# ALGORITMA SINGLE PASS CLUSTERING UNTUK KLASTERING HALAMAN WEB

**Herny Februariyanti [1], Eri Zuliarso[2]**
[1]Sistem Informasi, Unisbank Semarang
[2] Teknik Informatika, Unisbank Semarang
herny@unisbank.ac.id, eri@unisbank.ac.id

***Abstract***

*The process of information retrieval with the internet is often get information very much, but most of information are unneeded. From the viewpoint of information retrieval, a number of information available on the internet more than difficult to get the relevant information, the information that feasible with we needs.*

*This research to cluster documents using Single Pass Clustering Algorithm. This research is emphasized to clustering the Indonesian documents. Links between documents is measured by the simmilarity between the documents. The algorithm was tested with the use of online news archives news documents in HTML format http://www.kompas.com/archive Compass.*

*Test results show that this algorithm can be applied to the grouping of Indonesian documents . The selection of the right keywords will increase the quality of information retrieval in the documents.*

*Keywords—information retrieval, simmilarity, singgle pass clustering.*

## 1. PENDAHULUAN

Usaha untuk memperoleh informasi secara digital telah banyak dilakukan dan perkembangannya sangat pesat seiring dengan perkembangan teknologi komputer. Salah satu cara untuk memperoleh informasi yang seimbang seperti apa yang diinginkan adalah dengan membaca beberapa dokumen yang membahas topik yang sama. Akan tetapi cara ini menyulitkan pembaca untuk menangkap topik bahasan utama dari dokumen - dokumen tersebut karena harus mengingat – ingat isi dokumen yang telah dibaca sebelumnya. Pembaca harus mengintegrasikan dahulu dokumen – dokumen yang tetah dibaca didalam pikirannya sebelum dapat merangkum maksud dan topik utama dokumen – dokumen tersebut secara keseluruhan. Dalam proses penelusuran informasi melalui internet sering diperoleh informasi yang sangat banyak, tetapi sebagian besar diantaranya adalah informasi yang tidak dibutuhkan. Oleh karena itu, dari sudut pandang temu kembali informasi (*information retrieval*), semakin banyaknya informasi yang tersedia di internet justru semakin mempersulit untuk menemukan kembali informasi yang relevan, yaitu informasi yang sesuai dengan kebutuhan.

Dalam suatu sistem temu kembali informasi, kemampuan untuk menemukan informasi yang tersedia diukur dengan *recall* dan kemampuan untuk menemukan informasi yang relevan diukur dengan ketelitian, maka proses penelusuran dalam situasi seperti tersebut di atas akan menghasilkan *recall* yang tinggi tetapi ketelitian rendah.

Sistem yang tepat untuk masalah tersebut adalah Sistem Temu Kembali Informasi yang dapat menghasilkan integrasi dari beberapa dokumen elektronik yang berbeda dengan topik bahasan yang sama secara otomatis. Proses integrasi akan menghasilkan dokumen baru yang mengandung semua bagian dari dokumen – dokumen awal, namun memiliki susunan antar kalimat serta antar paragraf yang berbeda. Perbedaan ini karena saat proses integrasi topik – topik bahasan yang serupa (*similar*) dari semua dokumen dikumpulkan menjadi satu paragraf dan disusun ulang kalimat per kalimat sesuai dengan besarnya kesamaan (*similiarity*) antar kata (*term*). Dengan membaca hasil integrasi diharapkan pembaca dapat terbantu dalam menyerap informasi penting yang ada dalam kumpulan dokumen yang berbeda dan tidak perlu lagi membaca sekumpulan dokumen satu per satu.

## 2. METODE PENELITIAN

Metode yang akan digunakan dalam penelitian ini terdiri dari langkah-langkah sebagai berikut:

1. Obyek Penelitian

Obyek penelitian dari penelitian ini adalah dokumen teks berupa halaman web http://www.kompas.com.

2. Data yang diperlukan
   Merupakan data yang mendukung dalam penelitian ini meliputi data primer dan data sekunder.
   a. Data primer
      Data yang diperoleh dari arsip berita online Kompas http://www.kompas.com/archive dalam format HTML oleh penulis disimpan dalam format teks.
   b. Data Sekunder
      Data yang diperoleh dengan membaca dan mempelajari referensi mengenai *stemming, text mining, clustering, indexing, term weighting, simmilarity, query expand.*
3. Teknik Pengumpulan Data
   Pengumpulan data mempunyai tujuan mendapatkan materi – materi yang mempunyai keterkaitan dengan topik penelitian. Pengumpulan data dimaksudkan agar mendapatkan bahan-bahan yang relevan, akurat dan reliable. Maka teknik pengumpulan data yang dilakukan dalam penelitian ini adalah dengan metode Observasi, Studi Pustaka dan Metode pengembagan dengan menggunakan model prototyping.

