STUDI KINERJA FUNGSI-FUNGSI JARAK DAN SIMILARITAS DALAM *CLUSTERING* DOKUMEN TEKS BERBAHASA INDONESIA

Amir Hamzah[1)], F. Soesianto[2)] , Adhi Susanto[2)] , dan Jazi Eko Istiyanto[3)],

[1)]*Jurusan Teknik Informatika, Institut Sains & Teknologi AKPRIND Yogyakarta*
Jl. Kali Sahak No.28, Komplek Balapan Yogyakarta 55222 Tlp(0274)563029
E-mail : *amir@akprind.ac.id, miramzah@yahoo.co.id*
[2)]*Jurusan Teknik Elektro, Fakultas Teknik, Universitas Gadjah Mada*
Jl. Grafika 2 Yogyakarta, Phone: (62)-274-510983,902200,902201
[2] fhsoes@mti.gadjahmada.edu
[3)]*Jurusan Fisika, Fakultas MIPA, Universitas Gadjah Mada*
Sekip Utara, Phone: (0274) 513339, Yogyakarta, Indonesia 555281
jazi@ugm.ac.id

Abstrak

Clustering dokumen teks banyak diteliti karena peranan pentingnya dalam bidang text-mining dan information retrieval. Dalam algoritma clustering pemilihan fungsi jarak atau fungsi similaritas antar objek menjadi kunci keberhasilan algoritma. Pada fungsi jarak, jarak euclidean paling sering digunakan. Fungsi ini memiliki kelemahan jika digunakan untuk vektor berdimensi sangat tinggi yang menyebabkan kinerja clustering menurun. Alternatif dari fungsi jarak adalah fungsi similaritas, antara lain jaccard, dice, cosine dan pearson. Penelitian ini melakukan kajian tentang unjuk kerja fungsi jarak euclidean dengan empat fungsi similaritas tersebut di atas jika diterapkan untuk melakukan clustering dokumen teks berbahasa Indonesia. Dua pendekatan clustering yang dicobakan adalah pendekatan hierarchi dan partisi. Untuk pendekatan hierachi digunakan teknik aglomeratif dengan 2 metode similaritas cluster yaitu GroupAverage dan CompleteLink. Untuk pendekatan partisi juga dicobakan 2 metode, yaitu Bisecting K-Mean dan Buckshot. Koleksi dokumen yang digunakan 12 koleksi dokumen teks berita, yaitu dengan cacah dokumen 50, 100, 200, 300, 400, 500, 600, 700, 800, 1009, 1270 dan 1370 dokumen. Semua koleksi telah dilakukan clustering secara manual. Kriteria kinerja clustering diukur berdasarkan waktu komputasi dan validitas clustering. Untuk validitas digunakan nilai F-measure, yaitu nilai yang diturunkan dari Recall dan Precision yang mengukur kemampuan algoritma melakukan klasifikasi secara benar. Hasil penelitian menunjukkan bahwa hasil clustering terbaik adalah jika digunakan fungsi Cosine dengan rata-rata F-measure untuk seluruh koleksi 0,9313; sementara yang terburuk adalah jika digunakan fungsi jarak eucledian dengan rata-rata F-measure 0,4668. Secara waktu komputasi fungsi cosine juga memiliki kinerja tercepat dengan rata-rata 12,9 detik sedangkan terjelek adalah pearson dengan rata-rata 58,2 detik.

Kata Kunci: clustering dokumen, validitas clustering, fungsi similaritas

1. PENDAHULUAN

Document Clustering banyak diteliti karena peranan pentingnya dalam bidang *text-mining* dan *information retrieval*. Dalam teknik *clustering* berbasis *feature* kata dengan model ruang vektor setiap objek dokumen diandaikan sebagai vektor dimensi tinggi dengan kata-kata yang muncul dalam koleksi dokumen dianggap sebagai absis dalam ruang vektor tersebut. Dalam kajian *clustering* objek secara umum ukuran kedekatan antar vektor objek yang dikluster lebih sering digunakan ukuran jarak, yaitu jarak *euclidean*. Ukuran ini mengasumsikan bahwa antar sumbu koordinat dalam ruang vektor adalah saling bebas. Dalam vektor dokumen dimana koordinat adalah kata yang diekstrak dari koleksi dokumen asumsi ini sebenarnya sulit dipenuhi karena biasanya dalam dokumen selalu ada kata yang kemunculannya tergantung pada kata yang lain. Untuk mengatasi hal ini ukuran kedekatan dokumen dapat diukur dengan fungsi lain, yaitu fungsi similaritas. Beberapa fungsi similaritas yang biasa digunakan antara lain adalah fungsi *jaccard*, *dice*, *cosine* dan *pearson*. Perbandingan kinerja antar fungsi-fungsi similaritas ini dan perbandingannya dengan fungsi jarak *euclidean* jarang dilakukan. Pada sisi lain fungsi jarak dan similaritas ini merupakan persoalan paling krusial dalam kinerja *clustering*. Untuk itulah dalam penelitian ini akan dilakukan upaya perbandingan kinerja fungsi-fungsi tersebut dalam *clustering* dokumen teks. Penelitian ini mencari fungsi mana yang paling efektif dan efisien secara komputasional. Ukuran efektivitas diukur dengan F-*measure*, yaitu nilai yang diukur dari *Recall* dan *Precision*, sedangkan efisiensi diukur dengan waktu komputasi yang diperlukan untuk menyelesaiakn proses *clustering*.

