PENGARUH ALGORITMA *STEMMING* DALAM PEMILAHAN ARTIKEL BERITA MENGGUNAKAN METODE *SINGLE PASS CLUSTERING*

SKRIPSI

Oleh:
VERLY RAHMADHANI
0510960062-96

Sebagai salah satu syarat untuk memperoleh gelar Sarjana dalam bidang Ilmu Komputer

PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011


UNIVERSITAS BRAWIJAYA

LEMBAR PENGESAHAN SKRIPSI

PENGARUH ALGORITMA *STEMMING* DALAM PEMILAHAN ARTIKEL BERITA MENGGUNAKAN METODE *SINGLE PASS CLUSTERING*

Oleh**:**
VERLY RAHMADHANI
0510960062-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 21 Februari 2011
dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana
dalam bidang Ilmu Komputer

Pembimbing I	Pembimbing II
Drs. Achmad Ridok, M.Kom	**Muhammad Tanzil Furqon, S.Kom**
NIP. 19680825 199403 1 002	**NIP. 19820930 200801 1 004**

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya
Ketua,

Dr. Abdul Rouf Alghofari, MSc
NIP. 19670907 199203 1 001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama **:** Verly Rahmadhani
NIM **:** 0510960062
Jurusan **:** Matematika
Penulis skripsi berjudul **:** Pengaruh Algoritma *Stemming* dalam Pemilahan Artikel Berita Menggunakan Metode *Single Pass Clustering.*

Dengan ini menyatakan bahwa :

1. **Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.**
2. **Apabila dikemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.**

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 21 Februari 2011
Yang menyatakan,

(Verly Rahmadhani)
NIM. 0510960062

ABSTRAK

Clustering dokumen merupakan proses klasifikasi dokumen ke dalam kelompok atau grup. Ada banyak metode klasifikasi, seperti *k-means, hierarchical clustering, KNN clustering, single pass clustering* dan masih banyak metode yang lain. *Single pass* efisien dan biasa digunakan dalam penanganan data dalam jumlah yang banyak.

Dalam skripsi ini, pada proses *preprocessing* pada tahap *stemming* digunakan dua algoritma *stemming* dan penggunaan tanpa *stemming* guna mengetahui pengaruh algoritma *stemming* terhadap *clustering* dokumen. Algoritma *stemming* yang digunakan yaitu *stemming* Arifin Setiono dan *stemming* Nazief Adriani.

Setelah dilakukan serangkaian uji coba, algoritma *stemming* dapat meningkatkan nilai *precision* dan *recall* dalam *clustering* dokumen menggunakan *single pass clustering*.

Kata Kunci : *text mining, stemming* Arifin, *stemming* Nazief, TF-IDF, *Vector Space Model*, *Single Pass Clustering*


UNIVERSITAS BRAWIJAYA

ABSTRACT

Document clustering is a process of classification of documents into groups or group. There are many classification methods, like k-means, hierarchical clustering, kNN clustering, single pass clustering and many other methods. Single pass efficient and commonly used in the handling of data in large numbers.

In this thesis, the preprocessing stage of the process used two stemming algorithm and use without stemming also, to determine the effect of stemming algorithm in clustering documents. The stemming algorithm used is stemming Arifin and Setiono Nazief Adriani.

After a series of tests, stemming algorithms can increase the value of precision and recall in document clustering using single pass clustering.

Key Word : text mining, stemming Arifin, stemming Nazief, TF-IDF, Vector Space Model, Single Pass Clustering

KATA PENGANTAR

Alhamdulillahi rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, atas segala rahmat dan karuniaNya, penulis dapat menyelesaikan skripsi yang berjudul "Perancangan Sistem Deteksi Plagiarisme Dokumen Teks dengan Menggunakan Algoritma Rabin-Karp".

Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Dalam penyelesaian tugas akhir ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari berbagai pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih kepada:

1. Drs. Achmad Ridok, M.Kom selaku pembimbing I dan Muhammad Tanzil Furqon, S.Kom sebagai pembimbing II. Terima kasih atas semua waktu dan bimbingan yang telah diberikan.
2. Segenap bapak dan ibu dosen yang telah mendidik dan mengamalkan ilmunya kepada penulis.
3. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya yang telah membantu kelancaran pengerjaan skripsi.
4. Papa, Mama, dan Adik. Terima kasih atas cinta, kasih sayang, doa, dukungan dan semangat yang tiada henti.
5. Eko Nugroho, Tresnaningtiyas S. P., Jayanti Utari, Mega Satya C., Widhy Hayuhardhika N. P., Dharma Surya P., Indah, Killa, Ucup, Dee, Cece, Dhie, Ninot, Ferni atas bantuan, dukungan, semangat dan doanya.
6. Sahabat-sahabat ilkomers angkatan 2005 dan seluruh warga Program Studi Ilmu Komputer Universitas Brawijaya.
7. Pihak lain yang telah membantu terselesaikannnya skripsi ini yang tidak bisa penulis sebutkan satu-persatu.

Penulis sadari bahwa masih banyak kekurangan dalam laporan ini disebabkan oleh keterbatasan kemampuan dan pengalaman. Oleh karena itu Penulis sangat menghargai saran dan kritik yang sifatnya

membangun demi perbaikan penulisan dan mutu isi skripsi ini untuk kelanjutan penelitian serupa di masa mendatang.

Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya, baik oleh Penulis selaku mahasiswa maupun pihak-pihak lain yang tertarik untuk menekuni pengembangan *Text Mining*.

Malang, 21 Februari 2011

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE PROGRAM

DAFTAR LAMPIRAN

UNIVERSITAS BRAWIJAYA

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pengertian berita dalam Kamus Besar Bahasa Indonesia adalah cerita atau keterangan mengenai kejadian atau peristiwa yang hangat, berita juga berarti laporan pemberitahuan atau pengumuman. Dalam penyampaian atau publikasi berita memerlukan media komunikasi sebagai sarana perantara. Bentuk media tersebut ada dua yaitu media cetak dan media elektronik. Berita media cetak merupakan berita yang disampaikan dalam bentuk tertulis yang melalui proses pencetakan seperti koran, majalah, tabloid dan lain sebagainya. Berita media elektronik merupakan berita dengan menggunakan sarana elektronik yang disampaikan dengan cara audio, visual seperti televisi dan radio, selain itu juga dapat melalui jaringan komunikasi berupa telepon, satelit dan radio link. Internet termasuk sarana berita yang melalui jalur komunikasi tersebut.

Hadirnya jaringan internet merupakan jawaban atas kebutuhan masyarakat terhadap informasi dan berita yang semakin meningkat sebagai kebutuhan harian karena dengan internet, informasi dapat diperoleh lebih cepat, efisien, dan efektif. Hal ini didukung oleh semakin banyak akses internet yang tersedia, jadi melalui akses internet banyak situs atau blog yang menyediakan berbagai macam artikel berita. Artikel berita yang akan diterbitkan, sebelumnya dipilah sesuai jenisnya oleh editor. Artikel berita jenisnya ada berbagai macam seperti politik, ekonomi, teknologi dan lain sebagainya. Pemilahan jenis artikel berita relatif mudah dilakukan secara manual oleh manusia, tetapi dengan berkembangnya teknologi, proses manual tersebut dapat dilakukan oleh komputer. Selain itu, hal ini dapat menjadi masalah baru, apakah komputer dapat menentukan kategori artikel berita seperti yang editor lakukan secara manual.

Text mining merupakan salah satu cara yang dapat mengatasi permasalahan tersebut. *Text mining* atau *text data mining* merupakan proses pengambilan data-data berupa *text* dari sebuah sumber. Dengan *text mining* dapat di cari kata-kata yang dapat mewakili isi dari artikel berita, yang kemudian akan di analisis untuk mengelompokkan artikel berita tersebut. Tahapan yang dilakukan dalam *text mining* ini adalah

tokenizing, filtering, *stemming*. Selain itu *text mining* melibatkan *information retrieval* (IR*), text analysis, information extraction* (IE), *clustering*, dan *data mining*.

Clustering dokumen merupakan proses klasifikasi dokumen ke dalam kelompok atau grup. Ada banyak metode klasifikasi, seperti *k-means, hierarchical clustering, KNN clustering, single pass clustering* dan masih banyak metode yang lain. *Single pass* efisien dan biasa digunakan dalam penanganan data dalam jumlah yang banyak.

Dalam proses *preprocessing* terdapat tahapan yang disebut *stemming*. Menurut Dani Yogatama *stemming* adalah proses pemotongan (pembuangan) *affix*, baik *prefix* maupun *suffix*, dari sebuah *term*. Hasil proses *stemming* berupa kata dasar. Dalam penelitian Magnus Rosell yaitu penelitian dalam bahasa swedia diperoleh hasil bahwa *stemming* meningkatkan hasil *clustering* sekitar 4% dibandingkan dengan tidak menggunakan *stemming*. Selain itu dinyatakan juga bahwa *stemming* meningkatkan precision and recall 15 dan 18 %. Dalam makalah yang dibuat oleh Fadilah Z. Tala menyatakan bahwa penggunaan *stemming* pada bahasa Inggris dan bahasa lain dari benua Eropa yang lebih komplek dari bahasa Inggris, menunjukkan bahwa terjadi peningkatan pada *precision* dan *recall*. Proses *stemming* mengurangi *term* yang harus diproses sehingga mempercepat waktu proses dan menghemat media penyimpanan.