## 3. PENGACUAN PUSTAKA

Perkembangan internet yang sangat pesat dan kapasistas dokumen online yang besar menjadikan banyak riset yang membahas tentang sistem temu kembali informasi (information retrieval). [1] Yue W, dkk dalam penelitiannya mengusulkan algoritma sistem temu kembali informasi (*information retrieval*) berbasis *query expansion* dan klasifikasi. Algoritma tersebut diinduksi dari *query* yang pendek dan metode pencarian informasi tradisional (*traditional retrieval information method*) yang menghasilkan presisi yang rendah walaupun tingkat recall cukup tinggi. Penelitian ini berusaha untuk mendapatkan lebih banyak dokumen yang relevan dengan mengekspand *query* (*query expansion*) dan klasifikasi dokumen. Hasil dari penelitian dengan algoritma query expansion dan klasifikasi dokumen mengahasilkan ketepatan dan efisiensi pencarian dokumen dibandingakan dengan metode query expand tradisional.

Fakta penelitian selama 20 tahun menunjukkan bahwa sistem yang berbasis indeks teks dengan menggunakan pembobotan sebuah dokumen yang tepat mengahasilkan pencarian lebih tepat. Menurut Salton[2] hasilnya sangat tergantung dari efektifitas sistem pembobotan dokumen. Dalam penelitian ini merangkum informasi yang berhubungan dengan otomatisasi pembobotan dokumen dan menyediakan dasar pembobotan dokumen secara tunggal (*single-term-weghting*) dibandingan dengan menggunakan prosedur lain yang lebih rumit.

[3]Steinbach M, dkk melakukan penelitian dengan memberikan sebuah pendekatan baru yang lebih mudah dipahami dan dapat digunakan untuk membentuk klaster. Pendekatan tersebut dimotivasi dari sebuah penelitian bahwa jika ada sebuah pola yang besar dalam sebuah data, pola ini biasanya terbagi dalam klaster yang berbeda jika menggunakan pendekatan klaster yang sudah ada. Hal ini dikarenakan alagoritma yang dikembangkan tidak memanfaatkan pengetahuan tentang pola dan sering kali tujuan bertolak belakang dengan pemeliharaan pola (preserving pattern). Misalnya dengan meminimalkan jarak terdekat dari sentral klaster. Pada penelitian ini difokuskan untuk memberikan ciri, yaitu dengan menggunakan pola klaster dan memberikan cara yang paling efektif dalam proses klastering. Penelitian dilakukan dengan mengevaluasi dua algoritma klastering yaitu metode *Hierarchical Clustering* dan metode *Bisecting K-Mean Clustering*. Dari penelitian dengan mengunakan metode *Hierarchical Clustering* dalam melakukan proses klastering memberikan hasil lebih baik dibandingkan dengan menggunakan metode K-Means.

[4]Karypis, dkk melakukan analisis bahwa algoritma klastering yang cepat dan berkualitas tinggi sangat berperan dalam menyediakan pedoman dan mekanisme pencarian dalam mengelola informasi yang luas untuk ukuran klaster yang kecil yang sangat berarti. Secara khusus algoritma klastering membangun hierarki koleksi dokumen yang sangat besar dengan alat yang ideal untuk interaktif visualisasi dan pengembangan data-view yang konsisten. Secara khusus, algoritma klastering sangat berarti dalam membangun hasil hierarki dalam koleksi dokumen yang sangat besar. Algoritma tersebut adalah alat yang ideal untuk penyajian visualisasi dan eksplorasi ketika kita memberikan data-view yang sesuai dengan prediksi, dan pada tingkat ketelian yang berbeda. Penelitian ini fokus pada algoritma klastering dokumen yang benar-benar membangun solusi dengan hierarchical dan memberikan kajian yang lengkap tentang algoritma

*Agglomerative* dan algoritma *Partitional* dengan menggunakan kriteria yang berbeda dan menggabungkan skema dan menyajikan kelas (class) baru dari algoritma klastering. Yaitu dengan cara menggabungkan fitur dari kedua algoritma yaitu partitional dan agglomeratif. Dengan pendekatan tersebut akan mengurangi kesalahan pada tahap awal dengan menggunakan metode agglomerative meningkatkan hasil klastering. Hasil penelitian menunjukkan bahwa algoritma *agglomerative* lebih baik dibandingkan dengan metode algoritma *partitional*. Dengan menggunakan *Clustering Single Pass* membuat ideal dalam mengklaster koleksi dokumen yang sangat besar dengan komputasi yang relatif rendah, tetapi menghasilkan klaster dengan hasil kualitas tinggi. Selanjutnya metode *Agglomerative* secara konsisten dibatasi lebih baik dibandingkan dengan metode *Agglomerative* tunggal dan beberapa kasus metode tersebut lebih unggul dari metode partitional.