2. TINJAUAN PUSTAKA

2.1 Model ruang vektor

Model ruang vektor untuk koleksi dokumen mengandaikan dokumen sebagai sebuah vektor dalam ruang kata (*feature*) (Rijsbergen, 1979). *Clustering* dokumen dipandang sebagai pengelompokan vektor berdasarkan suatu fungsi jarak atau *similarity* antar dua vektor tersebut. Jika koleksi n buah dokumen dapat diindeks oleh t buah *term*/*feature* maka suatu dokumen dapat dipandang sebagai vektor berdimensi t dalam ruang term tersebut. Dengan demikian koleksi dokumen dapat dituliskan sebagai matrik kata-dokumen X, yang dapat ditulis :

$$X = \{x_{ij}\}\ i= 1,2,..t\ ;\ j =1,2,.. N \quad (1)$$

x_{ij} adalah bobot term i dalam dokumen ke j

Proses menyusunan matrik kata-dokumen (sering disebut tahap pre-*processing*) adalah sebagai berikut: tahap awal adalah pengubahan ekspresi kata ke *lower-case* dan penghilangan *stop-word*, seperti artikel atau preposisi misalnya 'ini','itu','yang', 'yaitu' dan lain-lain. Penghilangan *stop-word* ini dapat mengurangi frekuensi *feature* 30 sampai 40 persen (Rijsbergen,1979). Proses leksikal yang lain terhadap *feature* kata adalah proses *stemming*, yang akan mereduksi semua kata ke dalam akar katanya. Algoritma *stemming* yang sudah luas diterapkan dalam sistem IR adalah algoritma yang dibangun oleh Porter (1987), biasa disebut dengan *Porter Stemmer*. Algoritma ini telah dimodifikasi ke dalam berbagai bahasa. Modifikasi untuk bahasa Indonesia dilakukan oleh Tala(2004). Review detail tentang *stemmer* bahasa Indonesia telah dilakukan oleh Asian(2005), sedangkan efek *stemming* pada clustering dokumen bahasa Indonesia dilakuan oleh Hamzah (2006).

Untuk meningkatkan kemampuan *term* sebagai pembeda dokumen pembobotan atas *term* perlu dilakukan. Pembobotan dasar dilakukan dengan menghitung frekuensi kemunculan *term* dalam dokumen karena dipercaya bahwa frekuensi kemunculan *term* merupakan petunjuk sejauh mana *term* tersebut mewakili isi dokumen. Menurut Luhn (1958), kekuatan pembeda terkait dengan frekuensi term (*term-frequency*, *tf*), di mana *term* yang memiliki kekuatan diskriminasi adalah *term* dengan frekuensi sedang. Untuk itu pemotongan frekuensi bawah biasanya ditempuh dengan memberikan *treshold* tertentu untuk minimal jumlah dokumen yang memuat term tersebut. Pemotongan term dengan frekuensi tinggi dilakukan dengan membuang *stop-word*.

2.2 Pembobotan *Term* (*term – weighting*)

Penggunaan hanya frekuensi *term* dalam dokumen sebagai bobot *term* tersebut dalam representasi dokumen tidaklah memadai. Hal ini karena bias dapat muncul dari faktor lain, misalnya banyaknya dokumen yang memuat term tersebut, atau faktor panjang dokumen dimana term tersebut muncul (Rijsbergen, 1979). Faktor panjang dokumen dalam koleksi berakibat seolah-olah *term* yang sering muncul pada dokumen panjang lebih penting dari pada *term* yang kurang sering muncul pada dokumen pendek. Untuk itu normalisasi frekuensi *term* terhadap panjang dokumen diperlukan. Secara umum bentuk pembobotan akhir *term* dapat dirangkumkan sebagai berikut (Chisholm et.al.,1999) :

$$w_{ij}=L_{ij}.G_i.N_j \quad (2)$$

di mana w_{ij} adalah akhir bobot total term i dalam dokumen ke j, L_{ij} adalah bobot lokal term i dalam dokumen ke j yang mengukur seberapa penting peranan term i dalam dokumen j , G_i bobot global term i yang mengukur seberapa penting term i dalam seluruh koleksi dokumen, dan N_j adalah faktor normalisasi untuk dokumen ke j untuk menghilangkan pengaruh bias karena panjang dokumen. Berbagai variasi pembobotan lokal, global dan normalisasi yang dapat diterapkan pada vektor dokumen dirangkumkan dalam Chisholm et.al. (1999). Kombinasi terbaik yang sering digunakan adalah f_{ij} untuk bobot lokal L_{ij} (disebut TF), dan $\log\left(\frac{N}{n_i}\right)$ sebagai bobot global G_i (disebut IDF) dan pembobotan normal sehingga panjang vektor adalah satu, yaitu :

$$N_j= \frac{1}{\sqrt{\sum_{i=0}^{m}\left(G_i L_{ij}\right)^2}} \quad (3)$$

Sehingga bentuk akhir disebut sebagai pembobotan TF-IDF ternormalisasi, yaitu :

$$w_{ij}= \frac{(\log(f_{ij}+1).\log\left(\frac{N}{n_i}\right)}{\sqrt{\sum_{i=1}^{t}\left((\log(f_{ij}+1).\log\left(\frac{N}{n_i}\right)\right)^2}} \quad (4)$$

di mana w_{ij} adalah akhir bobot total term i dalam dokumen ke j, f_{ij} adalah frekuensi kata ke-i dalam dokumen ke-j, N cacah dokumen dalam koleksi, n_i cacah dokumen mengandung term i dan t adalah cacah total *term*.

