hal ini juga mengakibatkan sulitnya untuk menemukan informasi yang relevan atau sesuai dengan kebutuhan pengguna. Kesulitan menemukan informasi atau menyaring dokumen merupakan salah satu permasalahan yang dibahas dalam sistem temu kembali informasi. Salah satu penelitian yang dikembangkan untuk menangai masalah ini adalah dengan cara mengelompokkan dokumen.

Pengelompokan dokumen berdasarkan topik telah banyak dilakukan pada penelitian sebelumnya. Algoritma *Suffix Tree Clustering* (STC) merupakan salah satu algoritma yang berkembang dalam pengelompokan dokumen. STC pertama kali digunakan untuk mengelompokan hasil pencarian dari mesin pencari (Zamir & Etzioni, 1998). Algoritma STC dinilai memiliki tingkat *precision* yang tinggi karena menggunakan *phrase* sebagai dasar pembentukan *cluster* sehingga dimungkinkan terjadinya *overlapping cluster* (Arifin, Darwanto, Navastara & Ciptaningtyas, 2008). Hal ini didasarkan bahwa setiap dokumen bisa memiliki lebih dari satu topik berdasarkan *phrase* yang terdapat pada dokumen tersebut. Kelebihan lain dari *suffix tree* adalah menyimpan semua *phrase* yang ada secara terstruktur untuk menunjukkan tingkat kemiripan dokumen (Chim & Deng, 2008). Walaupun *suffix tree* memiliki kelebihan pada struktur datanya, pada penelitian lain diungkapkan bahwa STC memiliki kelemahan dimana beberapa *node* dapat terlabeli dengan *phrase* yang sama (Hammouda & Kamel, 2004). Selain itu, proses *scoring* pada STC yang hanya berdasarkan perhitungan *document frequency* (df) dan jumlah kata yang membentuk *base cluster* dianggap bisa ditingkatkan dengan perhitungan *term frequency* (tf). Sehingga pada beberapa penelitian dilakukan efisiensi *phrase* berdasarkan perhitungan *term frequency* dengan memetakan semua *node* atau semua *phrase* yang terbentuk dari *suffix tree* kemudian dilakukan pembobotan tf-idf pada setiap *phrase* (Chim & Deng, 2008; Jain & Maheshwari, 2013). Selain itu, pembobotan pada *phrase suffix tree* juga bisa dilakukan untuk mereduksi jumlah *phrase* (Huang, Yin & Hou, 2011).

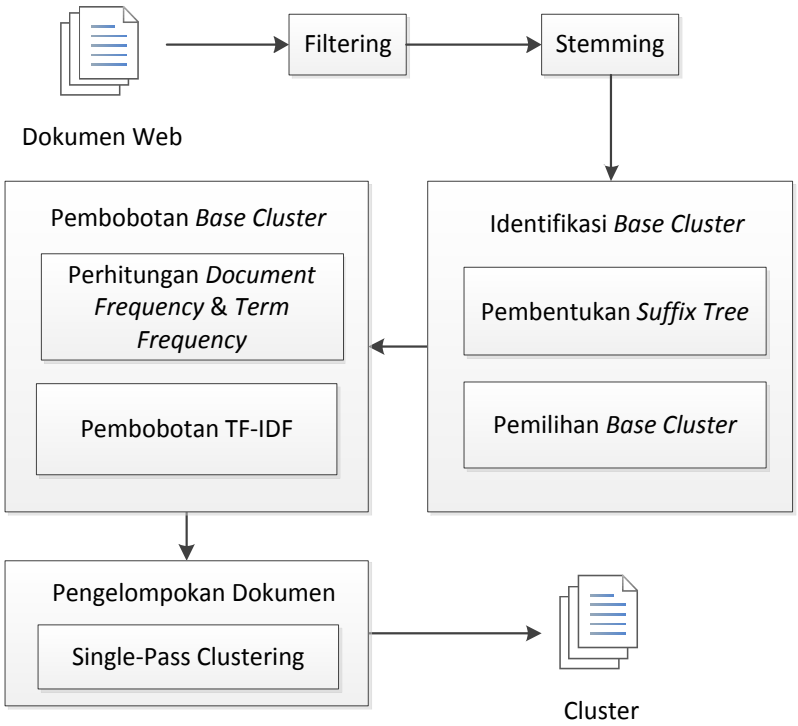
*Single Pass Clustering* (SPC) adalah metode yang digunakan untuk mengelompokan dokumen satu per satu dan setiap pembentukan *cluster* selalu dilakukan evaluasi atau perhitungan kembali representasi *cluster*. Beberapa penelitian telah menggunakan SPC untuk pengelompokan dokumen berbahasa Indonesia (Arifin & Novan, 2002; Februariyanti & Zuliarso, 2012). Pengelompokan dokumen berita online berbahasa Indonesia menggunakan STC telah dilakukan (Arifin, Darwanto, Navastara & Ciptaningtyas, 2008). Penggunaan metode STC untuk pengelompokan dokumen berbahasa Indonesia dapat ditingkatkan dengan melakukan efisiensi *phrase* berdasarkan pemilihan *phrase* yang terdapat pada *internal node* dan *leaf node* yang memiliki anak dan melakukan perhitungan *term frequency* dan *document frequency* pada setiap *base cluster*.

