PENERAPAN ALGORITMA *INDEX GRAPH* DAN *GROUP-AVERAGE HIERARCHICAL CLUSTERING* DALAM PENGELOMPOKAN JURNAL ILMIAH BERBASIS FRASA

Mohammad Faizal Nugroho[1] , Drs. Achmad Ridok, M.Kom[2] , Djoko Pramono, S.T.[3]
Mahasiswa Teknik Informatika, Universitas Brawijaya
[1]mfaizalnugroho@gmail.com
Dosen Teknik Informatika, Universitas Brawijaya
[2]acridokg@gmail.com
Dosen Teknik Informatika, Universitas Brawijaya
[3]djeqy@yahoo.com

ABSTRAK

Document Clustering merupakan sebuah metode pengelompokan dokumen yang akan membantu untuk menentukan dokumen satu dengan lainnya memiliki keterkaitan atau tidak. Pada kasus ini dokumen yang digunakan adalah dokumen jurnal ilmiah berbahasa Inggris. Namun pada kebanyakan metode *clustering* berbasis pada *Vector Space Model* yang menganalisa berdasar *single-term* (kata tunggal), padahal akan lebih baik jika analisis juga dilakukan terhadap frasa dari suatu dokumen. Salah satu metode untuk menentukan similaritas antar dokumen yang berbasis pada frasa yaitu *Document Index Graph*, algoritma yang menerapkan representasi graf dalam menentukan kesamaan frasa dan proses penghitungan similaritas antar dokumen. Untuk menguji keakuratannya, proses *clustering* akan menerapkan algoritma *group-average HAC*. Suatu metode *clustering* yang mengelompokkan dokumen menjadi suatu hirarki dari kelompok kecil menjadi kelompok besar atau sebaliknya.

Hasil pengujian menunjukkan, sistem yang mengimplementasikan graf untuk pengelompokan dokumen jurnal ilmiah bahasa inggris berbasis frasa mampu meningkatkan akurasi sebesar 20,18% dibandingkan pengolahan yang hanya memperhitungkan kata tunggal, namun dengan konsekuensi waktu komputasi yang lebih lama.

Kata kunci: Jurnal ilmiah, Berbasis frasa, Representasi graph, Group-average HAC, Clustering

1. PENDAHULUAN

Publikasi jurnal ilmiah dalam bentuk *digital* / elektronik banyak dilakukan oleh para peneliti seiring dengan akses internet yang semakin mudah dan berkembang dari tahun ke tahun. Penelitian yang dilaksanakan oleh Björk (2009) mengenai jumlah jurnal ilmiah, menyebutkan bahwa pada tahun 2006 terdapat 1.346.000 jurnal ilmiah di internet. Tentunya, jumlah tersebut akan terus bertambah karena perkembangan ilmu dan teknologi yang tidak akan ada habisnya. Banyaknya jumlah dokumen jurnal ilmiah yang tersebar ini, khususnya di web, harusnya memudahkan pengguna dalam mencari informasi yang dibutuhkan. Namun yang menjadi masalah adalah bagaimana menentukan jurnal satu dengan lainnya memiliki keterkaitan atau tidak. Maka dibutuhkan sebuah metode untuk mengelompokkan dokumen-dokumen jurnal tersebut sehingga memudahkan pengambilan informasi yang sesuai kebutuhan pengguna, yaitu dengan metode *Clustering*.

Clustering merupakan salah satu teknik dalam *data mining*, atau yang lebih spesifik yaitu *text mining*, dimana merupakan salah satu metode untuk menemukan keterkaitan antar dokumen. Secara umum, metode *clustering* dokumen teks mencoba untuk memisahkan dokumen-dokumen menjadi

beberapa kelompok dokumen sesuai dengan kemiripannya dari segi isi [HAM-04].

Pada kebanyakan metode *clustering* berbasis pada *Vector Space Model* yang merepresentasikan dokumen sebagai fitur vektor dari term yang muncul pada semua dokumen. Namun *clustering* dengan metode ini hanya menganalisa berdasar *single-term* (kata tunggal). Padahal memisahkan frasa menjadi kata-kata penyusunnya bisa berakibat makna kata menyimpang jauh dari konteks sebenarnya. Sehingga akan lebih baik jika analisis juga dilakukan terhadap frasa dari suatu dokumen, jadi nantinya kemiripan dari dokumen-dokumen akan dihitung berdasarkan *matching phrase* atau kesamaan frasa.

Basis metode penghitungan similaritas yang akan digunakan pada skripsi ini yaitu *Document Index Graph* (DIG), dimana ada atau tidaknya *matching phrase* yang terbentuk antara dokumen satu dengan lainnya diperhitungkan. Algoritma ini menerapkan representasi graf dalam menentukan *matching phrase* dan proses penghitungan similaritas antar dokumen. Graf yang dibentuk merupakan graf berarah dimana arah tersebut menunjukkan struktur dari kalimat [FAT-09].