Dalam penelitian Tuomo Korenius,dkk disebutkan bahwa *Stemming* dapat meningkatkan similaritas antara beberapa dokumen, dan menurut Mark Kantrowitz, dkk disebutkan bahwa peningkatan keakuratan *stemming* menghasilkan peningkatan *precision* yang signifikan, dimana *stemming* yang akurat adalah *stemming* yang menggunakan kamus kata. Dengan alasan ini maka *stemming* dapat meningkatkan keakuratan dalam clustering dokumen.

Atas latar belakang diatas maka dalam skripsi ini akan dilakukan penelitian pengaruh *stemming* terhadap hasil *precision* dan *recall* pada *clustering* berita berbahasa Indonesia dengan menggunakan *single pass clustering* yang pernah dilakukan oleh Agus Zaenal Arifin dan Ari Novan Setiono. Pada skripsi ini akan dilakukan kembali proses-proses yang ada pada jurnal Arifin Setiono tersebut, dan ditambah menggunakan algoritma *stemming* lain yaitu algoritma *stemming* Nazief dan Adriani.

Berdasarkan hal tersebut maka skripsi ini berjudul PENGARUH ALGORITMA *STEMMING* DALAM PEMILAHAN ARTIKEL BERITA MENGGUNAKAN METODE *SINGLE PASS CLUSTERING*.

1.2. Rumusan Masalah

Rumusan masalah dari penulisan skripsi ini adalah:

1. Bagaimana pengaruh algoritma *stemming* dalam pemilahan artikel berita menggunakan metode *single pass clustering*?

1.3. Batasan Masalah

Batasan masalah pada penulisan skripsi ini adalah:

1. Artikel yang digunakan sebagai masukan hanya artikel yang menggunakan bahasa Indonesia.
2. Format dokumen yang digunakan adalah teks (.txt).
3. Proses *stemming* yang dilakukan tanpa imbuhan sisipan.
4. Sistem dibangun dengan bahasa pemrograman Delphi dan diterapkan pada *Personal Computer* (PC).

1.4. Tujuan

Tujuan dari penulisan skripsi ini adalah:

1. Mengetahui pengaruh algoritma *stemming* dalam pemilahan artikel berita menggunakan metode *single pass clustering*.

1.5. Manfaat

Manfaat dari penulisan skripsi ini adalah:

1. Sebagai referensi mengenai *clustering* dokumen dengan menggunakan metode *single pass*.
2. Sebagai referensi mengenai algoritma *stemming* Arifin Setiono dan Nazief Adriani

1.6. Metode Penulisan

1. Studi Literatur

Studi literatur mencakup pengumpulan dan pemahaman teori dari berbagai referensi. Teori tersebut antara lain mengenai *text mining* beserta langkah-langkah *text mining* dan metode klasifikasi *single pass*

2. Pendefinisian dan Analisis Masalah

Pada tahap ini dilakukan analisis masalah dan metode yang akan diterapkan untuk mencari solusi yang tepat

3. Perancangan dan Implementasi Sistem

Membuat rancangan model perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil perancangan tersebut. Implementasinya berupa piranti lunak pemilahan artikel berita menggunakan klasifikasi *single pass*.

4. Pengujian dan Analisis Hasil Implementasi

Pengujian dilakukan untuk mendapatkan tingkat *recall* dan *precision* dari metode yang telah diterapkan. Kemudian menganalisis hasil implementasi tersebut dengan tujuan yang telah dirumuskan.

1.7. Sistematika Penulisan

Dalam penulisan hasil penelitian, dibuat suatu sistematika untuk menjaga alur informasi, yaitu:

1. BAB I. PENDAHULUAN

Bab ini berisi tentang latar belakang mengapa diambil judul PENGARUH ALGORITMA *STEMMING* DALAM PEMILAHAN ARTIKEL BERITA MENGGUNAKAN METODE *SINGLE PASS CLUSTERING*, rumusan masalah, batasan masalah, tujuan, manfaat penelitian, metode penulisan hasil penelitian, dan sistematika penyusunan hasil penelitian.

2. BAB II. TINJAUAN PUSTAKA

Bab ini berisi tentang hasil studi literatur yang telah dilakukan

3. BAB III. METODE PENELITIAN

Bab ini berisi tentang metode pelaksanaan penelitian secara terperinci

4. BAB IV. HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan dari penelitian yang telah dilakukan

5. BAB V. PENUTUP

Bab ini berisi tentang kesimpulan dari hasil dan pembahasan serta saran-saran untuk perbaikan penelitian-penelitian lebih lanjut.

BAB II
TINJAUAN PUSATAKA

2.1. Berita

Berita adalah cerita atau keterangan mengenai kejadian atau peristiwa yang hangat, berita juga berarti laporan pemberitahuan atau pengumuman. (Departemen Pendidikan Nasional, 2001)

2.2. Text Mining

2.2.1. Definisi Text Mining

Text mining memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen. (Raymond J. Money. CS, 2006)

Text mining secara luas didefinisikan sebagai proses mencari tahu secara intensif dimana pengguna berinteraksi dengan kumpulan dokumen sepanjang waktu dengan menggunakan serangkaian alat analisis. Pada cara yang sejalan dengan *data mining*, *text mining* berusaha mengutip informasi yang berguna dari sumber data melalui identifikasi dan eksplorasi pola yang menarik. Akan tetapi pada *text mining*, sumber data adalah kumpulan dokumen dan pola yang menarik tidak ditemukan pada *record database* yang terbentuk melainkan pada data kata per kata yang tidak terstruktur pada kumpulan dokumen tersebut. (Ronen Feldman, James Sanger, 2007)

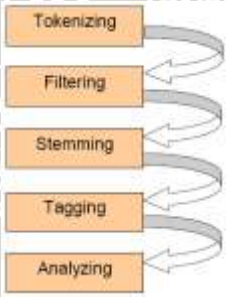
Tentu saja, *text mining* banyak berasal dari inspirasi dan arahan dari penelitian yang dapat berkembang dalam *data mining*. Karena itu, bukanlah hal yang mengejutkan jika menemukan bahwa sistem *text mining* dan *data mining* menunjukkan banyak kesamaan arsitektur tingkat tinggi. Sebagai contoh, kedua jenis dari sistem bergantung pada rutinitas *preprocessing*, algoritma *pattern-discovery*, dan elemen *presentation-layer* seperti alat visualisasi untuk meningkatkan *browsing* dari jawaban yang ada. Selanjutnya, *text mining* mengadopsi banyak jenis spesifik dari pola dalam operasi penemuan inti pengetahuan yang pertama dikenalkan dalam penelitian *data mining*. Karena *data mining* menganggap bahwa data telah tersimpan dalam format yang terstruktur, *preprocessing*-nya banyak yang difokuskan pada dua tugas penting :

penggosokan dan penormalan data serta pembuatan jumlah ekstensif dari tabel terkait. Sebaliknya, untuk sistem *text mining*, pusat operasi *preprocessing* pada identifikasi dan ekstraksi dari perwakilan fitur untuk dokumen berbahasa natural. Operasi *preprocessing* ini bertanggung jawab untuk mengubah data tidak terstruktur yang disimpan dalam kumpulan dokumen menjadi format lanjutan yang lebih terstruktur secara eksplisit, dimana perlu diperhatikan bahwa hal ini tidak relevan bagi kebanyakan sistem *data mining*. (Ronen Feldman, James Sanger, 2007)

Selain itu, karena pusat teks bahasa dasar terletak pada misinya, *text mining* juga mengacu pada kemajuan yang dibuat oleh mata pelajaran ilmu komputer lain yang berhubungan dengan penanganan bahasa dasar. Mungkin terutama, *text mining* memanfaatkan teknik dan metodologi dari bidang *information retrieval* (pencarian informasi), *information extraction* (penggalian informasi), dan kumpulan ilmu bahasa komputasi .(Ronen Feldman, James Sanger, 2007)

2.2.2. Tahapan-tahapan dalam *Text Mining*

Tahapan yang dilakukan secara umum dalam *text mining* ditunjukkan pada gambar 2.1:



Gambar 2.1 tahapan *text mining*

(Raymond J. Money. CS, 2006)