*3.1. Klastering Dokumen*

Klastering biasa digunakan pada banyak bidang, seperti : data mining, pattern recognition (pengenalan pola), image classification (pengklasifikasian gambar), ilmu biologi, pemasaran, perencanaan kota, pencarian dokumen, dan lain sebagainya.

Tujuan dari klastering adalah untuk menentukan pengelompokan dari suatu set data. Akan tetapi tidak ada ”ukuran terbaik” untuk pengelompokan data. Untuk pengelompokkan data tergantung tujuan akhir dari klastering, maka diperlukan suatu kriteria sehingga hasil klastering seperti yang diinginkan.

Penelitian tentang clustering document (klastering dokumen) telah banyak dilakukan. Secara umum klastering dokumen adalah proses mengelompokkan dokumen berdasarkan kemiripan antara satu dengan yang lain dalam satu klaster [5,6] Gordon, Ellis.

Tujuan klastering dokumen adalah untuk memisahkan dokumen yang relevan dari dokumen yang tidak relevan [7] Zhang J, dkk. Atau dengan kata lain, dokumen-dokumen yang relevan dengan suatu query cenderung memiliki kemiripan satu sama lain dari pada dokumen yang tidak relevan, sehingga dapat dikelompokkan ke dalam suatu klaster.

Klastering dokumen dapat dilakukan sebelum atau sesudah proses temu kembali [7] Zhang J, dkk. Pada klastering dokumen yang dilakukan sebelum proses temu kembali informasi, koleksi dokumen dikelompokkan ke dalam klaster berdasarkan kemiripan (similarity) antar dokumen. Selanjutnya dalam proses temu kembali informasi, apabila suatu dokumen ditemukan maka seluruh dokumen yang berada dalam klaster yang sama dengan dokumen tersebut juga dapat ditemukan.

Pada algoritma klastering, dokumen akan dikelompokkan menjadi *klaster-klaster* berdasarkan kemiripan satu data dengan yang lain. Prinsip dari *klastering* adalah memaksimalkan kesamaan antar anggota satu klaster dan meminimumkan kesamaan antar anggota *klaster* yang berbeda.

*3.2. Single Pass Clustering*

Single Pass Clustering merupakan suatu tipe clustering yang berusaha melakukan pengelompokan data satu demi satu dan pembentukan kelompok dilakukan seiring dengan pengevaluasian setiap data yang dimasukkan ke dalam proses cluster. Pengevaluasian tingkat kesamaan antar data dan cluster dilakukan dengan berbagai macam cara termasuk menggunakan fungsi jarak, vectors similarity, dan lain-lain.

Algoritma yang sering digunakan dalam Single Pass Clustering adalah sebagai berikut: [8] Salton G.,: untuk setiap perulangan data d

a. Mencari a klaster c dengan memaksimalkan fungsi tiap objek
b. jika nilai fungsi tiap objek > nilai threshold maka data d termasuk dalam klaster c
c. Jika tidak maka menciptakan klaster baru dalam data d

2. Perulangan berhenti

Dalam menggunakan algoritma single pass, dua hal yang perlu menjadi perhatian adalah penentuan objective function dan penentuan threshold value. Objective function yang ditentukan haruslah sebisa mungkin mencerminkan keadaan data yang dimodel dan dapat memberikan nilai tingkat kesamaan atau perbedaan yang terkandung di dalam data tersebut. Penentuan threshold value juga merupakan hal yang subjektif, makin besar nilai threshold, makin mudah suatu data untuk bergabung ke dalam suatu cluster, dan demikian juga sebaliknya.

*3.3. Stammer Tala*

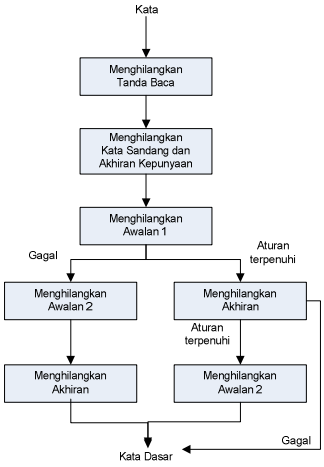
Stemmer Tala merupakan adopsi dari algoritma stemmer bahasa Inggris terkenal porter stemmer. Stemmer ini menggunakan rule base analisis untuk mencari root sebuah kata. Stemmer ini sama sekali tidak menggunakan kamus sebagai acuan, seperti halnya stemmer vega dan jelita.