Selain penerapan *graph*, dalam skripsi ini juga akan menggunakan algoritma *Hierarchical Agglomerative Clustering* (HAC) untuk proses klusterisasi/pengelompokan. Metode ini mengelompokkan data menggunakan hasil *clustering* yang sebelumnya (*Nested Clustering*). Dalam HAC ini, hasil *clustering* pada level 1 akan dikelompokkan lagi dengan *cluster* yang lain berdasarkan kemiripan yang terdapat pada *cluster* yang dihasilkan pada *clustering* level 1 [PRA-10].

2. TINJAUAN PUSTAKA

2.1 Jurnal Ilmiah

Jurnal adalah terbitan berkala yang berbentuk pamflet berseri berisi bahan yang sangat diminati orang saat diterbitkan. Bila dikaitkan dengan kata ilmiah di belakang kata jurnal berarti terbitan berkala yang berbentuk pamflet yang berisi bahan ilmiah yang sangat diminati orang saat diterbitkan [RIF-95].

2.2 Text Mining

Text Mining merupakan salah satu cabang dari *Data Mining* yang khusus menangani data berupa teks. Definisi *text mining* sendiri yaitu kegiatan berbantuan komputer untuk menemukan dan mengambil informasi baru yang sebelumnya tersembunyi, dengan secara otomatis menyarikan berbagai pesan tekstual dari berbagai sumber [ANO-10]. Salah satu manfaatnya yaitu mendapatkan informasi dengan mengelompokkan sekumpulan data berbentuk teks yang memiliki format tidak terstruktur atau minimal semi terstruktur.

2.3 Text Preprocessing

Tahapan dalam *Preprocessing* meliputi pengubahan semua huruf dalam dokumen menjadi huruf kecil (*case folding*), kemudian *filtering* yaitu tahap pembuangan atau penghapusan kata-kata yang tidak penting, dilanjutkan dengan pemotongan *string* berdasarkan tiap kata yang menyusunnya (*tokenizing*), dan proses *stemming* yaitu mencari kata dasar/*root* dari kata hasil *tokenizing* [HAR-06].

2.4 Frasa

Kata-kata digabungkan untuk membentuk suatu frasa, dan frasa merupakan pola dasar yang akan membentuk sebuah kalimat. Sehingga dapat didefinisikan bahwa frasa adalah sekelompok kata yang berperan sebagai unit tunggal dalam arti maupun tata bahasa, dan tidak memiliki kata kerja [ANO-11]. Khusus untuk penelitian ini, frasa yang dimaksud adalah urutan / pasangan kata pada kalimat yang terdapat dalam sebuah dokumen.

2.5 Index Graph

Index graph merupakan algoritma berbasis graf yang dapat dimanfaatkan untuk menemukan kesamaan frasa (pasangan kata) pada beberapa dokumen [HMO-02]. Sehingga analisa tidak hanya tertuju pada kata tunggal, namun juga pada pasangan kata

yang membangun sebuah kalimat dalam suatu dokumen.

Graf adalah sebuah struktur data yang terdiri dari sekumpulan simpul (*node*) dan sisi (*edge*). Biasanya graf digambarkan sebagai kumpulan titik-titik sebagai simpul yang dihubungkan oleh garis-garis sebagai sisi [FAT-09]. Graf yang dibangun merupakan komponen dari [ERN-09]:

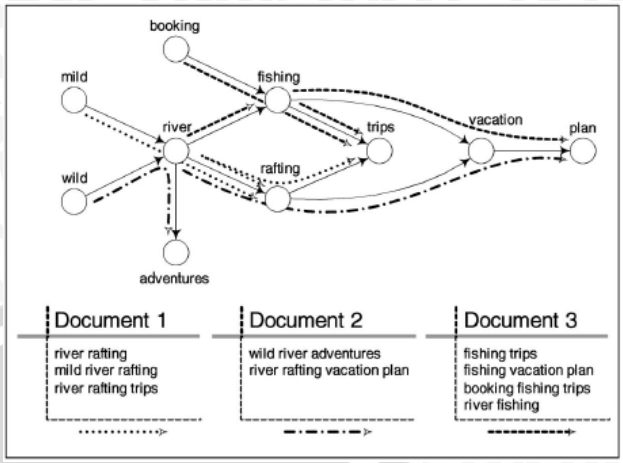
a. Simpul (*Node*): Berisi kata unik dari setiap kalimat dalam dokumen.
b. Sisi (*Edge*): Merupakan penghubung antar simpul. Pada sisi terdapat informasi berupa nomor sisi, posisi kata tersebut dalam kalimat dan dalam dokumen.
c. Lintasan / Jalur (*Path*): Simpul pada graf berisi informasi tentang kata unik dalam sebuah dokumen. Lintasan / jalur yang dibentuk oleh simpul dan sisi merupakan representasi dari kalimat tertentu.

Berdasarkan orientasi arah pada sisi, secara umum graf dapat dibagi menjadi 2 jenis, yaitu graf berarah dan graf tak berarah. Graf berarah merupakan graf dimana setiap sisinya memiliki arah sehingga simpul (1,2) dan (2,1) tidaklah sama. Sebaliknya, pada graf tak berarah, simpul (1,2) dan (2,1) dianggap sama [FAT-09].