Pada penelitian ini metode baru untuk mengelompokan dokumen web berbahasa Indonesia berdasarkan efisiensi *phrase* pada proses pemilihan *base cluster* dengan kombinasi perhitungan *document frequency* dan *term frequency* pada *phrase suffix tree* dengan algoritma *Single Pass Clustering* (SPC). *Phrase* yang digunakan pada penelitian ini adalah *phrase suffix tree* yang terlabeli pada *internal node* dan *leaf node* yang memiliki anak. Setiap *phrase* yang terpilih dianggap sebagai *base cluster* yang kemudian akan diproses dengan menghitung *document frequency* dan *term frequency*. Selanjutnya, dokumen akan dikelompokan dengan metode SPC.

## METODE

Data yang digunakan untuk uji coba berupa kumpulan dokumen teks bahasa Indonesia yang dikumpulkan dari situs Kompas dengan alamat URL www.kompas.com. Dokumen berita berkisar dari tanggal 11 Januari 2008 sampai 4 Juli 2008. Jumlah data uji sebanyak 60 dokumen yang terbagi

dalam 12 kategori. Struktur format data berupa isi berita, tanggal, dan kategori untuk masing-masing dokumen. Kategori dokumen pada data uji digunakan untuk pembanding dengan hasil dari pengelompokan dokumen dari metode yang diusulkan.

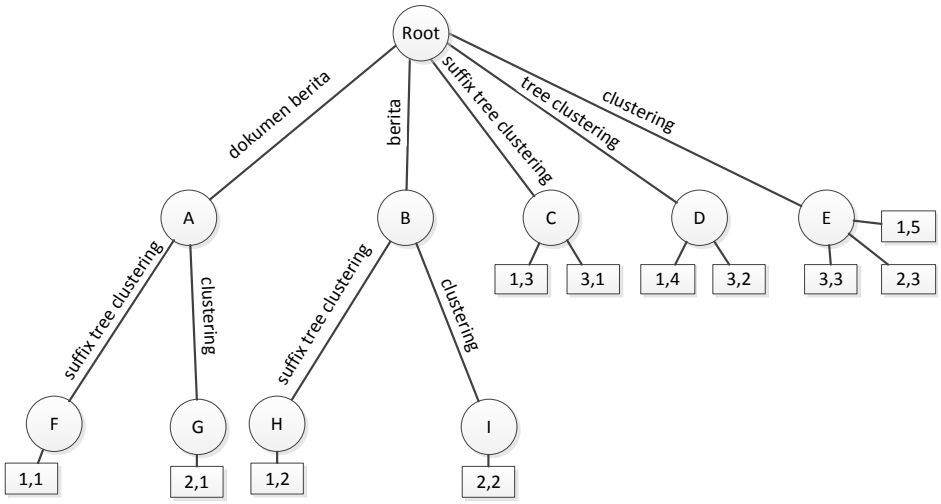



Gambar 1. Tahap usulan metode.