Morphologi bahasa Indonesia dapat terdiri dari turunan dan imbuhan kata. Imbuhan yang sederhana

digunakan akhiran dimana tidak akan merubah makna dari kata dasar.[9] Tala F., Z.

Dalam proses *stemming* bahasa Indonesia ini terdapat beberapa tahap. Sebuah kata akan dites dengan menggunakan *rule* yang dibuat pada setiap tahap. Pada setiap tahap, sebuah kata yang memenuhi kondisi untuk *rule* pada tahap itu maka kata tersebut akan diganti dengan kata baru yang dibentuk dengan *substitution rule* (aturan pengganti).

Arsitektur proses stemming untuk bahasa Indonesia dapat dilihat pada Gambar 1. Tahap pertama yang dilakukan adalah menghilangkan partikel kata (tanda baca) kemudian menghilangkan *possesive pronouns* (kata sandang dan akhiran kepunyaan). Baru setelah itu dilakukan proses untuk menghilangkan *prefiks* (awalan) pertama dari kata tersebut. Jika kata tersebut memiliki *prefiks* pertama maka proses selanjutnya yang dipilih adalah proses menghilangkan *sufiks* dan kemudian menghilangkan *prefiks* kedua. Namun jika kata tersebut tidak memiliki *prefiks* pertama maka proses selanjutnya adalah menghilangkan *prefiks* kedua berubah kemudian menghilangkan *sufiks*.




Gambar 1. Proses Stemming Algoritma Tala

### *3.4. Cosin Simmilarity*

Metode cosine distance merupakan metode yang digunakan untuk menghitung similarity (tingkat kesamaan) antar dua buah obyek. Untuk tujuan klastering dokumen fungsi yang baik adalah fungsi Cosine Similaritas. Berikut adalah persamaan dari metode Cosine Distance : [10] Salton, dkk.

Untuk notasi himpunan digunakan rumus :

$$Similarity(x,y) = \frac{\sum_{i=1}^{t} x_i \, y_i}{\sqrt{\sum_{i=1}^{t} x_i^2 \cdot \sum_{i=1}^{t} y_i^2}} \qquad (1)$$

dimana :
x dan y adalah dukumen yang berbeda.
$x_i$ = term i yang ada pada dokumen x
$y_i$ = term i yang ada pada dokumen y

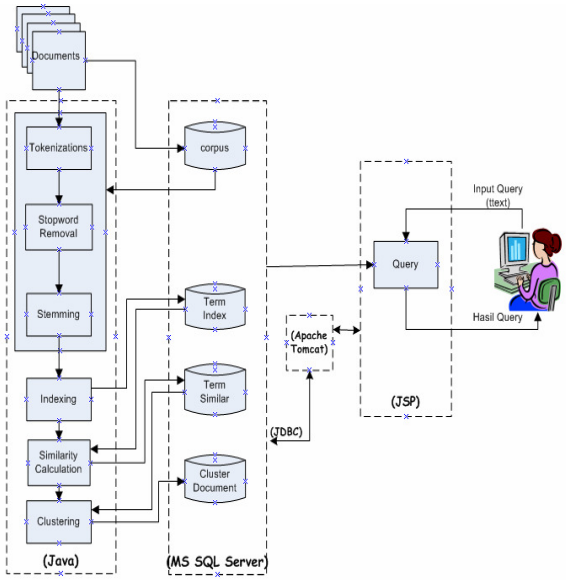
## 4. HASIL DAN PEMBAHASAN

Program aplikasi adalah membangun sistem klastering dokumen dengan menggunakan algoritma *Clustering Single Pass*. Program aplikasi digunakan user untuk membantu mendapatkan dokumen yang berada dalam klaster yang sama.

Dengan menggunakan program aplikasi ini user akan dengan mudah mendapatkan informasi dokumen yang sejenis tanpa harus membaca beberapa dokumen. User tidak perlu menyimpulkan sendiri dari dokumen yang dibacanya untuk mendapatkan informasi yang diinginkan.

### *4.1 Arsitektur sistem temu kembali informasi*

Sistem Temu Kembali Informasi dengan algoritma *Clustering Single Pass* sebagai suatu sistem memiliki beberapa proses (modul) yang membangun sistem secara keseluruhan. Modul Sistem Temu Kembali Informasi terdiri dari : modul tokenizations (tokenisasi), modul stop word removal (pembuangan stop word), modul stemming (pengubahan kata dasar), modul term indexing (pengindeksan kata), term similarity (kesamaan kata) dan modul clustering (pengelompokan). Secara lengkap arsitektur dari modul Sistem Temu Kembali Informasi dapat dilihat pada Gambar 2.