Index Graph adalah graf berarah G = (V,E) , dimana [HMO-02]:

a. V : Kumpulan simpul $\{v_1, v_2, \ldots, v_n\}$, tiap v merepresentasikan kata unik dalam seluruh kumpulan dokumen.
b. E : Kumpulan sisi $\{e_1, e_2, \ldots, e_m\}$, sehingga tiap sisi merupakan pasangan simpul terurut (v_i, v_j). Sisi (v_i, v_j) merupakan penghubung simpul v_i ke v_j , dan v_j berdekatan dengan v_i. Akan ada sisi dari v_i ke v_j jika dan hanya jika kata v_j muncul berurutan dengan kata v_i dalam dokumen.

Sebagai ilustrasi dari graf, gambar 2.3 ini menyajikan contoh sederhana dari graf yg mewakili 3 dokumen.



Gambar 2.1 *Document Index Graph*
Sumber: [HAM-04]

Seperti terlihat pada gambar 2.1 , sebuah sisi akan dibuat diantara 2 simpul jika sebuah kata yang direpresentasikan oleh sisi tersebut muncul berurutan di dokumen mana saja. Jadi, kalimat dipetakan dalam bentuk lintasan graf titik-titik untuk dokumen 1, garis-titik untuk dokumen 2, dan garis-garis untuk dokumen 3. Jika suatu frasa muncul lebih dari sekali pada dokumen, frekuensi dari kata tunggal yang menyusun frasa akan meningkat. Kesamaan frasa diantara dokumen-dokumen yang berbeda menjadi sarana untuk menemukan lintasan / jalur yang digunakan secara berulang pada graf.

2.6 *Single Term Similarity*

Term Frequency dan *Inverse Document Frequency* merupakan salah satu metode yang paling banyak digunakan untuk melakukan pembobotan terhadap *term* [FTO-11]. Rumus untuk menghitung nilai tf_{td} ditunjukkan oleh persamaan 2.1 [MAN-07].

$$tf_{td} = \frac{f_{td}}{max_t\{f_{td}\}}, \; f_{td} > 0 \qquad (2.1)$$

dimana:
tf_{td} : Nilai TF *term t* pada dokumen ke-*d*
f_{td} : Frekuensi *term t* pada dokumen ke-*d*
$max_t\{f_{td}\}$: Frekuensi maksimum dari suatu *term* dalam sebuah dokumen

Sedangkan rumus untuk menghitung nilai idf_t ditunjukkan oleh persamaan 2.2 [MAN-07].

$$idf_t = log\left(\frac{N}{df_t}\right),\ df_t > 0 \qquad (2.2)$$

dimana:

idf_t : Nilai IDF dari *term t*
N : Jumlah total dokumen
df_t : Jumlah dokumen yang mengandung *term t*

Setelah kedua nilai *tf* dan *idf* diketahui, maka persamaan 2.3 berikut merupakan gabungan dari persamaan 2.1 dan 2.2 untuk menentukan bobot (*weight*) w_{td} tiap *term* pada tiap dokumen [MAN-07]:

$$w_{td} = tf_{td} * idf_t = tf_{td} * log\left(\frac{N}{df_t}\right) \qquad (2.3)$$

dimana w_{td} merupakan bobot *term t* pada dokumen *d*

Salah satu cara standar untuk menentukan similaritas atau kesamaan antar 2 dokumen d_1 dan d_2 adalah dengan menghitung nilai *cosine similarity*, penjabarannya ditunjukkan pada persamaan 2.4 [MAN-07]:

$$CosSim(\vec{d}_1, \vec{d}_2) = \frac{\vec{d}_1 \bullet \vec{d}_2}{\sqrt{\sum_{t=1}^{n}(\vec{d}_{1t})^2}\sqrt{\sum_{t=1}^{n}(\vec{d}_{2t})^2}}$$

$$= \frac{\sum_{t=1}^{n} w_{t1} \bullet w_{t2}}{\sqrt{\sum_{t=1}^{n}(w_{1t})^2}\sqrt{\sum_{t=1}^{n}(w_{2t})^2}} \qquad (2.4)$$

dimana:

n : jumlah elemen vektor
t : *term* ke-*t*
w_{td} : bobot (*weight*) term *t* pada dokumen *d*

2.7 *Phrase Based Similarity*

Metode ini akan menggunakan frasa sebagai tolok ukur kesamaan dokumen. Persamaan dokumen yang diukur berdasarkan *term* dianggap belum memberikan hasil yang terbaik [HAM-04]. Dengan memperhatikan urutan dari beberapa kata yang terdapat di antara dua dokumen yang sedang dibandingkan diharapkan dapat meningkatkan nilai akurasi pengelompokan dokumen.