2.3 Fungsi jarak dan similaritas antar vektor dokumen

Kesamaan antara dokumen D_i dengan dokumen D_j dapat diukur dengan fungsi similaritas (mengukur kesamaan) atau fungsi jarak (mengukura ketidaksamaan). Beberapa fungsi similaritas dan fungsi jarak yang dapat dijumpai antara lain adalah *Dice, Jaccard, Euclidean distance, Pearson Correlation* dan *Cosine-similarity* (Tombros, 2002). Berikut ini formula fungsi-fungsi tersebut :

Dice: $\text{sim}(D_i,D_j) = \frac{2\sum_{k=1}^{d} D_{ik}D_{jk}}{\sum_{k=1}^{d} D_{ik} + \sum_{k=1}^{d} D_{jk}}$ (5)

Jaccard : $\text{sim}(D_i,D_j) = \frac{2\sum_{k=1}^{d} D_{ik}D_{jk}}{\sum_{i=1}^{d} D_{ik} + \sum_{k=1}^{d} D_{jk} - \sum_{k=1}^{d} D_{ik}D_{jk}}$ (6)

Eucledian distance: $\text{dis}(D_i,D_j) = \sqrt{\sum_{k=1}^{d}(D_{ik} - D_{jk})^2}$ (7)

Cosine : $\text{sim}(D_i,D_j) = \frac{\sum_{i=1}^{d} D_i D_j}{\sqrt{\sum_{k=1}^{d}(D_{ik})^2 \sum_{k=1}^{d}(D_{jk})^2}}$ (8)

Pearson Correlation : $\text{sim}(D_i,D_j) = \frac{1}{2}\left(\frac{(D_i - \overline{D}_i)^T (D_j - \overline{D}_j)}{\|D_i - \overline{D}_i\|_2 \|D_j - \overline{D}_j\|_2} + 1\right)$ (9)

Menurut Strehl et.al. (2000) untuk tujuan *clustering* dokumen jarak fungsi yang paling baik adalah fungsi similaritas *Cosine*. Akan tetapi dibandingkan dengan fungsi similaritas yang lain, seperti *jaccard*, *dice* dan pearson, kelebihan cosine ini belum dicobakan dan dielaborasi. Secara formula fo\ungsi cosine memiliki keuntungan untuk efisiensi komputasi, yaitu apabila vektor dokumen dtarnsformasikan ke dalam bentuk vektor satuan, maka sehingga $\|Di\|_2=1$ dan $\|Dj\|_2=1$ maka fungsi similaritas cosine menjadi sederhana, yaitu sekedar perkalian antar vektor, yaitu :

Cosine-$\text{sim}(D_i,D_j) = \sum_{i=1}^{t} D_i D_j$ (10)

2.4 Algoritma *Clustering* Dokumen

Dalam model ruang vektor dikenal dua pendekatan algoritma *clustering*, yaitu *hierarchi* dan *partisi* (Jain,1988). Dari penerapan *clustering* yang luas pada bidang sains dan teknik salah satu penerapannya adalah untuk *clustering* dokumen. Dari algoritma hierarchi ada dua pendekatan, yaitu divisive dan aglomerative. Dari dua pendekatan tersebut, pendekatan agglomerative lebih banyak diteliti untuk clustering dokumen.

Metode *Hierarchi Agglomerative* untuk *Clustering* dokumen

Berikut ini algoritma dasar klustering secara *aglomerative*, dengan menggunakan notasi $\hat{c}$ = himpunan cluster, N = cacah objek dan c = cacah cluster yang akan dibuat:

[1]. Andaikan $\hat{c}$ =N ; himpunan objek C={x_i }, i=1,2,…n
[2]. Jika $|\hat{c}| \leq c$ stop
[3]. Temukan dua kluster terdekat : C_i dan C_j
[4]. gabungkan C_i dan C_j, hapus C_j dan kurangi $|\hat{c}|$ dengan satu
[5]. 5. Pergi ke langkah 2

Tahap paling krusial yaitu langkah 3, tahap penggabungan kluster yang ditentukan dengan beberapa ukuran similaritas antar kluster. Beberapa metode penggabungan kluster antara lain : *UPGMA, CST, IST, Single Link* dan *Complete Link* (Jain,1988). Menurut Hamzah dkk (2007) untuk clustering dokumen teks metode yang terbaik adalah UPGMA dan Complete Link. Berikut ini ringkasan masing-masing teknik tersebut:

- *Unweighted Pair Group Method Average similarity* (UPGMA): Similaritas dua kluster diukur dengan rata-rata hitung similaritas antar seluruh pasangan titik antara kedua kluster.
- *Complete Link(CL)* : jarak terbaik dua kluster diwakili oleh jarak terjauh (similaritas terendah) dari dua titik dari dua kluster.

K-Means Clustering

Algoritma K-*means clustering* merupakan algortima iteratif dengan meminimalkan jumlah kuadrat *error* antara vektor objek dengan pusat kluster terdekatnya (Jain, 1988). Algoritma K-*means standard* dapat dituliskan sebagai :

[1]. Ambil K objek sebagai *seed* dari K pusat kluster
[2]. Untuk semua objek: cari kluster dengan jarak terdekat, dan tetapkan objek masuk dalam kluster tersebut.
[3]. Hitung ulang pusat kluster dengan rata-rata objek dalam kluster tersebut
[4]. Hitung fungsi kriteria dan lakukan evaluasi. Jika fungsi kriteria berubah cukup kecil algoritma berhenti.