1. Tokenizing

Memberikan urutan karakter menetapkan unit dokumen, *tokenization* adalah pekerjaan memotong karakter pada dokumen menjadi bagian-bagian, yang disebut *token*, mungkin pada saat yang sama membuang karakter-karakter tertentu, seperti tanda baca. *Token* ini

pada umumnya sering disebut istilah atau kata, tetapi kadang-kadang penting untuk membuat jenis/perbedaan *token*. *Token* merupakan contoh dari urutan karakter dalam bebrapa dokumen tertentu. Jenis adalah kelas dari semua *token* yang berisi urutan karakter yang sama. Istilah adalah jenis yang diindeks dalam kamus sistem IR. (Christoper D. manning, Prabhakar Raghavan & Hinrich Schütze, 2007)

2. Filtering

Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil *token*. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *word list* (menyimpan kata penting). (Raymond J. Money. CS, 2006)

3. *Stemming* Bahasa Indonesia

a. Algoritma Arifin Setiono

Algoritma ini didahului dengan pembacaan tiap kata dari file sampel. Sehingga input dari algoritma ini adalah sebuah kata yang kemudian dilakukan :

a. Pemeriksaan semua kemungkinan bentuk kata. Setiap kata diasumsikan memiliki 2 Awalan (prefiks) dan 3 Akhiran (sufiks). Sehingga bentuknya menjadi :

Prefiks 1+Prefiks 2+Kata dasar+Sufiks 3+Sufiks 2+Sufiks 1

Seandainya kata tersebut tidak memiliki imbuhan sebanyak imbuhan di atas, maka imbuhan yang kosong diberi tanda x untuk prefiks dan diberi tanda xx untuk sufiks.

b. Pemotongan dilakukan secara berurutan sebagai berikut :
AW : awalan
AK : akhiran
KD : kata dasar
1) AW I, hasilnya disimpan pada p1
2) AW II, hasilnya disimpan pada p2
3) AK I, hasilnya disimpan pada s1
4) AK II, hasilnya disimpan pada s2
5) AK III, hasilnya disimpan pada s3

Pada setiap tahap pemotongan di atas diikuti dengan pemeriksaan di kamus apakah hasil pemotongan itu sudah berada dalam bentuk dasar. Kalau pemeriksaan ini berhasil maka proses dinyatakan selesai dan tidak perlu melanjutkan proses pemotongan imbuhan lainnya

Contoh pemenggalan kata "mempermainkannya"
Langkah 1 : Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AW I
Kata = permainkannya
Langkah 2 : Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AW II
Kata = mainkannya
Langkah 3 : Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AK I
Kata = mainkan
Langkah 4 : Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AK II
Kata = main
Langkah 5 : Cek apakah kata ada dalam kamus
Ya : Success
Tidak : lakukan pemotongan AK III.
Dalam hal ini AK III tidak ada, sehingga kata tidak diubah.
Kata = main
Langkah 6 : Cek apakah kata ada dalam kamus
Ya : Success
Tidak : "Kata tidak ditemukan"

c. Namun jika sampai pada pemotongan AK III, belum juga ditemukan di kamus, maka dilakukan proses kombinasi. KD yang dihasilkan dikombinasikan dengan imbuhan-imbuhannya dalam 12 konfigurasi berikut :

1) KD
2) KD + AK III
3) KD + AK III + AK II
4) KD + AK III + AK II + AK I
5) AW I + AW II + KD
6) AW I + AW II + KD + AK III
7) AW I + AW II + KD + AK III + AK II
8) AW I + AW II + KD + AK III + AKII + AKI
9) AW II + KD
10) AW II + KD + AK III
11) AW II + KD + AK III + AK II
12) AW II + KD + AK III + AK II + AK I

Sebenarnya kombinasi 1), 2), 3), 4), 8), dan 12) sudah diperiksa pada tahap sebelumnya, karena kombinasi ini adalah hasil pemotongan bertahap tersebut. Dengan demikian, kombinasi yang masih perlu dilakukan tinggal 6 yakni pada kombinasi-kombinasi yang belum dilakukan [5), 6), 7), 9), 10), dan 11)]. Tentunya bila hasil pemeriksaan suatu kombinasi adalah 'ada', maka pemeriksaan pada kombinasi lainnya sudah tidak diperlukan lagi. Pemeriksaan 12 kombinasi ini diperlukan, karena adanya fenomena *overstemming* pada algoritma pemotongan imbuhan. Kelemahan ini berakibat pada pemotongan bagian kata yang sebenarnya adalah milik kata dasar itu sendiri yang kebetulan mirip dengan salah satu jenis imbuhan yang ada. Dengan 12 kombinasi itu, pemotongan yang sudah terlanjur tersebut dapat dikembalikan sesuai posisinya. (Agus Zainal Arifin, Ari Novan Setiono, 2002)

b. Algoritma Nazief Adriani

Tahap-tahap algoritma Bobby Nazief dan Mirna Adriani :

1. Mencari kata yang akan di*stemming* dalam kamus. Jika ditemukan dalam kamus, dapat diasumsikan bahwa kata tersebut merupakan kata dasar, dan algoritma berhenti.
2. *Inflection suffixes* (*-lah*, *-kah*, *-ku*, *-mu*, atau *-nya*) dihapus. Jika berhasil dan akhirannya adalah partikel (*-lah* atau *-kah*), langkah ini diulangi untuk menghapus *inflectional possessive pronoun suffixes* (*-ku*, *-mu*, *-nya*).

3. *Derivation suffix* (*-i* atau *-an*) dihapus. Jika berhasil, ke langkah 4. Jika langkah 3 tidak berhasil:
   a. Jika *-an* telah dihapus, dan huruf terakhir adalah *-k*, kemudian *-k* juga dihapus dan dilanjutkan ke langkah 4. Jika gagal, dilanjutkan ke langkah 3b.
   b. Akhiran yang dihapus (*-i*, *-an*, atau *-kan*) dikembalikan.
4. *Derivational prefix* dihapus. Tahap ini mempunyai beberapa langkah:
   a. Jika akhiran telah dihapus pada langkah 3, kemudian kombinasi awalan-akhiran yang tidak diijinkan telah dicek menggunakan daftar pada tabel 2.1. Jika dalam kamus ditemukan maka algoritma berhenti.

Tabel 2.1 kombinasi awalan dan akhiran yang tidak diijinkan

Awalan	Akhiran yang tidak diijinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

   b. Jika awalan saat ini sama dengan awalan yang sebelumnya, maka algoritma berhenti.
   c. Jika tiga awalan sebelumnya telah dihapus, maka algoritma berhenti.
   d. Tipe awalan dijelaskan pada langkah di bawah ini:
      i. Jika awalan adalah *di-*, *ke-*, atau *se-*, maka tipe awalan berturut-turut adalah *di*, *ke*, atau *se*.
      ii. Jika awalan adalah *te-* (seperti pada tabel 2.2), *be-*, *me-*, atau *pe-*, maka diperlukan proses tambahan untuk menentukan tipe awalannya.

Tabel 2.2 menentukan tipe awalan untuk kata yang berawalan *te-*

karakter yang mengikuti				tipe awalan
set 1	set 2	set 3	set 4	
"-r-"	"-r-"	-	-	tidak ada
"-r-"	huruf vokal	-	-	ter-luluh
"-r-"	bukan (huruf vokal atau "-r-")	"-er-"	huruf vokal	ter
"-r-"	bukan (huruf vokal atau "-r-")	"-er-"	bukan huruf vokal	ter-
"-r-"	bukan (huruf vokal atau "-r-")	bukan "-er-"	-	ter
bukan (huruf vokal atau "-r-")	"-er-"	huruf vokal	-	tidak ada
bukan (huruf vokal atau "-r-")	"-er-"	bukan huruf vokal	-	te

iii. Jika dua karakter pertama bukan *di-*, *ke-*, *se-*, *te-*, *be-*, *me-* atau *pe-* maka algoritma berhenti

e. Jika tipe awalan adalah "none", maka algoritma berhenti. Jika tipe awalan tidak "none", maka tipe awalan ada di tabel 2.3 awalan yang ditemukan dihapus.

Tabel 2.3 menentukan awalan dari tipe awalan

Tipe awalan	Awalan yang harus dihapus
Di	di-
Ke	ke-
Se	se-
Te	te-
Ter	ter-
ter-luluh	ter-

f. Jika kata dasar tidak ditemukan, langkah 4 dilakukan untuk menghapus awalan berikutnya secara berulang.

g. Melakukan *recoding*. Langkah ini tergantung pada tipe awalan, dan dapat mengakibatkan awalan yang berbeda.

5. Jika semua langkah sudah selesai tetapi tidak berhasil, maka kata awal dianggap kata dasar.

(Jelita Asian, Hugh. E Williams & S.M.M Tahaghoghi, 2005)

Morfologi dari kata-kata Bahasa Indonesia dapat terdiri dari dua struktur yaitu infleksional dan derivasional. Infleksional adalah struktur yang paling sederhana yang diungkapkan dengan akhiran yang tidak mempengaruhi makna dasar asal kata yang mendasarinya. Akhiran infleksional dapat dibagi menjadi dua kelompok:

1. Akhiran *-lah*, *-kah*, *-pun*, *-tah*. Akhiran ini sebenarnya merupakan partikel atau kata-kata fungsional yang tidak mempunyai makna. Fungsinya dalam kata adalah untuk menekankan
2. Akhiran *-ku, -mu, -nya*. Akhiran ini, yang melekat pada kata, membentuk kata ganti posesif

Setiap kelompok akhiran 1 dan 2 dapat terjadi pada kata yang sama. Ketika keduanya digunakan, maka mengikuti urutan berikut kelompok akhiran kedua selalu mendahului kelompok akhiran pertama.