Sebelum dilakukannya proses pengelompokan dokumen, terlebih dahulu dilakukan proses pembersihan dokumen dengan 2 tahap, yaitu tahap *filtering* yang dilakukan untuk membersihkan dokumen dari tag-tag HTML dan proses penghapusan *stopword* atau kata-kata yang dianggap tidak memiliki makna penting dalam dokumen yang disimpan dalam stoplist. Tahap kedua adalah *stemming* yang merupakan proses pengambilan kata dasar. Setelah itu terdapat 3 tahap utama yang dilakukan, yaitu identifikasi *base cluster*, pembobotan dan pengelompokan *base cluster* (Gambar 1).

Tahap identifikasi *base cluster* terdiri dari dua tahap yaitu tahap dimana *phrase* dibentuk dengan menemukan *shared phrase* antar dokumen (Zamir & Etzioni, 1998) dan tahap kedua adalah pemilihan *phrase* yang menjadi *base cluster* . Metode ini merepresentasikan dokumen sebagai kumpulan kata-kata.

Proses pembentukan *phrase suffix tree* terdiri dari beberapa langkah. Pertama, mengidentifikasi semua kalimat yang terdapat dalam dokumen. Untuk setiap akhiran kalimat akan diidentifikasi sebagai *phrase*. Selanjutnya akan dilakukan pengecekan apakah *phrase* sudah ada pada struktur *suffix tree* yang telah terbentuk. Apabila *phrase* sudah ada, maka dilakukan penambahan informasi *phrase* berupa nomor dokumen dan posisi *phrase* ke dalam *node* yang mewakili *phrase* tersebut. Apabila *phrase* tidak ada pada struktur *suffix tree*, maka *phrase* akan ditambahkan sebagai *node* baru disertakan dengan label *phrase* yang baru ditambahkan. *Phrase* yang terlabeli pada semua *internal node* dan *leaf node* yang memiliki anak akan diambil sebagai *base cluster*.




Gambar 2. Pembentukan *suffix tree*.

Contoh pembentukan *suffix tree* dapat dilihat pada Gambar 2. Pada Gambar 2, *suffix tree* dibentuk berdasarkan 3 dokumen. Dimana dokumen pertama memiliki kumpulan kata "dokumen berita suffix tree clustering", dokumen kedua "dokumen berita clustering", dan dokumen ketiga "suffix tree clustering". *Root* merupakan pusat terbentuknya *node*. *Node* yang terbentuk digambarkan dengan lingkaran (A sampai I). Setiap *node* merepresentasikan *phrase* dari dokumen dan dilabelkan dengan nomor dokumen dan posisi *phrase*. Sebagai contoh node G untuk menyatakan *phrase* "dokumen berita clustering" pada dokumen 2 di posisi *phrase* pertama. Kondisi sebuah *node* dengan memiliki label lebih dari satu menyatakan *phrase* yang direpresentasikan pada *node* tersebut terdapat di beberapa dokumen. Sebagai contoh pada *node* D yang menyatakan *phrase* "tree clustering" terdapat pada dokumen 1 di posisi *phrase* keempat dan pada dokumen 3 di posisi *phrase* kedua.

Pada tahap pemilihan *base cluster*, setiap *phrase* yang terlabeli di *internal node* dan *leaf node* yang memiliki anak akan dianggap sebagai *base cluster*, sementara *phrase* yang terlabeli di *leaf node* yang tidak memiliki anak akan diabaikan. Pada contoh Gambar 2, *node* yang akan terpilih sebagai *base cluster* adalah *node* A,B,C,D,E yang terlabeli dengan *phrase* dokumen berita, berita, *suffix tree clustering*, *tree clustering*, dan *clustering*. *Node* F,G,H,I tidak akan diproses karena *phrase* yang terlabeli di *node* tersebut telah di terlabeli di *internal node*. Proses ini dilakukan untuk mereduksi jumlah *phrase* yang akan menjadi *base cluster* dan menghapus *node* yang terlabeli dengan *phrase* yang sama.