Bisecting K-Means Clustering

Metode *Bisecting K-means* Steinbach et.al. (2000) mencoba menggabungkan pendekatan *partitional* dengan *divisive hierarchical*, yaitu mula-mula seluruh dokumen dibagi dua dengan cara *K-means* (*bisecting-step*). Selanjutnya cara itu dikenakan pada suatu kluster yang dipilih dengan cara tertentu sampai diperoleh K buah kluster. Berikut ini algoritmanya :

[1]. Ambil satu kluster untuk displit dengan K-means (*bisecting step*)
[2]. Ulangi langkah [1] sebanyak ITER kali, dan ambil hasil terbaik yang memiliki *overal similarity* terbesar.
[3]. Ulangi langkah [1] dan [2] sampai didapatkan K buah kluster.

Overall similarity pada langkah [2] ditentukan sebagai rata-rata similaritas setiap titik terhadap pusat klusternya masing-masing. Sedangkan pemilihan kluster pada langkah satu pada setiap iterasinya dapat digunakan cara memilih kluster dengan ukuran cacah objek terbesar atau memilih kluster yang memiliki *variance* terbesar.

Buckshot Clustering

Algoritma *Buckshot* menggunakan pendekatan *hierarchie agglomerative* untuk mendapatkan k buah vektor sebagai pusat kluster awal (Cutting et.al.,1992). Langkah *Buckshot* mula-mula mengambil sampel acak sebesar $\sqrt{kn}$ buah dokumen, yang dikluster dengan *cluster subroutine*, yaitu prosedur *hierarchie agglomerative* untuk mendapatkan k buah kluster. Selanjutnya dengan partisi awal yang didapat dari *Buckshot* proses *refinement* dilakukan sebagaimana dalam K-*means clustering*.

2.5 Parameter kinerja *clustering*

Parameter kinerja pada studi *clustering* terdiri dari dua parameter, yaitu parameter yang mengukur sejauh mana algoritma dapat menyerupai *clustering* yang dilakukan secara manual oleh manusia ('ahli') dan waktu yang diperlukan untuk melakukan *clustering*. Untuk penelitian ini 'ahli' telah dibentuk tim kecil terdiri 3

dosen (termasuk penulis) dan 2 mahasiswa untuk melakukan *clustering* secara manual seluruh koleksi dokumen yang dicobakan. Untuk kinerja yang pertama parameter yang digunakan adalah dengan menghitung F-*measure* yang diturunkan dari tabel *confusion-matrix* seperti Tabel 1 berikut :

Tabel 1. Hubungan Nilai *Actual-Class* dan *Predicted*

Manual (i)	**Algoritma Clustering (j)**		Total Per Class
Class (i)	**Custer-1**	*Cluster-2*	
Cass-1	n_{11}	n_{12}	n_i , i=1
Class-2	n_{21}	n_{22}	ni, i=2
Total Per Cluster	nj, j=1	nj , j=2	n

Dari tabel diturunkan nilai R dan P, yakni jika j mewakili *clustering* oleh algoritma dan i mewakili klasifikasi manual, dapat ditentukan :

$$R = Recall(i , j) = n_{ij}/n_i \quad (11)$$
$$P = Precision(i , j) = n_{ij}/n_j \quad (12)$$
$$F(i) = 2PR/(P+R) \quad (13)$$

dengan F(i) diambil nilai terbesar dari setiap kluster untuk *class* i. F-*measure* keseluruhan *cluster* hasil *clustering* adalah :

$$\text{F-}measure = \frac{\sum_i n_i x F(i)}{n} \quad (14)$$

3. METODE PENELITIAN

Bahan dari penelitian ini adalah koleksi dokumen teks berita yang didownload dari beberapa sumber *news* online di internet, yaitu Tempo online , Kompas online dan e-library.com. Koleksi yang dokumen asalnya berupa file-file html diformat ulang dengan membuang tag-tag html dan menggabungkannya menjadi satu file. Setiap koleksi disimpan dalam satu file yang terdiri dari dokumen-dokumen yang sudah murni berupa dokumen teks. Setiap dokumen yang satu dengan dokumen lain dipisahkan dengan tag <DOC>..</DOC>. Adapun format setiap dokumen adakah seperti Gambar 1. Statistik koleksi berita yang digunakan dalam penelitian ini secara lengkap disajikan seperti pada Tabel 1.

Beberapa perangkat lunak sebagai alat yang digunakan pada penelitian ini adalah sebagai berikut.

a) Komputasi dilakukan dengan menggunakan komputer PC Intel Pentium IV 2.8GHz, RAM 1GB, Hard Disk 80 GB, dan sistem operasi Windows XP Professional.
b) Bahasa pemrograman yang dipergunakan adalah Borland C++ Builder 5.0, java jdk1.4.1_2, dan Matlab versi 7.0.4

```
<DOC>
<DOCNO>news10513-html</DOCNO>
mayjen syafrie samsuddin akan jadi kapuspen tni
jakarta  media   mantan pangdam jaya mayjen syafrie
samsuddin akan menjadi kapuspen tni menggantikan
marsekal muda graito husodo  menurut informasi yang
diperoleh  antara jakarta kamis  syafrie samsuddin
menjadi kapuspen tni dan serah terima jabatan akan
dilakukan pada akhir februari 2002  namun kebenaran
informasi tersebut hingga kini belum dapat
dikonfirmasikan ke kapuspen tni    m-1
</DOC>
```