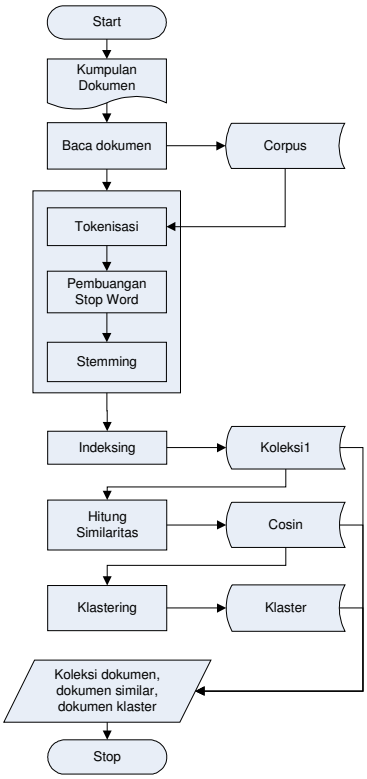



Gambar 2. Arsitektur Sistem Temu Kembali Informasi

*4.2. Flowchart Sistem Temu Kembali Informasi*

Flowchart sistem untuk Sistem Temu Kembali Informasi seperti terlihat pada Gambar 3, dimulai dari proses untuk pembentukan database klaster dokumen. Dilanjutkan dengan proses untuk Sistem Temu Kembali Iinformasi.

Proses untuk pembentukan database klaster dokumen terdiri dari: 1) proses preprosesing terdiri dari 3 proses : proses *tokenization* (tokenisasi), proses *stop word removal* (pembuangan stopword) dan proses *stemming*, 2) proses indexing, 3) proses *similarity calculations* (hitung similaritas), 4) proses *clustering* (klastering).




Gambar 3. Flowchart Pembentukkan Database Klaster Sistem Temu Kembali Informasi

Masing-masing proses dapat dijelaskan sebagai berikut :

*4.2.1. Tokenisasi*

Sebelum kata dipisahkan dari kalimatnya, terlebih dahulu dibersihkan dari tanda baca, tag html dan angka. Pada penelitian ini untuk membersihkan tanda baca dapat digunakan perintah yang disediakan oleh Java. Pembersihan dilakukan sebelum proses tokenisasi (*tokenizations)* dimaksudkan untuk memperkecil hasil dari tokenisasi. Pada proses tokenisasi akan dibaca dokumen abstrak dalam format teks akan dilakukan proses pemotongan string input berdasarkan tiap kata yang menyusunnya. Pada umumnya setiap kata terindentifikasi atau terpisahkan dengan kata yang lain oleh karakter spasi, sehingga proses tokenisasi mengandalkan karakter spasi pada dokumen untuk melakukan pemisahan kata.

*4.2.2. Pembuangan stopword*

Proses pembuangan *stopword* dimaksudkan untuk mengetahui suatu kata masuk ke dalam *stopword* atau tidak. Pembuangan *stopword* adalah proses pembuangan *term* yang tidak memiliki arti atau tidak relevan. *Term* yang diperoleh dari tahap tokenisasi dicek dalam suatu daftar *stopword*, apabila sebuah kata masuk di dalam daftar *stopword* maka kata tersebut tidak akan diproses lebih lanjut. Sebaliknya apabila sebuah kata tidak termasuk di dalam daftar stopword maka kata tersebut akan masuk keproses berikutnya. Daftar *stopword* tersimpan dalam suatu tabel, dalam penelitian ini menggunakan daftar stop word yang digunakan oleh .[9] Tala F., Z. yang merupakan *stopword* Bahasa Indonesia yang berisi kata-kata seperti ; ini, itu, yang, ke, di, dalam, kepada, dan seterusnya sebanyak 780 kata.

*4.2.3. Stemming*

Proses *stemming* adalah proses pembentukan kata dasar. *Term* yang diperoleh dari tahap pembuangan *stop word* akan dilakukan proses *stemming*. Algoritma stemming yang digunakan adalah modifikasi Porter stemmer .[9] Tala F., Z. *Stemming* digunakan untuk mereduksi bentuk *term* untuk menghindari ketidak cocokan yang dapat mengurangi *recall*, di mana *term-term* yang berbeda namun memiliki makna dasar yang sama direduksi menjadi satu bentuk.

Proses stemming adalah bagian dari proses filtering, tujuan utama dari proses *stemming* adalah mengembalikan kata dalam bentuk dasarnya. Dengan kata dasar dapat mereduksi bentuk term untuk menghindari ketidak cocokan yang dapat mengurangi *recall*, di mana *term-term* yang berbeda namun memiliki makna dasar yang sama direduksi menjadi satu bentuk.