Sruktur derivasional Bahasa Indonesia terdiri dari awalan, akhiran dan sepasang kombinasi dari keduanya. Awalan yang paling sering adalah: *ber-, di-, ke-, meng-, peng-, per-, ter-*. Beberapa awalan seperti *ber-, meng-, peng-, per-, ter-* dapat muncul dalam beberapa bentuk yang berbeda. Bentuk dari masing-masing awalan ini tergantung pada karakter pertama dari kata yang terpasang awalan tersebut seperti pada tabel 2.4.

Tabel 2.4 variasi bentuk awalan

Awalan	Variasi bentuk	Aturan
meng	meng	+ huruf vocal|k|g|h
	meny	+ s
	mem	+ b|f|p
	men	+ c|d|j|t
	me	+ l|m|n|r|y|w
peng	peng	+ huruf vocal|k|g|h
	peny	+ s
	pem	+ b|f|p
	pen	+ c|d|j|t
	pe	+ l|m|n|r|y|w

ber	bel	+ ajar
	be	+ r|KVr
	ber	+setiap kata selain ketentuan bel dan be
per	pel	+ ajar
	pe	+ r|KVr
	per	+setiap kata selain ketentuan pel dan pe
ter	te	+ r
	ter	+ K|V, dimana K ≠ r

Tidak seperti struktur infleksional, ejaan kata dapat berubah ketika awalan ini dipasang. Akhiran derivasional yaitu: *-i, -kan, -an*. Berbeda dengan awalan, akhiran yang dipasang tidak pernah berubah ejaan asal dari kata asalnya.

Seperti disebutkan sebelumnya struktur derivasional juga mengakui *confixes*, dimana kombinasi dari awalan dan akhiran menempel mersama dalam sebuah kata untuk mendapatkan kata baru. Tidak semua kombinasi awalan dan akhiran dapat bergabung bersama untuk membentuk *confix*. Ada beberapa kombinasi awalan dan akhiran yang tidak diizinkan. Tabel 2.5 di bawah ini menunjukkan semua illegal *confix*.

Tabel 2.5 illegal *confix*

Awalan	**Akhiran**
ber	i
di	an
ke	i|kan
meng	an
peng	i|kan
ter	an

Awalan/akhiran dapat ditambahkan ke kata yang sudah *confix*/berawalan, yang menyebabkan struktur awalan ganda. Sama seperti kontruksi dari *confix*, tidak semua awalan/*confix* dapat ditambahkan ke beberapa kata confix/berawalan untuk membentuk awalan ganda. Terdapat aturan yang mengatur urutan awalan ganda ini, tetapi ada beberapa pengecualian terhadap aturan ini. Tabel 2.6 di bawah ini menunjukkan aturan urutan.

Tabel 2.6 Aturan urutan

Awalan 1	Awalan 2
meng	per
di	ber
ter	
ke	

(Fazdillah Z Tala, 2003).

4. *Tagging*

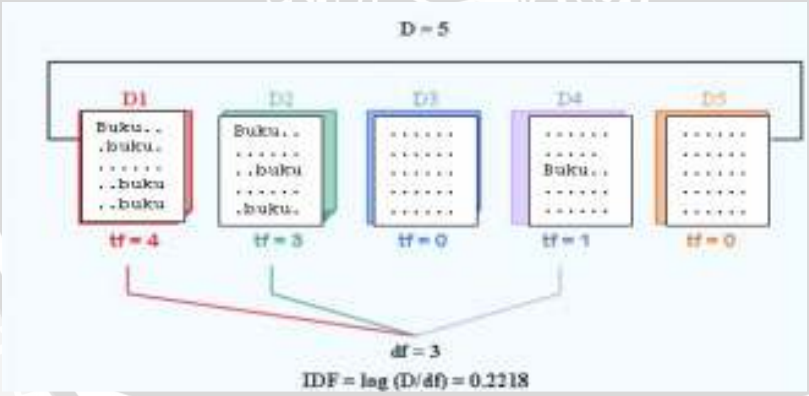
Tahap *tagging* adalah tahap mencari bentuk awal/*root* dari tiap kata lampau atau kata hasil *stemming*. Tahap ini hanya dipakai untuk teks berbahasa Inggris. Hal ini dikarenakan bahasa Indonesia tidak memiliki bentuk lampau. (Raymond J. Money. CS, 2006)

Dalam skripsi ini tahap *tagging* tidak dilakukan, karena skripsi ini mengolah berita berbahasa Indonesia dimana kosakatanya tidak memiliki bentuk lampau.

5. *Analyzing*

Tahap *analyzing* merupakan tahap penentuan seberapa jauh keterhubungan antar kata-kata antar dokumen yang ada. (Raymond J. Money. CS, 2006)

2.3. TF/IDF



Gambar 2.2 gambar ilustrasi TF-IDF

D1, D2, D3, D4, D5 = dokumen
tf = banyak kata yang dicari pada sebuah dokumen
D = total dokumen
df = banyak dokumen yang mengandung kata yang dicari
Formula yang digunakan untuk menghitung bobot (w) masing-masing terhadap kata kunci adalah

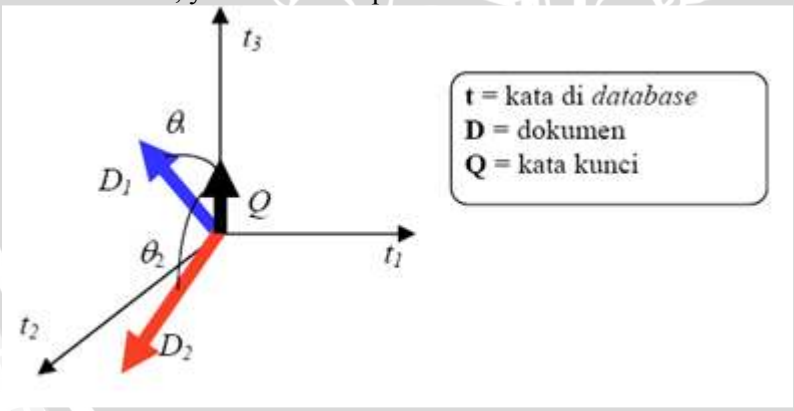
$$W_{d,t} = tf_{d,t} * IDF_t \quad (2.1)$$

Dimana:
D = dokumen ke-d
T = kata ke-t dari kata kunci
$W_{d,t}$ = bobot dokumen ke-d terhadap kata ke-t

Setelah bobot (w) masing-masing dokumen diketahui, maka dilakukan proses sorting/pengurutan dimana semakin besar nilai w, semakin besar tingkat similaritas dokumen tersebut terhadap kata yang dicari, demikian sebaliknya. (Raymond J. Money. CS, 2006)

2.4. *Vector Space Model*

Ide dari metode ini adalah dengan menghitung nilai *cosines* sudut dari dua *vector*, yaitu W dari tiap dokumen dan W dari kata kunci



Gambar 2.3 *vector space model*

Rumus: $$\text{CosSim}(dj, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$ (2.2)

Apabila studi kasus di atas dicari nilai *cosines* sudut antara *vector* masing-masing dokumen dengan kata kunci, maka hasil yang didapatkan akan lebih presisi. (Raymond J. Money. CS, 2006)

2.5. *Single Pass Clustering*

Clustering dokumen adalah aktivitas klasifikasi otomatis dokumen tanpa *supervise* ke dalam *clusters*/grup-grup. (A. K. Jain, dkk, 1999)

Teknik *single-pass clustering* pada dasarnya algoritma *greedy* yang selalu memberikan catatan bibliografi untuk *cluster* yang paling mirip. Karena setiap catatan bibliografi membaca hanya sekali, teknik *single-pass clustering* efisien dan mudah diimplementasikan. Namun demikian, teknik *single-pass clustering* membutuhkan kesamaan *threshold* ID yang ditentukan oleh pengguna. ID yang digunakan untuk menentukan apakah kesamaan antara *record* dan *cluster* cukup tinggi untuk menempatkan *record* pada *cluster*. Ketika ID kecil, setiap *cluster* dapat menampung *record* yang kurang sama. Dengan itu, jumlah yang *cluster* yang lebih kecil akan dihasilkan.