Gambar 1. Format dokumen berita

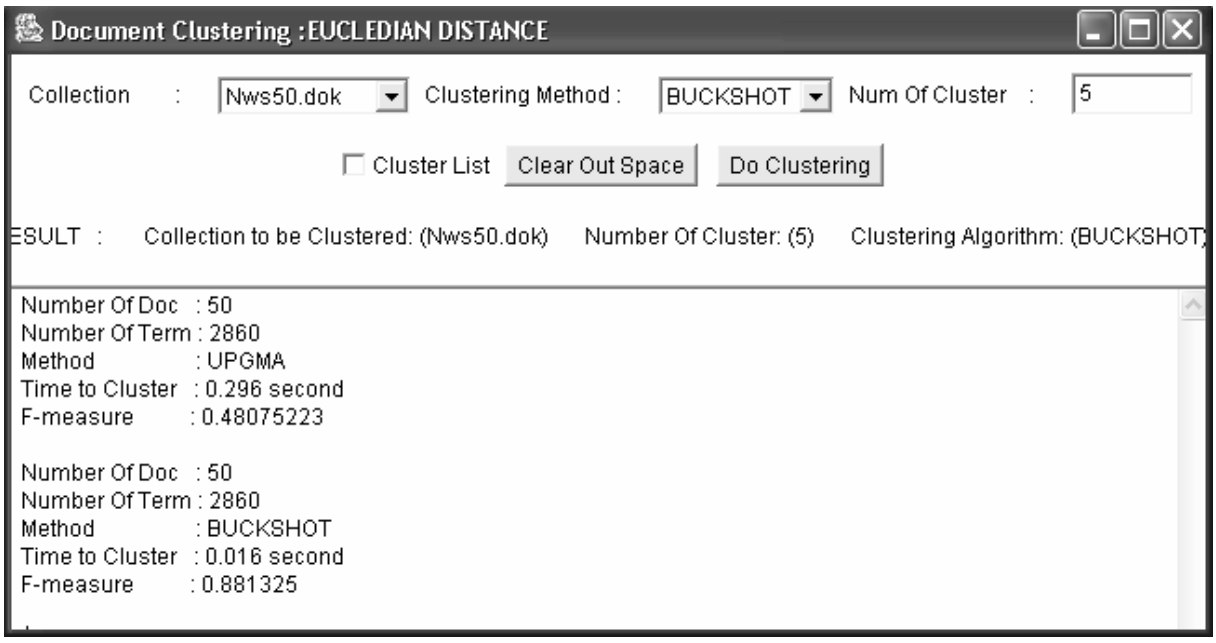
Tabel 2. Nama Koleksi dokumen teks berita

Nama Koleksi	Cacah dok	Cacah cluster	Ukuran Cluster	Cch Kata Unik	Min Cch kata/dok	Max Cch kata/ dok	Rerata juml kata/ dok
Nws50	50	5	Sama	2.860	167	860	354
Nws100	100	10	Sama	4.385	171	866	368
Nws200	200	10	Sama	6.634	110	902	372
Nws300	300	10	Beda	8.471	110	954	373
Nws400	400	11	Beda	10.152	110	1.072	388
Nws500	500	13	Beda	11.636	110	1.072	385
Nws600	600	13	Beda	13.432	97	1.849	388
Nws700	700	13	Beda	14.800	97	1.849	385
Nws800	800	14	Beda	15.751	97	1.849	410
Nws1009	1009	21	Beda	18.259	97	1.849	425
Nws1270	1270	25	Beda	22.431	97	2.892	419
Nws1370	1370	25	Beda	23.398	97	2.892	411

Program *clustering* dan evaluasi *clustering* dikodekan dengan java (jdk1.4.1_2). Pada seluruh koleksi dilakukan *clustering* dengan berbagai algoritma *clustering*. Setiap algoritma dikodekan dengan 5 macam fungsi, 1 fungsi jarak eucledian dan 4 fungsi similaritas. Hasil kinerja *clustering* dibandingkan dengan membandingkan nilai F-*measure*nya.

4. HASIL DAN PEMBAHASAN

Hasil perancangan antar muka untuk proses clustering dengan berbagai macam fungsi similaritas memiliki tampilan seperti pada gambar 2 berikut. Pada tampilan tersebut list dokumen pada tiap-tiap cluster tidak ditampilkan untuk menghemat ruang. Selanjutnya untuk algoritma hiararchi dan partisi masing-masing dilakukan pengujian secara terpisah.