*4.2.4. Indexing*

Proses *indexing* merupakan tahapan preprocessing yang sangat penting dalam sistem temu kembali informasi sebelum pemrosesan query. Pada proses ini seluruh dokumen dalam koleksi disimpan dalam suatu file dengan format sedemikian sehingga dokumen satu dengan dokumen yang lain dapat dibedakan. Setelah kata telah dikembalikan dalam bentuk asal (kata dasar), kata-kata tersebut disimpan kedalam tabel basis data. Penelitian ini menggunakan metode

*Inverted Index*, dengan struktur terdiri dari: kata (*term*) dan kemunculan. Kata-kata tersebut adalah himpunan dari kata-kata yang ada pada dokumen, merupakan ekstraksi dari kumpulan dokumen yang ada. Setiap term akan ditunjukkan informasi mengenai semua posisi kemunculannya secara rinci.

*4.2.5. Hitung similaritas*

Relevansi sebuah dokumen ke sebuah *query* didasarkan pada *similarity* (similaritas) diantara vektor dokumen dan vektor *query*. Koordinat dari bobot istilah secara dasarnya diturunkan dari frekuensi kemunculan dari istilah. Pada modul ini akan dihitung presentase kemunculan tiap kata (term) dan presentase kesamaan antar dua *term.* Metode yang digunakan untuk menghitung adalah metode *cosine simmilarity* dengan menggunakan rumus seperti diuraikan pada persamaan (1).

Masing-masing dokumen akan dihitung cacah term yang sama antara dokumen yang satu dengan dokumen yang lain. Hasil dari hitung cacah akan dihasilkan dokumen dengan nilai similaritas dokumen. Nilai similaritas dokumen yang tertinggi dapat dianggap bahwa dokumen tersebut paling simmilar, yaitu memiliki banyak kesamaan.

*4.2.6. Klastering*

Pada penelitian ini dokumen akan dibuat klaster dengan menggunakan metode *Clustering Single Pass*. Metode ini berawal dari objek-objek individual. Jadi pada awalnya banyaknya klaster sama dengan banyaknya objek. Pertama-tama objek-objek yang paling mirip dikelompokkan, dan kelompok-kelompok awal ini digabungkan sesuai dengan kemiripannya (similaritas). Akhirnya, sewaktu kemiripan berkurang, semua subkelompok digabungkan menjadi satu klaster tunggal. Begitu seterusnya dari hasil similaritas yang tertinggi akan dibandingkan dengan dokumen yang satu dengan dokumen yang lain, sehingga didapat similaritas terendah. Hasil similaritas terendah menyatakan bahwa dokumen tersebut merupakan klaster yang berbeda.

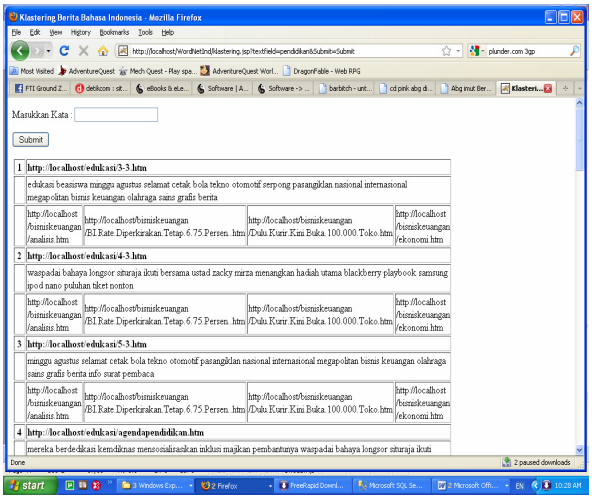
Setelah Sistem Informasi Temu Kembali Informasi Klastering Berita on Line dapat diimplementasikan sesuai dengan desain yang telah dibuat. Tahap selanjutnya adalah tahap melakukan percobaan atau testing dan evaluasi terhadap sistem yang dibuat. Pada tahap pengetesan ini penulis tidak menemukan kesalahan pada program baik secara logika maupun sintaks pada kode program.

Pengujian yang penulis lakukan dalam Sistem Informasi Temu Kembali Informasi Klastering Berita on Line yang berjumlah 60 file dalam format html, telah mampu untuk tidak melakukan indeks-indeks kata umum (stop word) dan telah membentuk kata dasar dari tiap *term* yang ada dalam dokumen abstrak tersebut. Selanjutnya setiap term telah dihitung frekuensinya dan diberikan pembobotan menggunakan cosine similaritas dan selanjutnya term tersebut disimpan pada database korpus.

Selanjutnya dalam pengujian terhadap sistem penulis melakukan pengujian input string *query* dan kemudian hasil pengujian input *query* dilakukan pengukuran hasil retrieval (temu kembali informasi) hasil dengan pengujian *recall precision*.