Sebelum dilakukan pembobotan, setiap *phrase* yang dijadikan *base cluster* diubah menjadi vektor. Vektor ini akan merepresentasikan sebuah dokumen dengan sekumpulan *phrase*. Misalkan $t_1, t_2, \dots, t_n$ menyatakan *phrase* yang digunakan untuk mengindeks *database* yang terdiri dari dokumen $D_1, D_2, \dots, D_n$, maka dokumen $D_1$ dinyatakan dengan $D_1 = (a_{i1}, a_{i2}, \dots, a_{in})$,dimana $a_{ij} = bobot\ phrase\ t_j$ dalam dokumen $D_1$. *Node* yang terbentuk dari *suffix tree* kemudian dipetakan ke dalam bentuk vektor.

Langkah selanjutnya adalah menghitung *document frequency* (*df*) dan *term frequency* (*tf*) pada setiap *base cluster/node* yang terpilih. Nilai $df(v)$ didapat dengan menghitung berapa banyak dokumen berbeda yang melewati sebuah *node* $v$ sedangkan $tf(v,d)$ didapat dengan menghitung berapa kali sebuah dokumen $d$ melewati *node* $v$ tersebut (Chim & Deng, 2008). Kemudian dilakukan pembobotan tf-idf berdasarkan N jumlah dokumen, *document frequency* (*df*), dan *term frequency* (*tf*) menggunakan persamaan (1). *Tf-idf* adalah salah

satu metode pembobotan frekuensi kata dalam sebuah dokumen berdasarkan jumlah kemunculannya (Huang, Yin & Hou, 2011).

$$w(v,d) = \left(1 + \log\left(tf(v,d)\right)\right) \times log\left(1 + \frac{N}{df(v)}\right) \quad \text{.....(1)}$$

Algoritma yang digunakan untuk pengelompokan dokumen adalah algoritma *Single Pass Clustering* (SPC). Algoritma SPC merupakan metode yang melakukan pengelompokan data satu demi satu. Setiap data yang akan dikelompokan akan dihitung kemiripannya untuk menentukan data tersebut masuk ke *cluster* mana.

Pengelompokan dokumen menggunakan algoritma SPC, terdiri dari beberapa tahap (Klampanos, Jose & van Rijsbergen, 2006). Tahap penting pada SPC adalah menentukan dokumen $P_i$ dikelompokkan ke *cluster* $C_j$ mana berdasarkan nilai kemiripan dokumen dengan masing-masing *cluster* yang ada. Kondisi untuk dokumen pertama $P_1$, digunakan sebagai representasi *cluster* pertama $C_1$. Hal ini dikarenakan belum adanya *cluster* yang terbentuk. Selanjutnya yaitu melakukan perhitungan kemiripan dokumen $P_{i+1}$ dengan keseluruhan *cluster* $C_{j...k}$. Jika nilai kemiripan $Sim_{x,y}$ lebih besar dari nilai *threshold* $t$ ($Sim_{x,y} > t$), maka dokumen dikelompokan ke *cluster* $C_j$ kemudian dilakukan perhitungan ulang vektor representasi *cluster* $C_j$. Apabila sebaliknya, nilai kemiripan $Sim_{x,y}$ tidak lebih besar dari *threshold* $t$, maka dokumen digunakan sebagai representasi *cluster* baru $C_{j+1}$. Kemudian jika masih ada dokumen yang belum dikelompokan, maka dilanjutkan perhitungan dokumen $P_{i+1}$ terhadap masing-masing *cluster* yang sudah terbentuk sampai semua dokumen selesai kelompokkan.

Untuk menjaga *overlapping cluster*, tidak diberlakukan kondisi $S(max)$ atau kondisi dimana satu dokumen memiliki nilai kemiripan lebih dari *threshold t* pada dua *cluster* yang berbeda maka dokumen tersebut hanya akan dimasukkan kedalam *cluster* yang nilai kemiripannya paling besar. Hal ini dilakukan agar dapat meningkatkan *precision* dokumen yang kemungkinan mengandung lebih dari satu topik pembahasan.