Meskipun teknik *single-pass clustering* itu sederhana, *single-pass* telah dikritik untuk kecenderungannya menghasilkan *cluster* yang besar di awal proses *clustering*. Hal ini karena *cluster* yang dihasilkan oleh teknik *single-pass clustering* tergantung pada urutan *record* bibliografi yang diproses. (Trevor Bench-Capon, dkk, 1999)

Algoritma *single pass clustering* dapat dilakukan dengan langkah-langkah sebagai berikut:

1. Masukkan (dokumen pertama) D1 representasi (*cluster* pertama) C1
2. Untuk (dokumen ke-i) dihitung kesamaan (similarity) dengan setiap wakil dari masing-masing *cluster*
3. Jika (*maximum similarity*) S_{max} lebih besar dari batas nilai (*threshold* value) S_T, tambahkan item kepada *cluster* yang bersesuaian dan hitung kembali representasi *cluster* baru.
4. Jika masih ada sebuah item Di yang belum dikelompokkan, kembali ke langkah ke-2

(Agus Zainal Arifin, Ari Novan Setiono, 2002)

2.6. Evaluasi

Performa dari identifikasi topik sebuah dokumen biasanya diukur menggunakan *Precision and Recall scores* (Hovy, 2003). Jika terdapat dua perbandingan hasil, hasil ringkasan sistem dan hasil ringkasan manusia, penilaian dilakukan pada seberapa dekat hasil ekstraksi topik yang dilakukan sistem dengan hasil ekstraksi yang dilakukan manusia. Pada skripsi ini akan dibandingkan hasil ekstraksi kalimat utama yang dilakukan sistem terhadap masing-masing tipe paragraf yang telah diketahui letak kalimat topiknya sesuai dengan tipe paragraf. *Precision* adalah nilai yang menandakan seberapa besar sistem berhasil mengekstrak kalimat topik yang sesuai, dan *recall* adalah nilai yang menandakan seberapa besar sistem gagal mengekstrak kalimat topik yang sesuai (Hovy, 2003).

Jika dimisalkan *correct* adalah jumlah kalimat topik yang berhasil diekstrak sistem dan sesuai dengan hasil ekstrak manusia, *wrong* adalah jumlah kalimat yang berhasil diekstrak sistem tetapi tidak diekstrak oleh manusia, dan *missed* adalah jumlah kalimat yang diekstrak manusia tetapi tidak diekstrak oleh sistem, maka rumus *Precision and Recall* ditunjukkan pada persamaan 2.4 dan 2.5 berikut (Hovy, 2003).

$$Precision = \frac{correct}{(correct + wrong)} \quad (2.3)$$

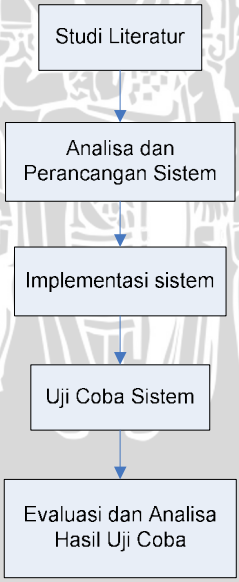
$$Recall = \frac{correct}{correct + missed} \quad (2.4)$$

BAB III
METODOLOGI DAN PERANCANGAN

Pada bab ini akan dibahas metode, rancangan yang digunakan dan langkah-langkah yang dilakukan dalam skripsi tentang pengaruh *stemming* terhadap hasil *precision* dan *recall* pada *clustering* berita dengan *single pass*.

Langkah-langkah yang dilakukan dalam penelitian ini meliputi:

1. Melakukan studi literatur tentang *stemming* Arifin Setiono dan Nazief Adriani serta algoritma *single pass clustering*
2. Menganalisa dan merancang sistem yang akan digunakan untuk meng-*cluster* dokumen
3. Membuat perangkat lunak berdasarkan analisa dan perancangan yang telah dilakukan
4. Memasukkan data *training* pada perangkat lunak yang telah dibuat dan kemudian memasukkan data *test* untuk melakukan uji coba pada perangkat lunak yang telah dibuat
5. Melakukan evaluasi hasil uji coba terhadap perangkat lunak yang telah dibuat



Gambar 3.1 Diagram Sistem

3.1. Perancangan Sistem Secara Umum

Secara umum, sistem berfungsi menerima inputan yang berupa dokumen berita dari *user* yang akan dipilah menjadi beberapa kelompok kategori berita, pemilahan dokumen akan menggunakan metode *single pass clustering*. Dalam sistem ini akan tersedia pilihan penggunaan *stemming* yang berbeda, yaitu *stemming* Arifin Setiono dan *stemming* Nazief Adriani.

Dalam skripsi ini akan dilakukan uji coba terhadap beberapa dokumen berita dengan jenis kategori berita yang berbeda, dimana dokumen tersebut sebelumnya telah diketahui kategorinya yang telah dipilah oleh ahli dari sumber dokumen berita tersebut diambil. Kemudian hasil dari uji tersebut akan diukur tingkat *precision* dan *recall* nya. Uji coba tersebut dilakukan dengan dua cara yaitu dengan mengganti *stemming* Arifin Setiono dan *stemming* Nazief Adriani untuk kemudian dibandingkan hasil *precision* dan *recall* nya.

Tabel 3.1 Tabel perancangan uji coba

percobaan	threshold	stemming Arifin			stemming Nazief			tanpa stemming		
		precision	recall	jumlah cluster	precision	recall	jumlah cluster	precision	recall	jumlah cluster

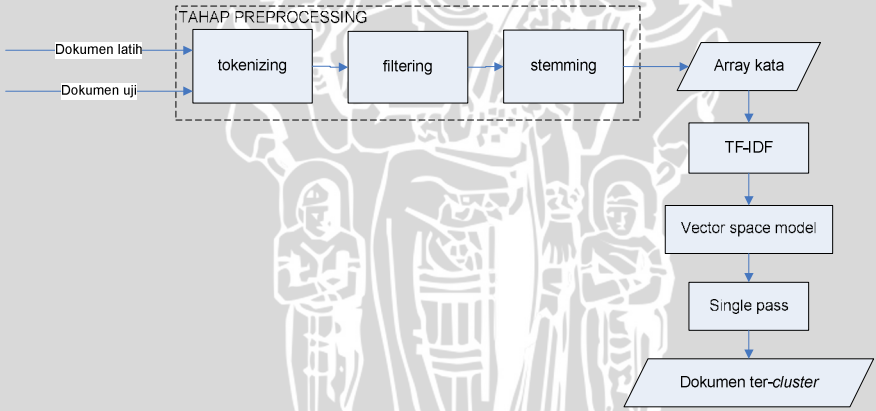
3.2. Perancangan Proses

Data masukan dari *user* berupa *file* dalam bentuk teks yang berekstensi .txt. Data masukan dalam sistem ini ada dua jenis, yaitu data latih dan data tes. Data masukan tersebut akan mengalami beberapa proses, yaitu proses *case folding*, *tokenizing*, *filtering* dengan menggunakan metode *stopword*, *stemming* bahasa Indonesia, pembobotan dengan metode TF-IDF, proses similaritas dengan metode *vector space model* dan kemudian *clustering* mnggunakan metode *single pass*.

3.2.1. *Preprocessing*

Pada tahapan ini, terdapat proses *case folding*, *tokenizing*, *filtering* dengan menggunakan metode *stopword*, dan *stemming*. *Case folding* merupakan proses merubah semua huruf yang terdapat dalam dokumen menjadi huruf kecil, *tokenizing* merupakan proses memecah dokumen menjadi kata per kata dan kemudian hasil dari proses *tokenizing* ini berupa kumpulan kata yang akan melalui proses *stop word filtering* dimana tiap kata hasil *tokenizing* yang terdapat dalam *stoplist* (daftar *stopword*) tidak dimasukkan ke dalam daftar kata penting. Kemudian dilakukan *stemming* untuk mendapatkan kata dasar dari kata-kata penting yang diperoleh setelah tahap *filtering*.

Proses *stemming* menggunakan kamus yang berisi kumpulan kata dasar asli yang tidak perlu melewati proses *stemming*. Kata yang tidak terdapat dalam kamus akan melewati proses *stemming*, setelah kata diproses maka kata tersebut dibandingkan kembali dengan kamus kata dasar dan jika kata tersebut tidak terdapat dalam kamus, maka kata tersebut akan ditambahkan pada kamus sebagai kata baru.



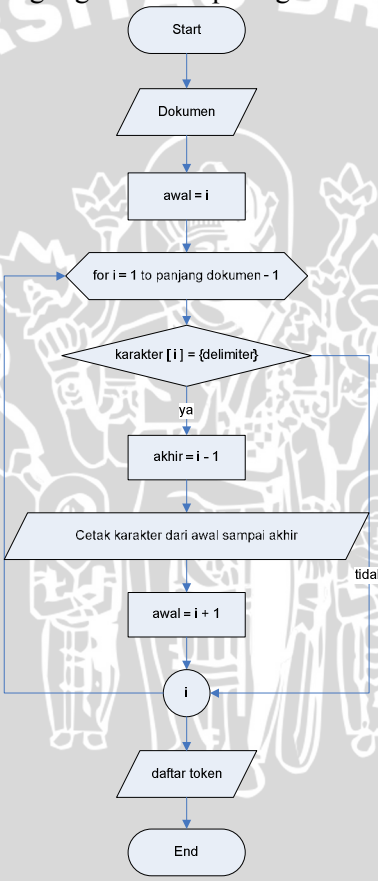
Gambar 3.2 Diagram Arsitektur Sistem

Dalam tahap preprocessing terdapat proses *tokenizing* dokumen menjadi kata per kata seperti digambarkan pada gambar 3.3.