Ukuran kesamaan dokumen dihitung berdasarkan frasa sama yang muncul pada masing-masing pasangan dokumen. Faktor-faktor dalam menentukan kesamaan dokumen yaitu [ERN-09]:

a. Jumlah frasa cocok (*P*)
b. Panjang frasa cocok ($l_i : i = 1,2,...,P$)
c. Frekuensi frasa cocok di kedua dokumen (f_{i1} dan $f_{i2} : i = 1,2,...,P$)

Persamaan 2.5 berikut digunakan untuk menghitung similaritas berbasis frasa antara 2 dokumen [HMO-02].

$$Simp(\boldsymbol{d1}, \boldsymbol{d2}) = \frac{\sqrt{\sum_{i=1}^{P}[g(l_i)*(f_{i1}*w_{i1}+f_{i2}*w_{i2})]^2}}{\sum_j |s_{j1}|*w_{j1}+\sum_k |s_{k2}|*w_{k2}} \qquad (2.5)$$

dimana $g(l_i)$ adalah sebuah fungsi untuk menghitung panjang frase cocok, yaitu: $g(l_i) = (|ms_i| / |S_i|)^{\gamma}$. $|ms_i|$ merupakan panjang frase cocok, $|S_i|$ adalah panjang kalimat asli, dan γ adalah faktor fragmentasi kalimat dengan nilai lebih besar sama dengan 1 (satu). $|s_{j1}|$ dan $|s_{k2}|$ merupakan panjang kalimat asli dari dokumen d_1 dan d_2, f_{i1} dan f_{i2} adalah frekuensi frasa cocok di dokumen d_1 dan d_2. Serta w_{i1} dan w_{i2} adalah level signifikansi dari frase cocok di kedua dokumen.

2.8 *Combined Similarity*

Similaritas dokumen akhir dihitung dengan mengombinasikan penghitungan similaritas berbasis kata (*single term*) dengan similaritas berbasis frasa (yaitu persamaan 2.4 dan persamaan 2.5) sehingga didapatkan persamaan 2.6 [HMO-02].

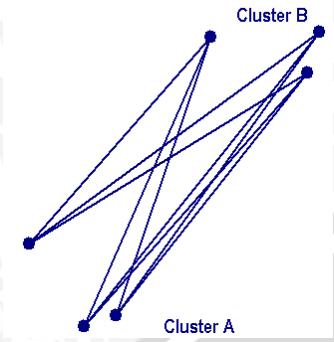
$$Sim(d_1, d_2) = \alpha\ Sim_p(d_1, d_2) + (1-\alpha)\ Sim_t(d_1, d_2) \qquad (2.6)$$

dimana α adalah *Similarity Blend Factor* yang mempunyai nilai dengan interval [0,1] atau $0 \leq \alpha \leq 1$, dimana menentukan bobot penghitungan similaritas berbasis frasa. Sim_p merupakan nilai similaritas berbasis frasa, dan Sim_t merupakan nilai similaritas berbasis kata tunggal.

2.9 *Group-Average Hierarchical Clustering*

Pada metode *group-average* dari *HAC* ini, jarak 2 buah *cluster* didefinisikan sebagai rata-rata jarak / similaritas dari semua

pasangan obyek pada 2 buah *cluster* yang dibandingkan tersebut. Gambar 2.8 menunjukkan ilustrasi dari metode *group-average HAC*.



Gambar 2.2 *Group-Average* HAC
Sumber: [YEJ-11]

Gambar 2.2 menunjukkan bahwa tiap anggota *cluster* A dibandingkan dengan semua anggota pada *cluster* B. Setelah dibandingkan dan didapatkan nilai similaritas atau jaraknya, selanjutnya akan dipilih nilai similaritas terbesar (atau nilai jarak yang terkecil) untuk digabungkan dan menghasilkan nilai similaritas atau jarak baru.

Jika menggunakan nilai similaritas, maka cara penghitungannya [MAN-07]:

$$\boldsymbol{Sim(c_i, c_j)} = \frac{1}{|c_i \cup c_j|\,(|c_i \cup c_j|-1)} * \sum_{\vec{x}} \sum_{\vec{y}:\, \vec{y} \neq \vec{x}} \boldsymbol{Sim}(\vec{x}, \vec{y}) \quad (2.7)$$

dimana:
$\boldsymbol{c_i}$ dan $\boldsymbol{c_j}$: dua *cluster* yang dibandingkan
$|\boldsymbol{c_i} \cup \boldsymbol{c_j}|$: jumlah anggota *cluster* setelah digabung
$\boldsymbol{Sim}(\vec{x}, \vec{y})$: nilai similaritas antar dua obyek

2.10 Evaluasi

Pada penelitian ini digunakan pengukuran validitas eksternal dengan menerapkan *F-Measure*, karena sudah ada data pembanding eksternal yang digunakan dalam evaluasi.

Umumnya parameter evaluasi yang digunakan ada 2, yaitu *Recall* dan *Precission*. *Recall* adalah tingkat keberhasilan mengenali suatu *event* dari seluruh *event* yang seharusnya dikenali. *Precission* adalah tingkat ketepatan hasil klasifikasi terhadap suatu *event*.