Perhitungan kemiripan yang digunakan adalah *cosine similarity*. *Cosine similarity* menghitung kemiripan antar pasangan dokumen yang akan dikelompokan. Kemiripan $Sim_{x,y}$ dihitung menggunakan persamaan (2). Perhitungan dilakukan antara $d_x = \{x_1, x_2, \dots x_n\}$ vektor dan vektor $d_y = \{y_1, y_2, \dots y_n\}$, dimana $d_x, d_y$ adalah dokumen dan $x_1, y_1$ adalah bobot dari *node term* $v_i$ (Chim & Deng, 2008).

$$Sim_{x,y} = \frac{\overrightarrow{d_x} \cdot \overrightarrow{d_y}}{|\overrightarrow{d_x}| \times |\overrightarrow{d_y}|} \quad \text{.....(2)}$$

Metode yang digunakan untuk mengevaluasi hasil pengujian adalah *F-Measure*. *F-Measure* merupakan standar yang digunakan untuk mengevaluasi algoritma *clustering* dan klasifikasi pada bidang temu kembali informasi. Perhitungan *F-measure* menggunakan *precision* dan *recall*.

$$Pre_{ij} = \frac{N_{ij}}{N_j} \quad \text{.....(3)}$$

$$Rec_{ij} = \frac{N_{ij}}{N_i} \quad \text{.....(4)}$$

*Presicion* ($Pre_{ij}$) dihitung berdasarkan perbandingan banyaknya dokumen kategori *i* pada *cluster j* ($N_{ij}$) dengan jumlah seluruh dokumen dalam *cluster j* ($N_j$). Nilai *precision* dihitung menggunakan persamaan (3). *Recall* ($Rec_{ij}$) dihitung berdasarkan perbandingan banyaknya dokumen kategori *i* pada *cluster j* ($N_{ij}$) dengan jumlah dokumen dalam kategori *i* ($N_i$). Recall dihitung menggunakan persamaan (4).

$$F_{i,j} = \frac{2 \times Pre_{ij} \times Rec_{ij}}{Pre_{ij} + Rec_{ij}} \quad \text{.....(5)}$$

Nilai *F-Measure* kategori *i* pada *cluster j* diperoleh dengan menggabungkan nilai *precision* dan *recall*. Nilai tersebut dihitung menggunakan persamaan (5).

$$F = \sum_i^m \frac{N_i}{N} max\{F_{i,j}\} \quad \text{.....(6)}$$

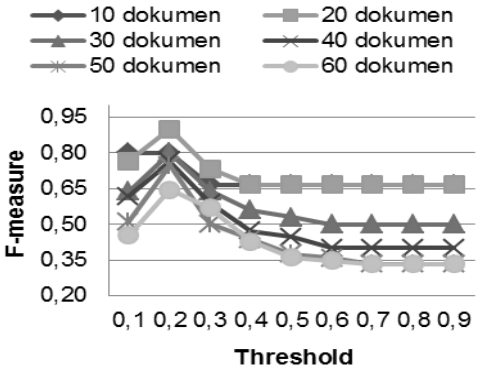
*F-measure* keseluruhan diperoleh dengan menghitung jumlah dokumen pada kategori *i* ($N_i$) dibagi dengan jumlah dokumen (*N*) dan dikalikan dengan *F-measure* tertinggi untuk kategori *i* ($maxF_{ij}$). Kemudian hasil untuk masing-masing kategori dijumlahkan sebanyak jumlah kategori *m*. Perhitungan ini bisa dilihat pada persamaan (6).

**PEMBAHASAN**

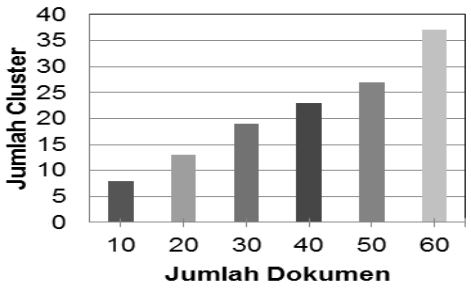
Percobaan dilakukan terhadap 6 data uji coba dengan jumlah dokumen yang berbeda yaitu 10 dokumen dalam 5 kategori berbeda, 20, 30, 40 dan 50 dokumen dalam 10 kategori berbeda serta 60 dokumen dalam 12 kategori berbeda. Setiap data diujicobakan menggunakan nilai *threshold* 0,1 sampai 0,9.