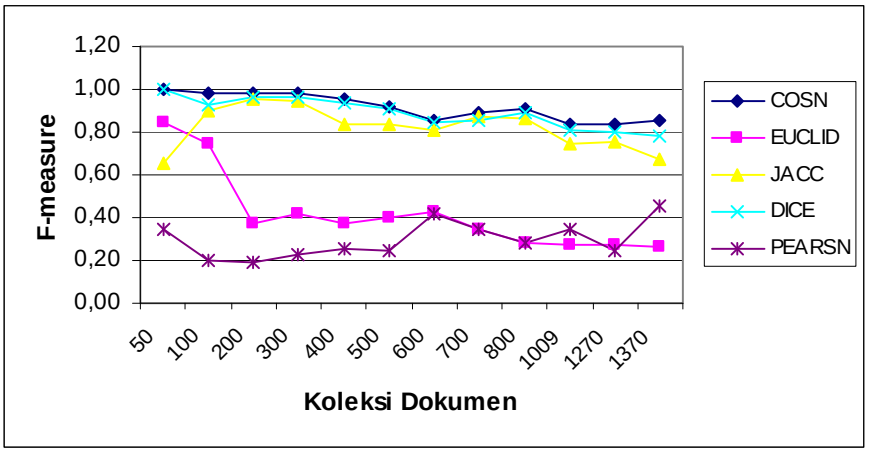
Gambar 2. Perancangan antar muka proses clustering dokumen

4.1 Hasil perbandingan pada algoritma *hierarchi*

Berikut ini hasil perbandingan kinerja fungsi jarak dan similaritas pada algoritma *hierarchi* dengan metode penggabungan kluster UPGMA. Tabel 3 dan gambar 2 menunjukkan hasil kinerja tersebut.

Tabel 3. Nilai F-measure hasil *clustering* dengan metode UPGMA dengan 5 fungsi

Koleksi	COSN	EUCL	JACC	DICE	PEAR
50	1,0000	0,8493	0,6566	1,0000	0,3434
100	0,9856	0,7474	0,9006	0,9256	0,1965
200	0,9848	0,3762	0,9562	0,9625	0,1937
300	0,9834	0,4139	0,9425	0,9654	0,2295
400	0,9545	0,3765	0,8397	0,9354	0,2516
500	0,9139	0,3987	0,8333	0,9078	0,2427
600	0,8581	0,4287	0,8123	0,8421	0,4177
700	0,8873	0,3457	0,8735	0,8521	0,3456
800	0,9109	0,2786	0,8656	0,8917	0,2856
1009	0,8345	0,2761	0,7473	0,8111	0,3453
1270	0,8403	0,2745	0,7573	0,8012	0,2456
1370	0,8509	0,2657	0,6734	0,7824	0,4534
Average	**0,9170**	0,4193	0,8215	**0,8898**	0,2959



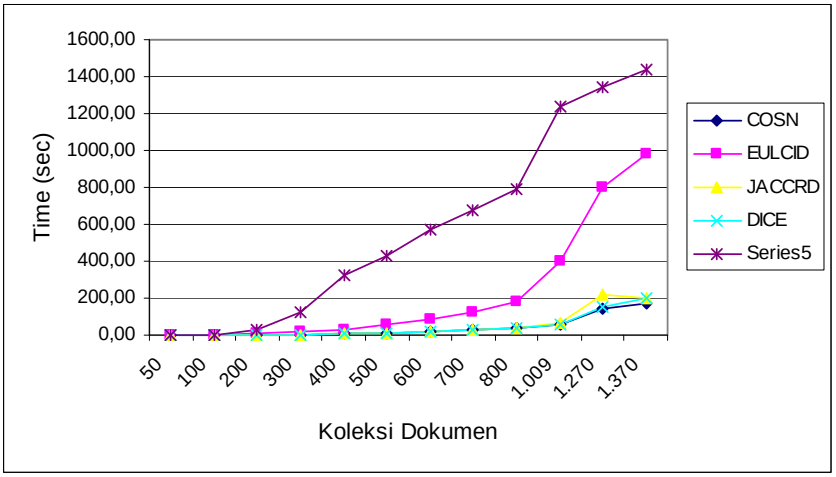
Gambar 3. Nilai F-measure dengan algoritma UPGMA untuk 5 fungsi

Dari tabel 3 terlihat bahwa fungsi similaritas *cosine* memiliki kinerja rata-rata terbaik (0,9170) dan terbaik pada kinerja setiap koleksi seperti tampak dalam grafik gambar 3. Urutan kedua adalah fungsi *dice* (0,8898). Fungsi similaritas dice memiliki formula yang mirip dengan cosine. Fungsi dengan kinerja terjelek adalah similaritas dengan korelasi pearson (0,2959). Untuk penggunaan fungsi jarak *euclidean* kinerjanya, meskipun bukan terjelek, tetapi cukup kecil yaitu rata-rata 0,4193.

Tabel 4. Waktu komputasi (detik) *clustering* dengan metode UPGMA dengan 5 fungsi