Proses *filtering* dilakukan dengan menggunakan *stopword*, dimana terdapat pengecekan kata-kata apakah terdapat dalam *stoplist*,

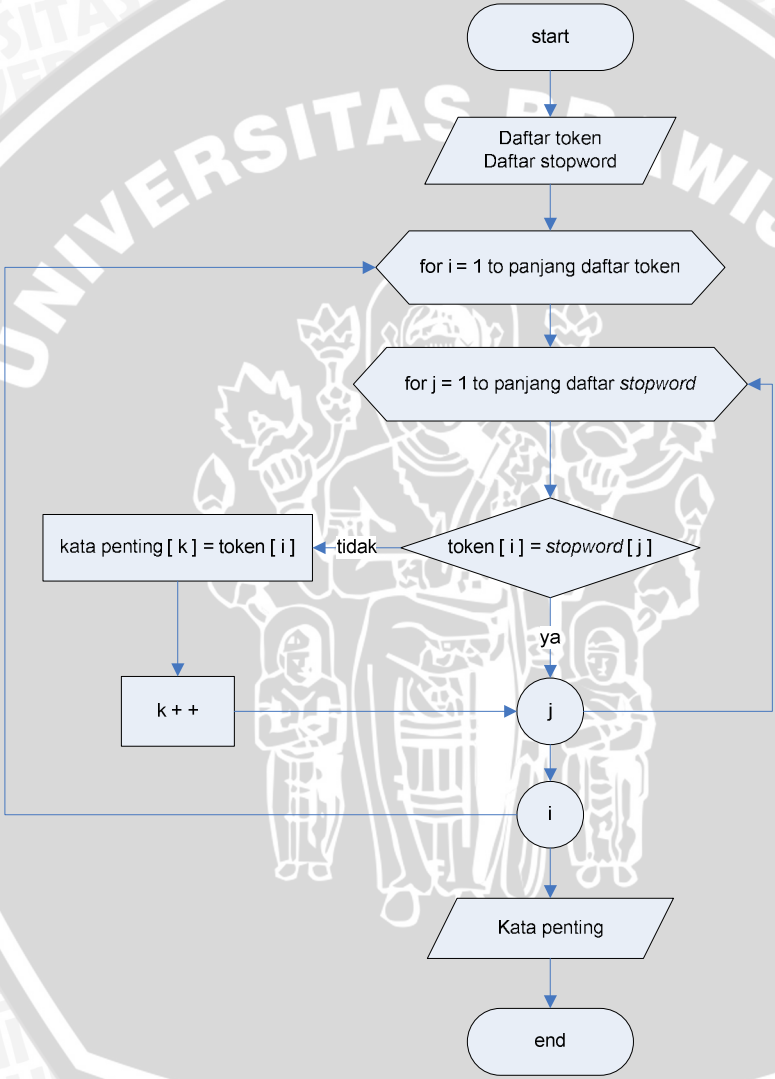
jika terdapat maka kata tersebut tidak dimasukkan dalam daftar kata-kata penting (dibuang). Proses ini digambarkan pada gambar 3.4.

Hasil kata penting yang diperoleh setelah melalui tahap *tokenizing* dan *filtering*, maka kata penting tersebut dicari kata dasarnya dengan memotong imbuhan yang terdapat pada kata tersebut, proses ini disebut dengan proses *stemming*. Pada skripsi ini proses stemming yang dilakukan ada dua metode yaitu *stemming* dengan metode Arifin Setiono yang digambarkan pada gambar 3.5 dan *stemming* dengan metode Nazief Adriani yang digambarkan pada gambar 3.11

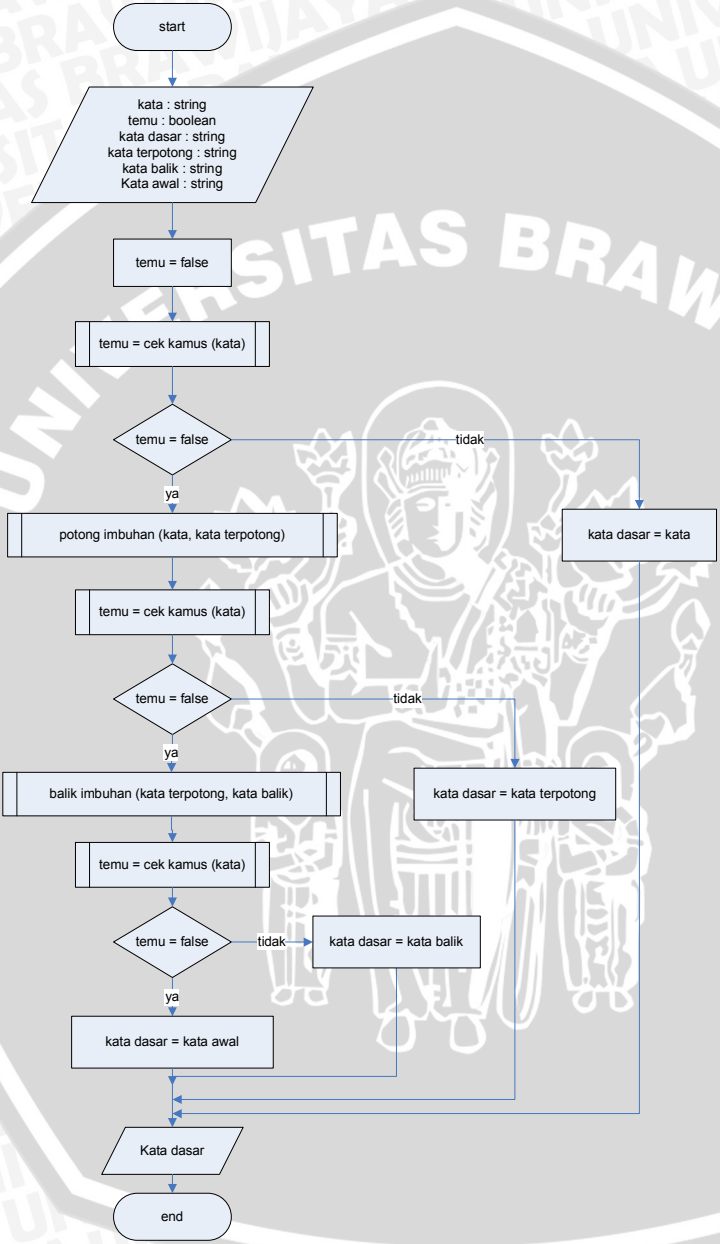


Gambar 3.3 *Flowchart tokenizing*

Proses filtering merupakan penyaringan kata dengan cara membandingkan kata-kata hasil *tokenizing* dengan daftar *stopword*, apabila sama dengan kata yang ada di dalam *stopword* maka kata tersebut akan di hapus dan bila tidak ada, maka kata akan di simpan dalam daftar kata penting.