Berdasarkan hasil percobaan terhadap data uji yang bervariasi yang ditunjukan pada Gambar 3, nilai *threshold* yang menghasilkan *F-Measure* tertinggi pada semua data uji coba adalah 0.2 dan nilai *F-Measure* paling tinggi yaitu 0.90 pada data uji 20 dokumen dengan jumlah *cluster* yang ditunjukan pada Gambar 4.

Hasil pengujian pada Tabel 1, menunjukkan bahwa dokumen ke-34, ke-38 dan ke-40 adalah dokumen yang *overlapping cluster* karena berada pada lebih dari satu *cluster*. Dokumen ke-34 berada pada *cluster* 18,19 dan 20, dokumen ke-38 dan ke-40 berada pada *cluster* 22 dan 23.




Gambar 3. Perbandingan *F-measure* masing-masing *threshold* pada jumlah dokumen berbeda.




Gambar 4. Jumlah *cluster* yang diperoleh pada setiap data uji coba dengan nilai *threshold* 0,2.

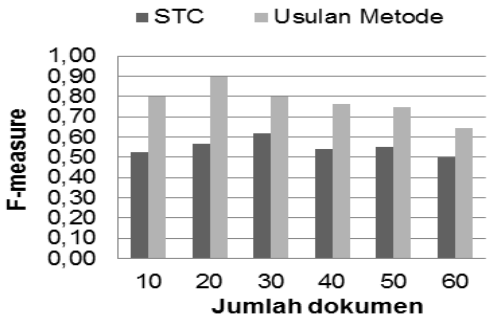
Uji coba untuk perbandingan hasil pengelompokan dokumen menggunakan metode yang diusulkan dan metode STC tradisional dilakukan terhadap 6 data uji coba di atas. Gambar 5 menunjukan bahwa hasil perhitungan *F-Measure* pada setiap data uji coba menggunakan metode yang diusulkan mampu meningkatkan nilai *F-measure* dari metode STC tradisional. Nilai rata-rata dari hasil yang diperoleh dengan menggunakan metode STC tradisional adalah 0,55. Rata-rata dari hasil yang diperoleh dengan menggunakan metode usulan adalah 0,78. Hasil nilai *F-Measure* yang diperoleh dengan menggunakan metode usulan mengalami peningkatan rata-rata 22%.

Berdasarkan dari uji coba, nilai *threshold* yang berbeda menghasilkan *F-measure* yang berbeda. Hal ini menunjukkan bagaimana nilai *threshold* (*t* pada algoritma pengelompokan dokumen Bab 2) mempengaruhi kualitas hasil pengelompokan. Dari percobaan yang dilakukan,nilai *threshold* 0,2 menghasilkan *F-measure* yang paling tinggi. Untuk itu, nilai *threshold* 0,2 digunakan untuk pengelompokan dokumen menggunakan metode yang diusulkan.

Hasil penelitian menunjukan dengan melakukan efisiensi *phrase* berdasarkan proses pemilihan *base cluster* yang terlabeli pada *internal node* dan *leaf node* yang memiliki anak serta melakukan perhitungan *term frequency* dan *document frequency* pada *base cluster* yang terpilih mampu meningkatkan akurasi pengelompokan dokumen.

Tabel 1. Hasil Pengelompokan 50 Dokumen

| Cluster | Dokumen |
|---|---|
| 1 | 1,2,3,4,5 |
| 2 | 6,7,8,9,10 |
| 3 | 11,12 |
| 4 | 13,14 |
| 5 | 15 |
| 6 | 16 |
| 7 | 17 |
| 8 | 18 |
| 9 | 19 |
| 10 | 20 |
| 11 | 21 |
| 12 | 22,24 |
| 13 | 23 |
| 14 | 25 |
| 15 | 26,27,30 |
| 16 | 28 |
| 17 | 29 |
| 18 | 31,34 |
| 19 | 32,34 |
| 20 | 33,34 |
| 21 | 35 |
| 22 | 36,38,39,40 |
| 23 | 37,38,40 |
| 24 | 41,43,44,45 |
| 25 | 42 |
| 26 | 46,47,48,50 |
| 27 | 49 |




Gambar 5. Perbandingan *F-measure* STC tradisional dengan usulan metode pada jumlah dokumen berbeda.