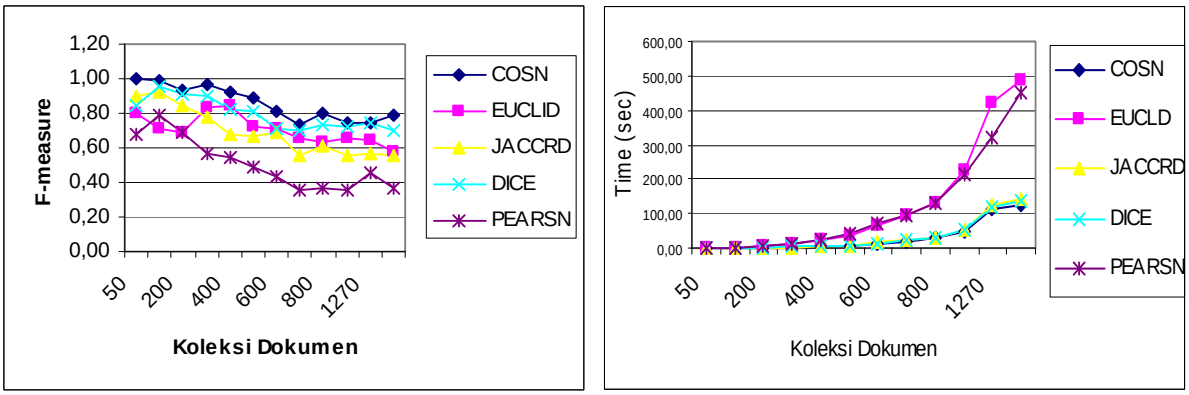
Koleksi	COSS	EUCL	JACC	DICE	PEAR
50	0,031	0,094	0,047	0,110	0,234
100	0,094	0,453	0,110	0,110	2,454
200	0,797	4,812	0,813	0,813	32,600
300	2,657	16,610	2,687	2,790	126,480
400	6,016	32,870	6,328	5,125	327,300
500	9,688	53,311	10,266	9,880	432,560
600	16,297	87,860	16,500	16,890	567,864
700	24,390	125,500	25,560	25,120	678,457
800	34,830	182,500	36,980	34,680	789,456
1.009	59,760	399,960	62,250	61,780	1.234,560
1.270	147,359	799,218	216,453	149,450	1.346,440
1.370	175,797	982,203	198,450	199,567	1.435,300
Average	**39,810**	223,783	48,037	**42,193**	581,142

Secara waktu komputasi pada algoritma UPGMA juga terlihat bahwa fungsi similaritas cosine memiliki efisiensi yang paling tinggi, yaitu rata-rata 39,81 detik. Sedangkan waktu komputasi yang paling buruk adalah pada fungsi pearson yang memiliki rata-rata 581,142 detik. Fungsi jarak euclidean juga memiliki efisiensi yang rendah dibandingkan dengan fungsi similaritas seperti dice dan jaccard. Untuk waktu komputasi dengan algoritma UPGMA bagi 5 fungsi yang ditinjau dapat disajikan dalam tabel 4 dan gambar 4.



Gambar 4. Waktu komputasi (detik) clustering dengan algoritma UPGMA untuk 5 fungsi

Untuk algoritma hierarchi agglomerative menggunakan metode penggabungan cluster Complete Link, hasil kinerja fungsi-fungsi tidak jauh berbeda dengan UPGMA dari segi efektivitas dan efisiensi komputasinya. Grafik F-measure dan grafik waktu komputasi disajikan dalam gambar 5 (a) dan (b).



(a) (b)

Gambar 5. Efektifitas dan efisiensi fungsi jarak dan similaritas pada algoritma Complete Link
(a) Nilai F-measure dari 5 fungsi (b) Waktu komputasi (detik) dari 5 fungsi

4.2 Hasil perbandingan pada algoritma partisi

Pada perbandingan kinerja fungsi similaritas dengan algoritma dengan pendekatan partisi hasilnya memiliki pola yang hampir sama dengan pendekatan hierarchi. Berikut ini hasil perbandingan kinerja fungsi jarak dan similaritas pada algoritma partisi metode Buckshot dan metode Bisecting K-Means. Pada dua pendekatan ini kinerja terbaik juga dicapai jika fungsi similaritas adalah cosine, disusul yang kedua fungsi dice, fungsi Jaccar, euclidean dan yang terburuk hádala fungsi korelasi pearson. Tabel 5 menunjukkan nilai-nilai F-measure untuk metode Buckshot.

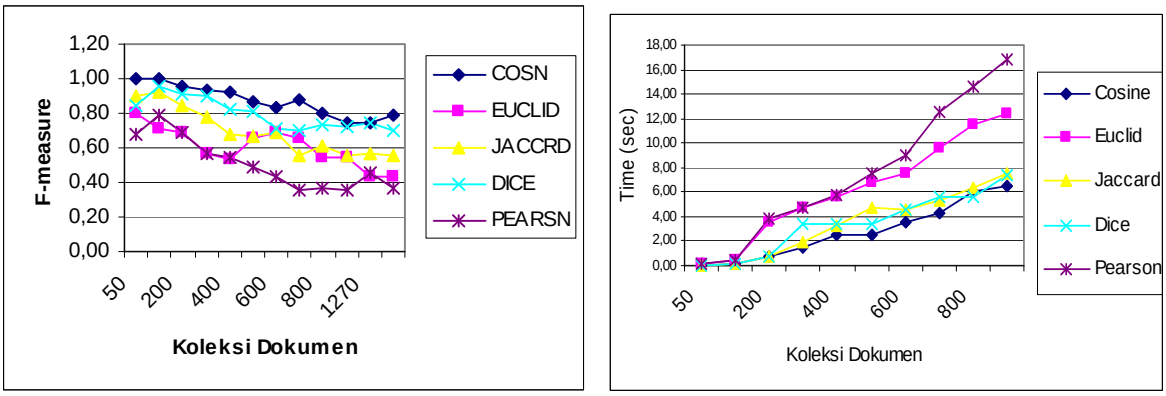
Untuk Bisecting K-mean hasil grafik F-measure dan waktu komputasi dapat dilihat pada gambar 6. Pola ini sama juga dengan pola pada BuckShot , Complete Link dan UPGMA.

Jika dicermati pada formula fungsi similaritas dan fungsi jarak yang dicobakan, fungsi cosine merupakan fungsi yang paling sederhana , yakni jika vektor dokumen dalam keadaan normal sehingga operasi yang dilakukan hádala perkalian vektor saja. Inilah yang menyebabkan fungsi cosine memiliki efisiensi komputasi yang paling baik. Pada fungsi lain seperti korelasi-pearson memiliki formula yang paling kompleks sehingga efisiensi komputasinya paling rendah dan kinerjanya paling buruk.