*4.3. Pengujian Input Query*

Pada tahap pengujian input *query* dilakukan dengan cara memasukkan *query* "pendidikan", "jaringan", "bisnis", dan "manajemen bisnis". Terlihat pada Gambar 4 adalah salah satu contoh hasil tampilan dari input *query* "pendidikan". Hasil proses dari *query* akan ditampilkan dokumen-dokumen yang berada dalam klaster yang sama.




Gambar 4. Hasil Dokumen Input Query "pendidikan"

*4.4. Pengujian Recall Presisi*

Untuk mengevaluasi secara manual kesamaan diantara dokumen dalam cluster-cluster yang telah dikelompokkan digunakan standar sebagaimana Tabel 1. Tabel tersebut berisi berbagai kemungkinan hasil klasifikasi pada tiap *event* (*Per Event contengency table*).

Tabel 1. kategori hasil klasifikasi

| | In Event | Not In event |
|---|---|---|
| *In cluster* | a | B |
| *Not In Cluster* | c | D |

Tabel 1 menunjukkan bahwa hasil klasifikasi adakalanya memang termasuk event (a) yang dimaksud dan adakalanya tidak (b). Sedangkan dokumen yang tidak termasuk dalam hasil klasifikasi suatu *event*, adakalanya memang bukan anggota *event* itu (d) dan adakalanya ternyata seharusnya menjadi anggota event tersebut (c). Dalam hal ini, kempat parameter di atas digunakan untuk menghitung 2 parameter evaluasi, yakni :

1. *Recall*, yakni tingkat keberhasilan mengenali suatu event dari seluruh event yang seharusnya dikenali. Rumusnya adalah r = a/(a+c) untuk a+c > 0. Selain itu tidak didefinisikan
2. *Precision*, yakni tingkat ketepatan hasil klasifikasi terhadap suatu event.Artinya,dari seluruh dokumen hasil klasifikasi, berapa persenkah yang dinyatakan benar.
   Rumusnya adalah p = a/(a+b) jika a+b > 0.

Selain itu tidak didefinisikan Dari hasil evaluasi yang dilakukan terhadap data training yang diambil dari suara pembaruan *online m*ulai tanggal 1 Agustus 2001 sampai dengan 31 Agustus 2001 dengan perincian sebagai berikut :

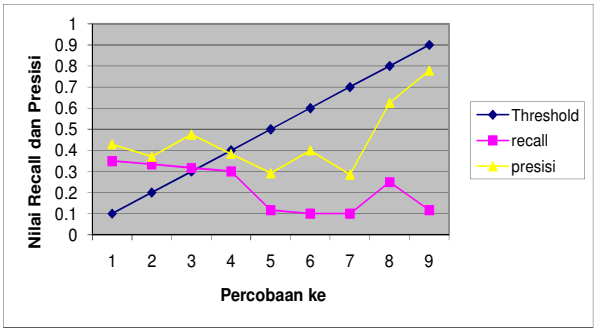
- Diambil 60 Dokumen sebagai Restrospective/training set yang dikalsifikasikan kedalam 23 event
- Setelah melalui proses *stemming* maka dapat dilakukan penghitungan frekuensi kata dalam dokumen dengan menggunakan kamus sejumlah 29.349 kata dasar bahasa Indonesia
- Dari matrik yang dibentuk dihasilkan sebanyak 13.000 *record* untuk kaitan dokumen dengan kata tf(t,d) dengan frekuensi di atas 0

Didapatkan hasil nilai Recall dan Precision sebagaimana Tabel 2.

Tabel 2 Tabel Hasil Uji Coba

| **Threshold** | **recall** | **presisi** |
|---|---|---|
| 0.1 | 0.35 | 0.428571429 |
| 0.2 | 0.333333333 | 0.37037037 |
| 0.3 | 0.316666667 | 0.475 |
| 0.4 | 0.3 | 0.382978723 |
| 0.5 | 0.116666667 | 0.291666667 |
| 0.6 | 0.1 | 0.4 |
| 0.7 | 0.1 | 0.285714286 |
| 0.8 | 0.25 | 0.625 |
| 0.9 | 0.116666667 | 0.777777778 |

Distribusi nilai kedua parameter dapat digambarkan dengan grafik pada Gambar 5




Gambar 5. Grafik Recall dan Precision untuk *Algoritma Single Pass*

Dari Gambar 5 terlihat bahwa nilai terbaik untuk recall didapat pada treshold pada angka 0.1. Sedanhkan nilai terbaik untuk presisi dihasilkan dari treshold pada angka 0.9. Treshold tersebut didapatkan dari percobaan secara linear terhadap 9 treshold yang berbeda. Percobaan dilakukan dengan menggunakan treshol mulai dari nilai treshold 0.1 sampai dengan mendapatkan jumlah kluster = jumlah dokumen atau nilai treshold di atas keseluruhan similarity maksimal. Untuk jumlah kluster yang dihasilkan dari 9 percobaan yang dilakukan dapat dilihat pada tabel 3