Cara untuk menghitung nilai *Recall*, *Precission* dan *F-Measure* (Gabungan dari *Recall* dan *Precission*), akan ditunjukkan oleh persamaan 2.8, 2.9 dan 2.10 berikut [MAN-07]:

$$\boldsymbol{Recall} = \frac{\mathbf{a}}{(\mathbf{a}+\mathbf{c})} \quad (2.8)$$

$$\boldsymbol{Precission} = \frac{\boldsymbol{a}}{(\boldsymbol{a}+\boldsymbol{b})} \quad (2.9)$$

$$\boldsymbol{F\text{-}Measure} = \frac{2\,x\,recall\,x\,precission}{recall+precission} \quad (2.10)$$

dimana:
a (*Retrieved - Relevant*) : *Clustering* oleh sistem **Ya**, *clustering* sebenarnya **Ya**.
b (*Retrieved – Not Relevant*) : *Clustering* oleh sistem **Ya**, *clustering* sebenarnya **Tidak**.
c (*Not Retrieved - Relevant*) : *Clustering* oleh sistem **Tidak**, *clustering* sebenarnya **Ya**.
d (*Not Retrieved – Not Relevant*) : *Clustering* oleh sistem **Tidak**, *clustering* sebenarnya **Tidak**.

3. PEMBAHASAN

Dataset yang akan digunakan dalam penelitian ini berupa jurnal ilmiah elektronik berbahasa Inggris dengan format PDF. Jurnal-jurnal ini diperoleh dari beberapa sumber, diantaranya dari internet, referensi / materi perkuliahan, dan arsip-arsip dosen maupun rekan mahasiswa.

Dataset dokumen jurnal yang digunakan berjumlah 166 jurnal, terdiri dari berbagai disiplin ilmu yang terbagi menjadi 5 (lima) *cluster*. Keterangan lebih lengkap mengenai dataset dokumen jurnal dapat dilihat pada tabel 3.1.

Tabel 3.1 Kategori dan Jumlah Dataset

No.	*Cluster*	Jumlah
1	Teknologi Informasi / Informatika (kode: FTI)	46
2	Kedokteran (kode: FK)	33
3	Teknologi Industri (kode: FIn)	39
4	Teknologi Pertanian (kode: FTP)	31
5	Teknologi Perikanan (kode: FPRI)	17
Total Dataset		166

Proses pengujian akan dilakukan terhadap empat macam jumlah *cluster* dan sembilan macam nilai α (alpha). Empat macam jumlah *cluster* ini akan diuji di setiap nilai α (alpha). Keempat macam jumlah *cluster* tersebut terdiri dari 2 (dua), 3 (tiga), 4 (empat), dan 5 (lima) *cluster*. Sedangkan sembilan macam nilai α (alpha) diantaranya 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, dan 0,9. Pengujian dua macam parameter ini (jumlah *cluster* dan nilai α) bertujuan untuk mengetahui pengaruh nilai keduanya terhadap tingkat akurasi hasil *clustering*.

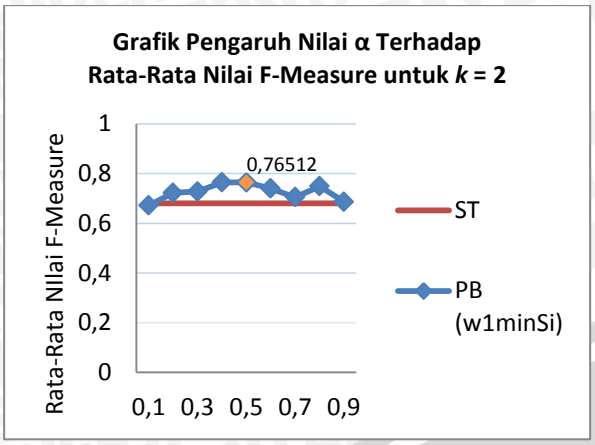
Selain jumlah *cluster* dan nilai α (alpha), pengujian juga dilakukan terhadap rumus/formulasi perhitungan similaritas yang berbasis frasa. Parameter yang diuji adalah nilai level signifikansi (*w*) *matching phrase* dan panjang kalimat asli ($|S_i|$) yang mengandung *matching phrase* tersebut. Untuk *w*, asumsi nilainya adalah $w = 1$ dan $w = f$ (frekuensi *matching phrase*). Sedangkan untuk $|S_i|$, asumsi nilainya adalah $|S_i|$ = minimum (kalimat terpendek) dan $|S_i|$ = maksimum (kalimat terpanjang).

Pengujian pertama dilakukan untuk mengetahui pengaruh jumlah *cluster* dan nilai α (*alpha*) terhadap tingkat akurasi sistem pada pengelompokan dokumen menggunakan algoritma *group-average*, baik untuk yang berbasis kata tunggal (*single term-based*) maupun yang berbasis frasa (*phrase-based*) dengan menerapkan algoritma *index graph* untuk asumsi nilai $w = 1$ dan $min|S_i|$. Kemudian pengujian kedua dengan parameter berbeda, yaitu untuk asumsi nilai $w = f$ dan $min|S_i|$. Pengujian ketiga, yaitu untuk asumsi nilai $w = 1$ dan $max|S_i|$, dan pengujian keempat untuk asumsi nilai $w = f$ dan $max|S_i|$. Keempat jenis pengujian ini diterapkan pada 4 nilai *k* dan 9 nilai *alpha* (α) yang berbeda. Dari keempat pengujian tersebut, untuk asumsi nilai $w = 1$ dan $min|S_i|$ memberikan peningkatan nilai akurasi yang paling baik. Peningkatan ini dihitung dengan membandingkan hasil *clustering* berbasis kata tunggal dengan hasil *clustering* berbasis frasa (gabungan). Hasil pengujian ditunjukkan tabel 3.2.