Untuk fungsi jarak eucliedean juga terlihat memiliki kenerja yang relatif buruk meskipun tidak lebih buruk dari korelasi pearson, tetapi kinerjanya lebih buruk dari fungsi similaritas dice dan jaccard. Kinerja yang jelek dari fungsi euclidean ini hádala karena fungsi jarak ini mengasumsikan independensi yang tinggi antar koordinat vektor dalam ruang vektor. Fungsi jarak euclidean juga terlihat menurun drastis jika dimenai ruang vektor semakin tingi, meskipun pada fungsi lainpun terjadi penurunan kinerja jika dimensi semakin tinggi.

Tabel 5. Nilai F-*measure* hasil *clustering* dengan metode BUCKSHOT dengan 5 fungsi

Koleksi	COSN	EUCL	JACC	DICE	PEAR
50	0,9845	0,7654	0,8974	0,9674	0,4234
100	0,9830	0,7532	0,8976	0,9323	0,3232
200	0,9343	0,5432	0,9562	0,9625	0,2983
300	0,9654	0,4356	0,9425	0,9842	0,2234
400	0,9545	0,4011	0,8754	0,9354	0,3234
500	0,9212	0,3987	0,8854	0,9234	0,3432
600	0,8432	0,4287	0,8123	0,8543	0,4323
700	0,8754	0,4532	0,7867	0,8512	0,3564
800	0,8944	0,4987	0,8123	0,8876	0,3023
1009	0,8023	0,4322	0,7473	0,8233	0,4230
1270	0,8430	0,4678	0,7640	0,8123	0,2987
1370	0,8220	0,4356	0,6734	0,7845	0,3232
Average	**0,9019**	0,5011	0,8375	**0,8932**	0,3392



Gambar 6. Efektifitas dan efisiensi fungsi jarak dan similaritas pada algoritma Bisecting K-Means
(a) Nilai F-measure dari 5 fungsi (b) Waktu komputasi (detik) dari 5 fungsi

5. KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari penelitian ini adalah :

- Pada semua algoritma yang dicobakan, yaitu UPGMA dan Complete pada pendekatan hierarchi dan Buckshot dan Bisecting K-Mean pada pendekatan partisi fungsi similaritas Cosine secara consisten menunjukkan kinerja yang terbaik, baik pada sisi hasil clusteringnya yang memiliki rata-rata F-measure paling tinggi maupun darisisi efisiensi komputasinya yang memiliki rata-rata paling rendah
- Fungsi yang kinerjanya terburuk adalah fungsi similaritas korelasi pearson. Diduga kinerja komputasi yang buruk karena kompleksitas formulanya
- Fungsi similaritas yang lain, yaitu Dice dan Jaccrd kinerjanya mendekati fungsi similaritas cosine, meskipun dari sisi efisiensi komputasinya masih kalah dengan cosine
- Fungsi jarak euclidean memiliki kinerja yang buruk meskipun tidak yang terburuk, baik dari sisi hasil clustering yang dihasilkan maupun dari sisi efisiensi komputasinya.

6. DAFTAR PUSTAKA

Asian, J., H. E. Williams, and S. M. M. Tahaghoghi, 2005, *Stemming Indonesian*, 28th Australian Computer Science Conference (ACS2005).

Chisholm, E. and T. G. Kolda, 1999, "New Term Weighting Formula for the Vector Space Method in Information Retrieval", *Research Report*, Computer Science and Mathematics Division, Oak Ridge National Library, Oak Ridge, TN 3781-6367, March 1999.

Cutting, D. R., D. R. Karger, J. O. Pederson, and J. W. Tukey,1992, *Scatter/Gather:A Cluster-based Approach to Browsing Large Document Collection*, Procedding 15th Annual Int 7ACM SIGIR Conference on R&D in IR, June 1992.

Hamzah, A., Adhi Susanto, F.Soesianto, Jazi Eko Istiyanto, 2007, "Studi Komparasi Algoritma *Hierarchical* Dan *Partitional* Untuk *Clustering* Dokumen Teks Berbahasa Indonesia", Jurnal Terakreditasi , ACADEMIA ISTA Agustus 2007

Jain, A.K. and R. C. Dubes, 2001, *Algorithms for Clustering Data*, Prentice-Hall.

Luhn, H.P. (1958), *The Automatic Creation of Literature Abstracts*. IBM Journal of Research and Development, 2:159-165.

Porter, M. , 1980, An Algorithm for Suffix Stripping, *Program 13(3), 130-137.*

Rijsbergen, C. J.,1979, *Information Retrieval*, Information Retrieval Group, University of Glasgow .

Steinbach, M., G. Karypis, and V. Kumar , 2000, A Comparison of Document Clustering Techniques, KDD Workshop on Text Mining, ***www.citeseer.ist.psu.edu/steincah00comparison.html***

Strehl, A., J. Ghosh, and R. Mooney, 2000, *Impact of Similarity Measures on Web-Page Clustering*, Proceeding of the Workshop of Artificial Intelligent for Web Search, 17th National Conference on Artificial Intelligence, July 2000.

Tala, F. Z., 2004, "A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia", *Master Thesis*, Universiteit van Amsterdam, The Netherlands