Tabel 3 Tabel Hasil Kluster

| No | Threshold | Banyak Klaster |
|---|---|---|
| 1 | 0.1 | 8 |
| 2 | 0.2 | 8 |
| 3 | 0.3 | 10 |
| 4 | 0.4 | 12 |
| 5 | 0.5 | 19 |
| 6 | 0.6 | 30 |
| 7 | 0.7 | 30 |
| 8 | 0.8 | 31 |
| 9 | 0.9 | 37 |

Sedangkan distribusi jumlah kluster yang dihasilkan dari percobaan yang dilakukan dapat dilihat pada Gambar 6

Gambar 6. Grafik Hasil Kluster

## 5. KESIMPULAN

Dari hasil penelitian yang telah dilakukan dapat disimpulkan hal-hal sebagai berikut:

1. Pembobotan term frekuensi dan cosine similaritas digunakan untuk menunjukkan kemiripan antar dokumen.
2. Sistem dapat menampilkan dokumen yang mempunyai kedekatan similaritas dari query yang diinputkan user.
3. Dokumen yang membahas topik yang sama cenderung untuk mengelompok menjadi satu klaster.
4. Klaster dapat membantu menemukan dokumen yang ada dalam satu klaster dengan *query* yang diinputkan user.
5. Klaster dapat membantu mendapatkan dokumen yang relevan.
6. Nilai *treshold* (nilai batas) yang paling bagus digunakan adalah 0.2 dengan nilai *recall* sebesar 0.33 dan precision 0.37

## 6. SARAN

Dengan keterbatasan kemampuan dan waktu yang tersedia penulis menyadari bahwa masih banyak terdapat kekurangan dalam sistem ini terutama metode klastering yang digunakan. Kedepan nantinya diharapkan dalam pengembangan Sistem Informasi berbasis web, penulis menyarankan beberapa hal:

1. Sistem yang dibuat dapat dikembangkan lebih lanjut dengan menerapkan pada file teks Bahasa Indonesia dengan melakukan modifikasi stopword dan algoritma stemming agar hasil stemming lebih optimal.
2. Bagi peneliti lain yang berniat mengembangkan sistem Informasi Temu Kembali Bahasa Indonesia ini disarankan untuk menggunakan metode klastering yang diperluas sehingga hasil klaster dokumen akan lebih baik.
3. Untuk term yang bernilai 0 (nol) dalam setiao dokumen tidak perlu dilibatkan dalam perhitungan, karena hanya akan menambah waktu perhitungan.

## UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Universitas Stikubank (Unisbank) Semarang. yang telah memberi dukungan financial terhadap penelitian ini.

## DAFTAR PUSTAKA

[1] Yue, W., 2005, *Using Query Expansion and Classification for Information Retrieval, College of Computer and Communication*, Hunan University ChangSha, Hunan Province, 410082,China.

[2] Salton, G., 1971, *Cluster Search Strategies and the Optimization of Retrieval Efectiveness*, dalam G. Salton, ed. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Englewood Cliffs: Prentice-Hall, 223-242

[3] Steinbach, M., Xiong H., Ruslim A., Kumar V., 2007, *Characterizing Pattern Preserving Clustering, Department of Management Science and Information Systems Rutgers*, the State University of New Jersey, USA.

[4] Karypis G., Zhao Y., 2004, *Hierarchical Clustering Algorithms for Document Datasets*, *University of Minnesota, Department of Computer Science and Engineering and Digital Technology Center and Army HPC Research Center, Minneapolis, MN 55455.*

[5] Gordon, Michael D., 1991, *User-Based Document Clustering by Redescribing Subject Descriptions with a Genetic Algorithm,*. Journal of American Society for Information Science, 311-322.

[6] Ellis, David, 1996, *Progress and Problems in Information Retrieval*, 2nd ed. London: Library Association.

[7] Zhang J., Jianfeng G., Ming Z., Jiaxing W., 2001, *Improving the Effectiveness of Information Retrieval with Clustering and Fusion*, Computational Linguistics and Chinese Language Processing, Vol. 6, No. 1, February 2001, pp. 109-125.

[8] Salton, G., 1989, *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*, Addison – Wesly Publishing Company, Inc. All rights reserved.

[9] Tala F.,Z., 2003, *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Institute for Logic, Language and Computation Universiteit van Amsterdam The Netherlands.

[10] Salton, G. and Buckley, 1988, *Term Weigting Approaches in Automatic Text Retrieval, Department of Computer Science*, Cornell University, Ithaca, NY 14853, USA.