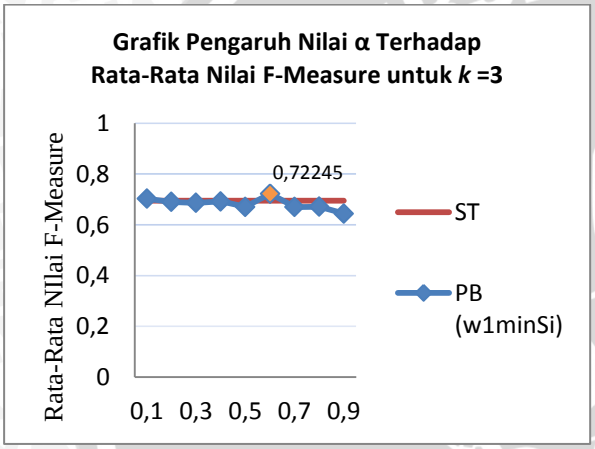
Tabel 3.2 Hasil Pengujian *Group Average Clustering* Berbasis Frasa dengan asumsi $w = 1$ dan $min|S_i|$

k	*Single Term-Based*	*Phrase-Based*	
	F-Measure (avg)	α	*F-Measure* (avg)
2	0,68006	0,1	0,67210
		0,2	0,72303
		0,3	0,72761
		0,4	0,76457
		0,5	0,76512
		0,6	0,74025
		0,7	0,70562
		0,8	0,74899
		0,9	0,68698
3	0,69485	0,1	0,70228
		0,2	0,68989
		0,3	0,68619
		0,4	0,69198
		0,5	0,67026
		0,6	0,72245
		0,7	0,66987
		0,8	0,67104
		0,9	0,64369
4	0,62232	0,1	0,60152
		0,2	0,58966
		0,3	0,59338
		0,4	0,63410
		0,5	0,67272
		0,6	0,65284
		0,7	0,70540
		0,8	0,67470
		0,9	0,65932
5	0,5690	0,1	0,7172
		0,2	0,6172
		0,3	0,6172
		0,4	0,6172
		0,5	0,6172
		0,6	0,6998
		0,7	0,7174
		0,8	0,8587
		0,9	0,5813

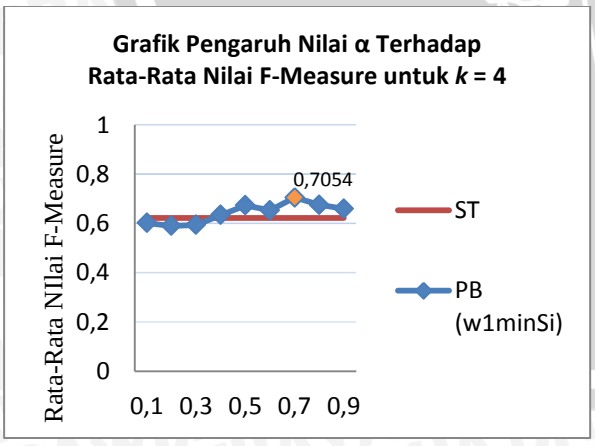
Berdasarkan tabel 4.9 dapat dibentuk grafik yang menyatakan pengaruh besarnya nilai *alpha* (α) terhadap nilai rata-rata *F-Measure* yang dihasilkan pada jumlah *k* tertentu (yang diuji). Gambar 3.1 hingga 3.4 menunjukkan hubungan tersebut.



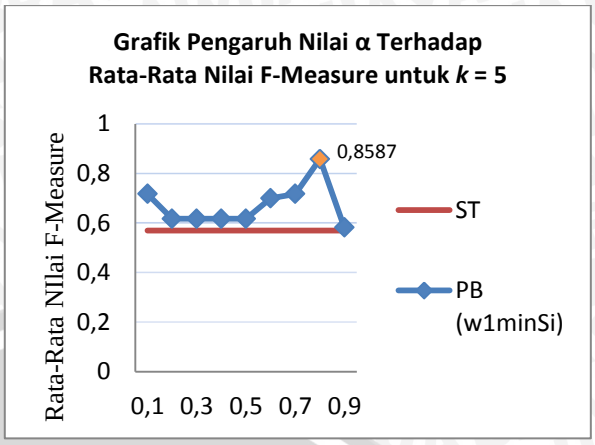
Gambar 3.1 Grafik pengaruh besarnya nilai *alpha* (α) terhadap nilai rata-rata *F-Measure* pada *k* = 2



Gambar 3.2 Grafik pengaruh besarnya nilai *alpha* (α) terhadap nilai rata-rata *F-Measure* pada *k* = 3



Gambar 3.3 Grafik pengaruh besarnya nilai *alpha* (α) terhadap nilai rata-rata *F-Measure* pada *k* = 4



Gambar 3.4 Grafik pengaruh besarnya nilai *alpha* (α) terhadap nilai rata-rata *F-Measure* pada *k* = 5