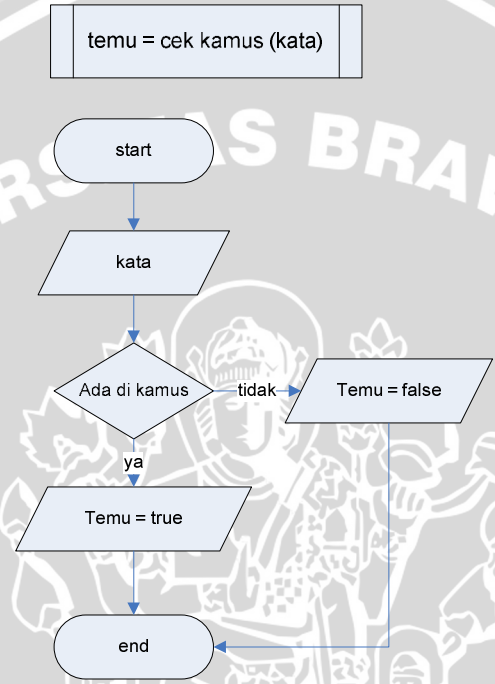


Gambar 3.4 *Flowchart Filtering*



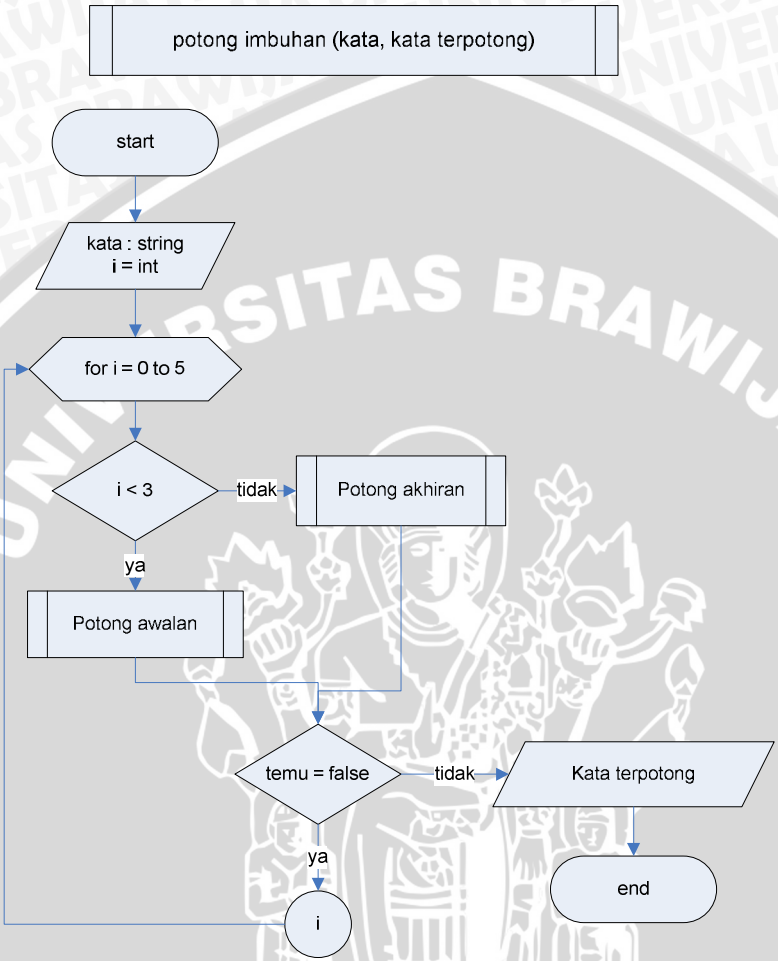
Gambar 3.5 *flowchart stemming* Arifin Setiono

Di dalam algoritma *stemming* Arifin Setiono, terdapat proses pengecekan kata dalam kamus kata dasar. Proses tersebut digambarkan pada gambar 3.6



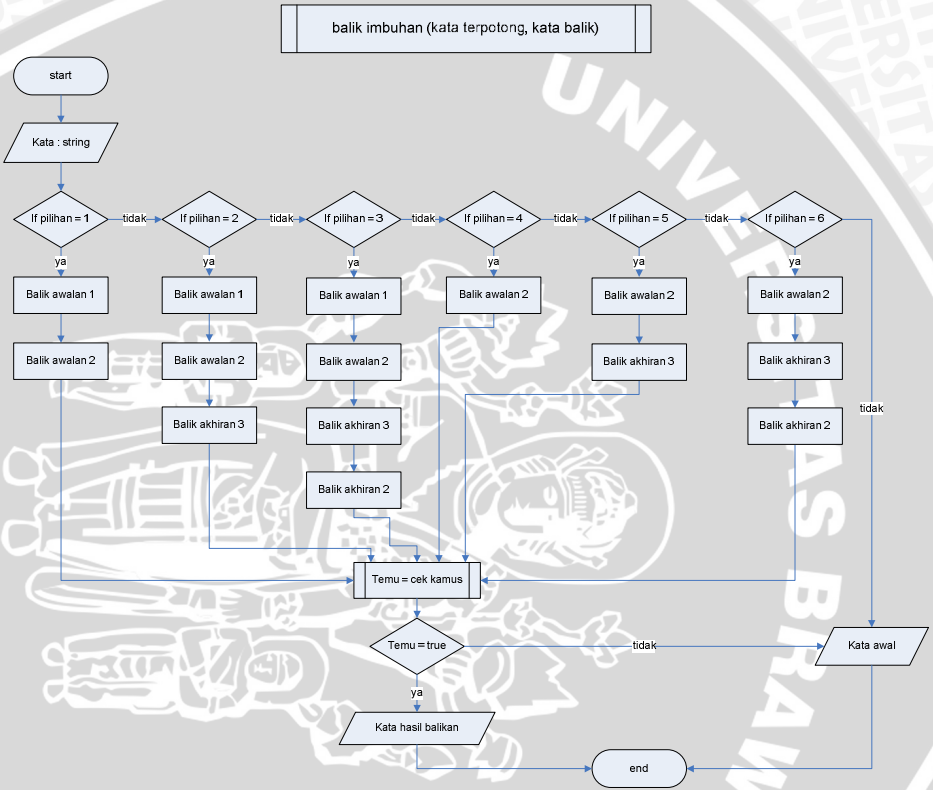
Gambar 3.6 *flowchart* cek kamus

Di dalam algoritma *stemming* Arifin Setiono, terdapat proses pemotongan imbuhan yang ditunjukkan pada gambar 3.7 dan terdapat pula proses kombinasi balik imbuhan yang ditunjukkan pada gambar 3.8. Di dalam proses pemotongan imbuhan ini ada dua proses lagi yaitu pemotongan imbuhan awalan yang dijelaskan dalam bentu pseudocode dan pemotongan imbuhan akhiran yang ditunjukkan pada gambar 3.10.



Gambar 3.7 *flowchart* potong imbuhan

Di dalam tahap potong imbuhan ini terdapat dua bagian proses lagi yaitu potong awalan dan potong akhiran, dimana dalam proses potong awalan aka nada beberapa proses pemotongan imbuhan awalan yang tidak memiliki aturan tertentu dan yang memiliki aturan-aturan jika bertemu dengan huruf-huruf tertentu.

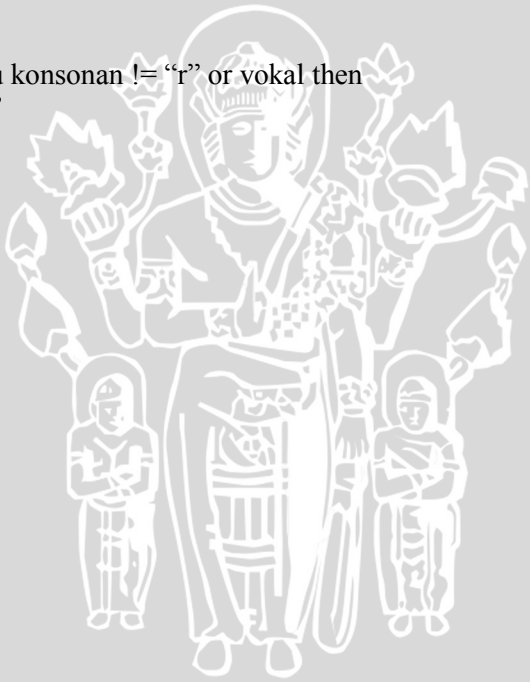


Gambar 3.8 *flowchart* balik imbuhan

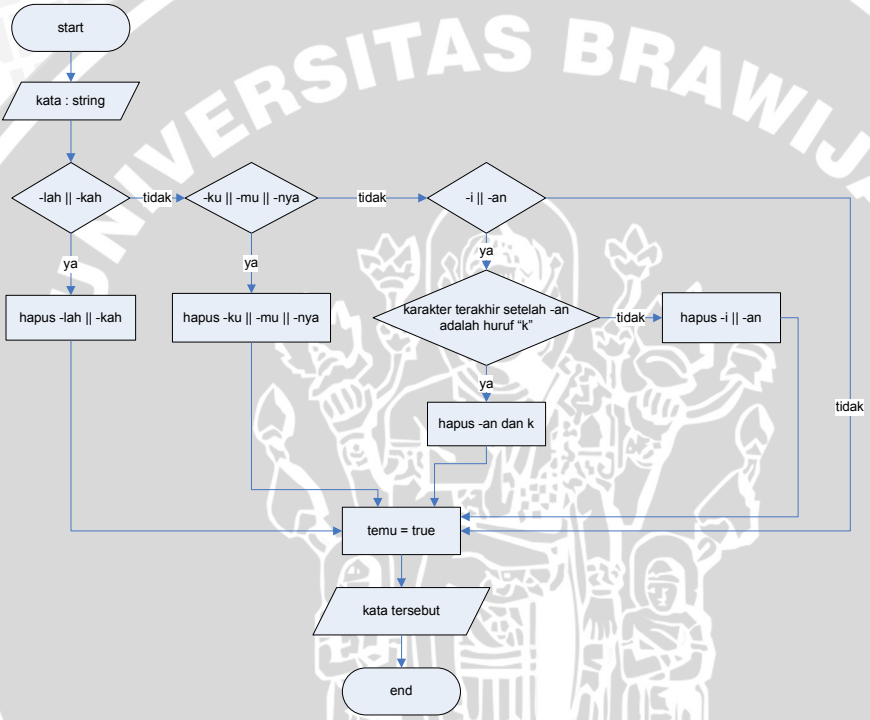
Tahap memotong imbuhan awalan merupakan salah satu tahap dalam proses pemotongan imbuhan. Dimana dalam tahap ini terdapat beberapa pengecekan jenis awalan, termasuk awalan yang melebur seperti pada tabel 2.4 yang menunjukkan daftar variasi bentuk awalan. Proses pemotongan awalan tersebut seperti yang ditunjukkan dalam algoritma di bawah ini.

```
Algoritma potong awalan
Begin
    If “di” or “ke” or “se” then
        Hapus “di” or “ke” or “se”
    Else if  “me” or “pe” then
        If “meng” or “peng” then
            Hapus “meng” or “peng” and ganti dengan vokal or “k” or “g” or
            “h”
            Cek kamus
        Else If “meny” or “peny” then
            Hapus “meny” or “peny” and ganti dengan “s”
            Cek kamus
        Else if “mem” or “pem” then
            Hapus “mem” or “pem” and ganti dengan “b” or “e” or “p”
            Cek kamus
        Else if “men” or “pen” then
            Hapus “men” or “pen” dang anti dengan “c” or “d” or “j” or “t”
            Cek kamus
        Else if “pelajar” then
            Hapus “pel”
            Cek kamus
        Else if “per” then
            Hapus “per”
            Cek kamus
        Else if bertemu “l” or “m” or “r” or “y” or “w” then
            Hapus “me” or “pe”
            Cek kamus
        End if
    Else if “be” then
        If bertemu “r” or susunan konsonan vokal “r” then
```