Perbandingan dengan metode STC menunjukkan bahwa usulan metode memperoleh *F-measure* lebih tinggi pada semua uji coba dan mampu meningkatkan nilai *F-measure* rata-rata 22%. Hasil uji coba juga menunjukan bahwa metode yang diusulkan mampu mempertahankan pengelompokan dokumen yang *overlapping cluster*.

Hasil penelitian ini mendukung hasil penelitian-penelitian sebelumnya yang melakukan efisiensi *phrase* berdasarkan perhitungan *term frequency* dan *document frequency* pada proses *scoring base cluster*. (Chim & Deng, 2008).

## KESIMPULAN

Pada makalah ini disajikan metode baru untuk mengelompokan dokumen web berbahasa Indonesia berdasarkan efisiensi *phrase* pada proses pemilihan *base cluster* dengan kombinasi perhitungan *document frequency* dan *term frequency* pada *phrase suffix tree* dengan algoritma *Single Pass Clustering* (SPC). Hasil uji coba menunjukkan metode yang diusulkan menghasilkan nilai *F-Measure* yang lebih tinggi dibandingkan dengan STC tradisional. Hal ini menunjukan bahwa efisiensi *phrase* pada proses pemilihan *base cluster* dengan kombinasi perhitungan *term frequency* dan *document frequnecy* mampu meningkatkan hasil pengelompokan dokumen yang semula menggunakan *scoring phrase* dengan kombinasi *document frequency* dan panjang *phrase* yang terlabeli pada STC tradisional. Metode yang diusulkan juga berhasil mempertahankan *overlapping cluster* yang merupakan kelebihan STC.

Penelitian selanjutnya adalah meningkatkan kinerja dengan mengembangkan metode efisiensi pada *base cluster*, melakukan proses reduksi *phrase* setelah *generate suffix tree* atau melakukan pengembangan algoritma untuk proses *document cleaning* agar *phrase* yang terbentuk benar-benar mewakili dokumen yang akan dikelompokan.

## DAFTAR PUSTAKA

Arifin, A.Z., Darwanto, R., Navastara, D.A. & Ciptaningtyas, H.T. Klasifikasi Online Berita dengan Menggunakan Algoritma Suffix Tree Clustering. Proceeding of SESINDO. 2008.

Arifin, A.Z. & Novan, A.N. Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algorit-ma Single Pass Clustering. Prosiding Seminar on Intelligent Technology and its Applications (SITIA), Teknik

Elektro, Institut Teknologi Sepuluh Nopember Surabaya. 2002.

Chim, H. & Deng, X. Efficient Phrase-Based Document Similarity for Clustering. IEEE Transactions on Knowledge and Data Engineering. Vol. 20: 1217–1229. 2008.

Februariyanti, H. & Zuliarso, E. Algoritma Single Pass Clustering untuk Klastering Halaman Web. Prosiding Seminar Nasional Komputer dan Elektro (SENOPUTRO). 1–8. 2012.

Hammouda, K.M. & Kamel, M.S. Efficient Phrase-Based Document Indexing for Web Document Clustering. IEEE Transactions on Knowledge and Data Engineering. Vol. 16: 1279–1296. 2004.

Huang, C., Yin, J. & Hou, F. Text clustering using a suffix tree similarity measure. Journal of Computers. Vol. 6: 2180–2186. 2011.

Jain, A.K. & Maheshwari, S. Phrase based Clustering Scheme of Suffix Tree Document Clustering Model. International Journal of Computer Application. Vol. 63: 30–37. 2013.

Klampanos, I.A., Jose, J.M. & van Rijsbergen, C.J. Single-Pass Clustering for Peer-to-Peer Information Retrieval : The Effect of Document Ordering. Proceedings of the 1st international conferen-ce on Scalable information sys-tems. 2006.

Zamir, O. & Etzioni, O. Web document clustering: A feasibility demonstration. Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval. 46–54. 1998.