Berdasarkan keempat grafik diatas, untuk *k* = 2 (Gambar 3.1) peningkatan nilai *f-measure* dari pengolahan berbasis frasa terjadi pada interval nilai *alpha* antara 0,1 sampai 0,5 dimana titik tertinggi terdapat di α = 0,5 sebesar 0,76150 dengan prosentase peningkatan sebesar 12,50% terhadap pengolahan berbasis kata tunggal. Sedangkan untuk *k* = 3 (Gambar 3.2) peningkatan nilai *f-measure* terjadi pada interval nilai *alpha* antara 0,5 sampai 0,6 dimana titik tertinggi terdapat di α = 0,6 sebesar 0,72245 dengan prosentase peningkatan sebesar 3,97%. Kemudian untuk *k* = 4 (Gambar 3.3) peningkatan nilai *f-measure* terjadi pada interval nilai *alpha* antara 0,3 sampai 0,7 dimana titik tertinggi terdapat di α = 0,7 sebesar 0,70540 dengan prosentase peningkatan sebesar 13,35%. Selanjutnya untuk *k* = 5 (Gambar 3.4) keseluruhan nilai *alpha* memberikan nilai *f-measure* yang lebih baik dari pengolahan berbasis kata tunggal, dimana peningkatan nilai *f-measure* yang signifikan terjadi pada interval nilai *alpha* antara 0,5 sampai 0,8 dimana titik tertinggi terdapat di α = 0,8 sebesar 0,8587 dengan prosentase peningkatan sebesar 50,91%. Sehingga jika dirata-rata, maka peningkatan akurasi yang terjadi antara *single term* dan *phrase-based* sebesar 20,18% dengan interval nilai α antara 0,5 hingga 0,8 ($0,5 \leq \alpha \leq 0,8$). Hal ini memberi bukti bahwa dengan menyertakan / memperhitungkan frasa pada proses pengolahan data teks, mampu meningkatkan akurasi / ketepatan pada

sistem. Namun konsekuensinya adalah waktu proses / komputasi yang lebih lama, karena terdapat 2 tahap pra-proses yang harus dilalui, yaitu tahap pra-proses berbasis kata tunggal (*single term based*) dan tahap pra-proses berbasis frasa (*phrase based*).

4. KESIMPULAN DAN SARAN

Sistem pengelompokan jurnal ilmiah dengan menerapkan algoritma *index graph* dan *group average hierarchical clustering* ini diimplementasikan ke 3 tahap utama, yaitu Ekstraksi Jurnal, Proses berbasis Kata Tunggal (*single term based*) dan Proses berbasis Frasa (*phrase based*). Penerapan *graph* sendiri terletak pada sub proses pengolahan berbasis frasa (*phrase based*), sedangkan penerapan *group-average clustering* terletak di sub proses pengolahan keduanya (*single term based* dan *phrase based*).

Proses pengolahan data teks yang melibatkan fitur frasa *2 term* dalam proses analisanya memiliki tingkat akurasi yang lebih baik daripada pengolahan yang hanya berbasis pada kata tunggal. Dibuktikan dengan meningkatnya nilai akurasi / ketepatan pada sistem, rata-rata sebesar 20,18% dari hasil semula yang hanya mengolah kata tunggal dalam proses analisanya. Nilai ini diperoleh saat memasukkan nilai *w* (level signifikansi) sama dengan 1 (satu), dan $|S_i|$ (panjang kalimat asli) adalah minimum pada rumus similaritas berbasis frasa dengan interval nilai *alpha* antara 0,5 hingga 0,8 ($0{,}5 \leq \alpha \leq 0{,}8$).

Keterbatasan *hardware* / perangkat keras membuat jumlah dataset yang diujicobakan dirasa masih kurang, alangkah baiknya jika dilakukan penambahan dataset yang diikuti dengan peningkatan kemampuan *hardware*. Kemudian untuk ukuran frase yang diolah agar tidak terbatas pada 2 kata/*term*, tetapi bisa lebih (*multi-words*). Selain itu juga dilakukan perbandingan dengan metode pengolahan data teks berbasis frasa lainnya, untuk mengetahui metode mana yang menghasilkan tingkat akurasi paling tinggi.

5. DAFTAR PUSTAKA

[ADY-07] Adythia, Resa. 2007. *Studi dan Penerapan Algoritma Pencarian Melebar (Breadth First Search) pada WebSpiders dengan Menggunakan Aplikasi Teleport Pro*. Bandung: Institut Teknologi Bandung.

[ANO-10] Anonymous. 2010. *Konsep Text Mining*. http://perpuspedia.digilib.pnri.go.id/index.php/Text_Mining tanggal akses: 05 November 2011.

[ANO-11] Anonymous. 2011. *Phrases*. http://www.phon.ucl.ac.uk/home/dick/tta/phrases/ phrases tanggal akses: 04 November 2011.

[BJO-09] Björk, Bo-Christer., Roos, Annikki dan Lauri, Mari. 2009. *Scientific journal publishing: yearly volume and open access availability*. Helsinki: Hanken School of Economics.

[ERN-09] Ernawati, S. , Ardiyanti, Arie, dan Setiawan, Erwin B. 2009. *Klusterisasi Dokumen Berita Berbahasa Indonesia Menggunakan Document Index Graph*. Makalah disampaikan pada Seminar Nasional Aplikasi Teknologi Informasi 2009 di Yogyakarta, 20 Juni 2009.

[FAT-09] Fathiya, Shofi N. 2009. *Pengelompokan Dokumen Menggunakan Algoritma DIG (Document Index Graph)*. Bandung: Institut Teknologi Bandung.

[FTO-11] Fatoni, M.A. 2011. *Optimasi Pengelompokan Jurnal Ilmiah Berbahasa Inggris Menggunakan PSO K-Means*. Skripsi Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya Malang.

[HAM-02] Hammouda, K.M. 2002. *Web Document Clustering Using Phrase-Based Document Similarity*. Ontario: University of Waterloo.

[HMO-02] Hammouda, K.M. dan Kamel, Mohamed S. 2002. *Phrase-Based Document Similarity Based on an Index Graph Model*. Ontario: University of Waterloo.

[HAM-04] Hammouda, K.M. dan Kamel, Mohamed S. 2004. *Efficient Phrase-Based Document Indexing for Web Document Clustering*. Ontario: University of Waterloo.

[HAR-06] Harlian, Mikha Ch. 2006. *Text Mining*. Austin: University of Texas.

[HOP-05] Hooper, Rob dan Paice, Chris. 2005. *The Lancaster Stemming Algorithm.*http://www.comp.lancs.ac.uk/computing/research/stemming/index.htm tanggal akses: 19 Desember 2011.

[MAN-07] Manning, C.D., Raghavan, P., dan Schütze, H. 2007. *An Introduction to Information Retrieval*. Cambridge: Cambridge University.

[MAY-78] Mayes, Paul. 1978. *Periodicals Administration in Libraries*. London: Clive Bingley.

[MUS-11] Mustafa, B.N. 2011. *Pengelompokan Jurnal Ilmiah Berbahasa Indonesia dengan Algoritma Agglomerative Complete Linkage Hierarchical Clustering*. Skripsi Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya Malang.

[NUR-10] Nurah, Boy. 2010. *Konversi PDF ke TXT Menggunakan PDFBox di Java*. http://boynurah.wordpress.com/2010/12/12/konversi-pdf-ke-txt-mengguna-kan-pdfbox-di-java/

[PET-00] Petridou, Koutsonikola, Vakali dan Papadimtriou. 2000. *A Divergence-Oriented Approach for Web Users Clustering*. Departement of Informatics Aristotle Univeristy: Greece

[PRA-03] Pramudiono, Iko. 2003. *Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data*. http://www.ilmukomputer.com tanggal akses: 04 November 2011.

[PRA-10] Pratiwi, I.D. 2010. *Klasifikasi Dokumen Berita Berbahasa Indonesia dengan Algoritma Agglomerative Single Linkage Hierarchical Clustering*. Skripsi Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya Malang.

[POR-06] Porter, M. 2006. *The Porter Stemming Algorithm*. http://www.comp.lancs.ac.uk/computing/research/stemming/general/porter.htm tanggal akses 13 November 2011.

[RAH-04] Rahmawati, Tuti. 2004. *Perancangan dan Pembuatan Aplikasi Clustering Dokumen Berbahasa Indonesia dengan Menggunakan Metode Hierarchical Clustering Pendekatan Complete Link*. Surabaya: Institut Teknologi Sepuluh November.

[RIF-95] Rifai, Mien A. 1995. *Pedoman Penerbitan Jurnal Ilmiah Perguruan Tinggi Islam.* Yogyakarta: Univeritas Gajah Mada.

[SES-09] Seshadri, Prasanna. 2009. *PDF Text Parser: Converting PDF to Text in Java Using PDFBox*. http://www.prasannatech.net/2009/01/convert-pdf-text-parser-java-api-pdfbox tanggal akses: 27 Oktober 2011. Source pdfBox: http://sourceforge.net/projects/pdfbox/files/

[SIR-08] Siregar, A.R. 2008. *Desain, Format dan Isi Jurnal Ilmiah*. Medan: Universitas Sumatera Utara.

[STE-00] Steinbach, M., Karypis, G., dan Kumar, V., 2000. *A Comparison of Document Clustering Techniques*. Technical Report. Departement of Computer Science and Engineering:University of Minnesota .

[THE-06] Therling, K. 2006. *An Introduction to DataMining: Discovering hidden value in your data warehouse*. Diambil dari pustaka Kurniawan dan Hidayat (2007).

[TRI-09] Triawati, Candra. 2009. *Text Mining*. http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=590.text-mining&catid=20:informatika&itemid=14 tanggal akses: 05 November 2011.

[WAH-11] Wahyuningsih, N. H. 2011. *Pengelompokan Dokumen Berbahasa Indonesia Menggunakan Algoritma Agglomerative Average Linkage Hierarchical*. Skripsi Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya Malang.

[YEJ-11] Ye, Jieping. 2011. *Cluster Analysis: Basic Concepts and Algorithms*. Department of Computer Science and Engineering: Arizona State University.