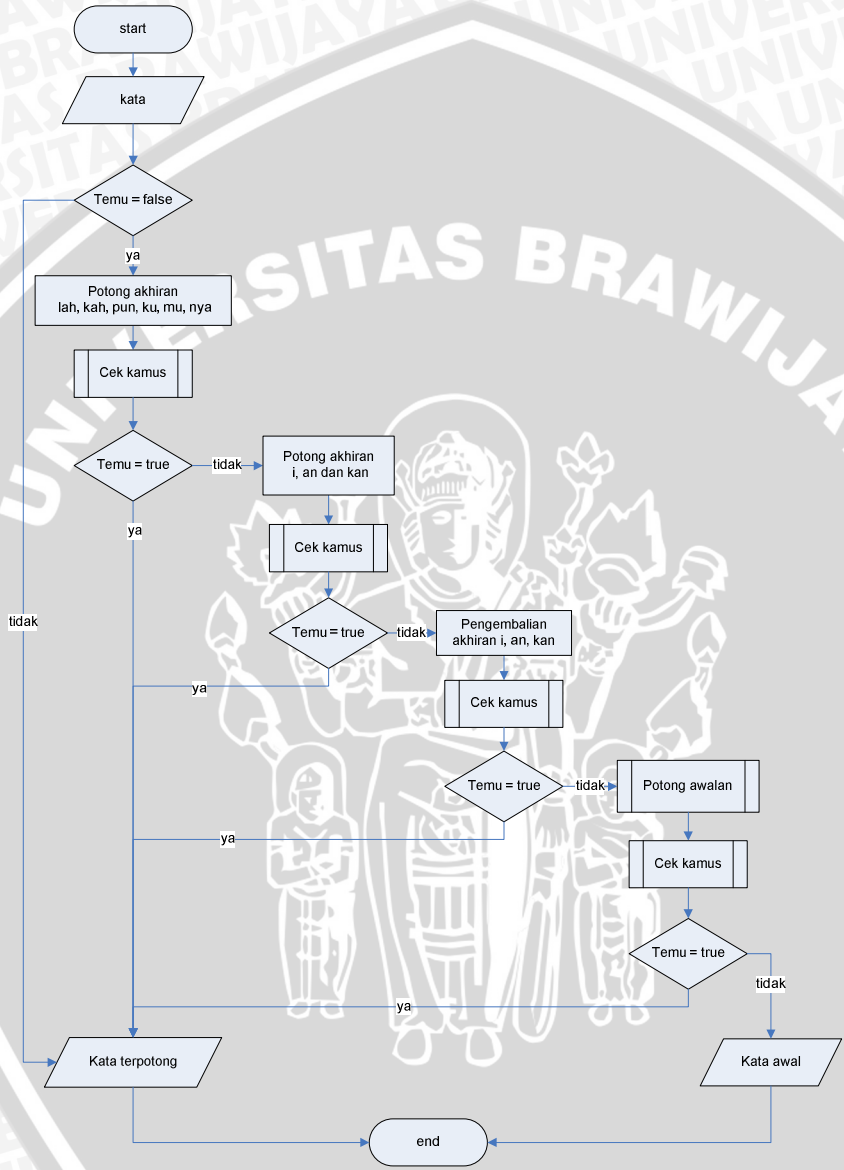
```
            Hapus “be”
            Cek kamus
        Else if “belajar” then
            Hapus “bel”
            Cek kamus
        Else if “ber”
            Hapus “ber”
            Cek kamus
        End if
    Else if “te” then
        If bertemu “r” then
            Hapus “te”
            Cek kamus
        Else if bertemu konsonan != “r” or vokal then
            Hapus “ter”
            Cek kamus
        End if
    endif
end
```



Dalam tahap pemotongan imbuhan akhiran ini, langkah pertama dengan melakukan pengecekan apakah akhiran adalah akhiran –lah atau –kah kemudian pengecekan akhiran –ku atau –mu atau -nya dan pengecekan akhiran -i atau –an. Tahap pemotongan imbuhan akhiran ini ditunjukkan pada gambar 3.10.



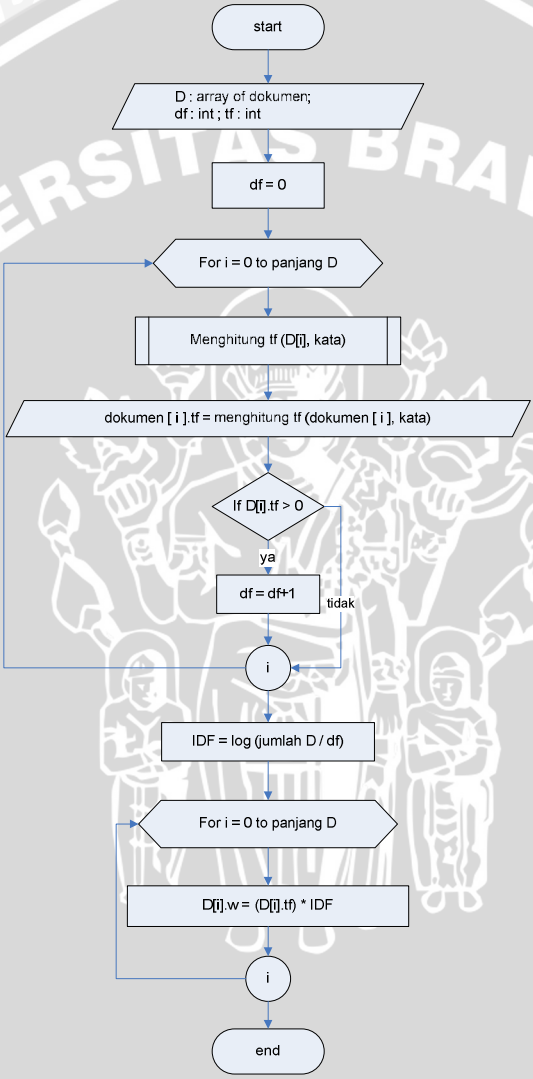
Gambar3.10 *flowchart* potong akhiran



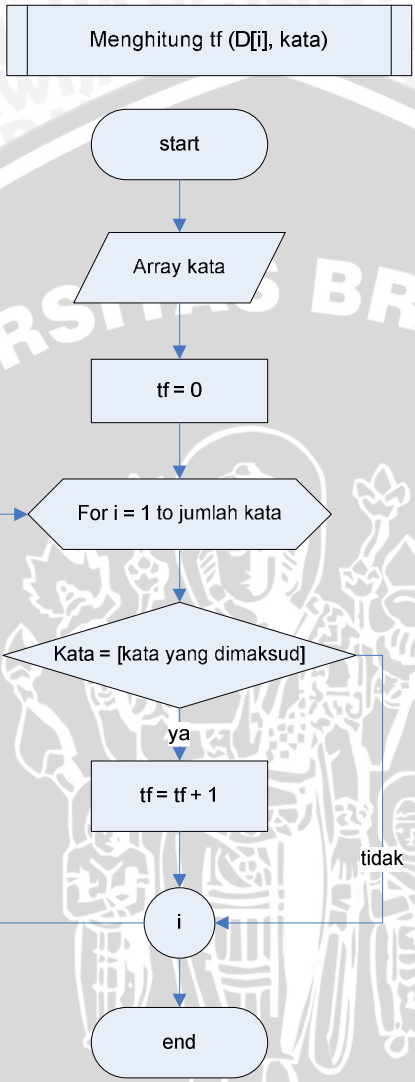
Gambar 3.11 *flowchart stemming* Nazief Adriani

3.2.2. TF-IDF

Setelah proses *preprocessing* maka akan dilakukan perhitungan pembobotan dengan menggunakan TF-IDF yang digambarkan pada gambar 3.12.



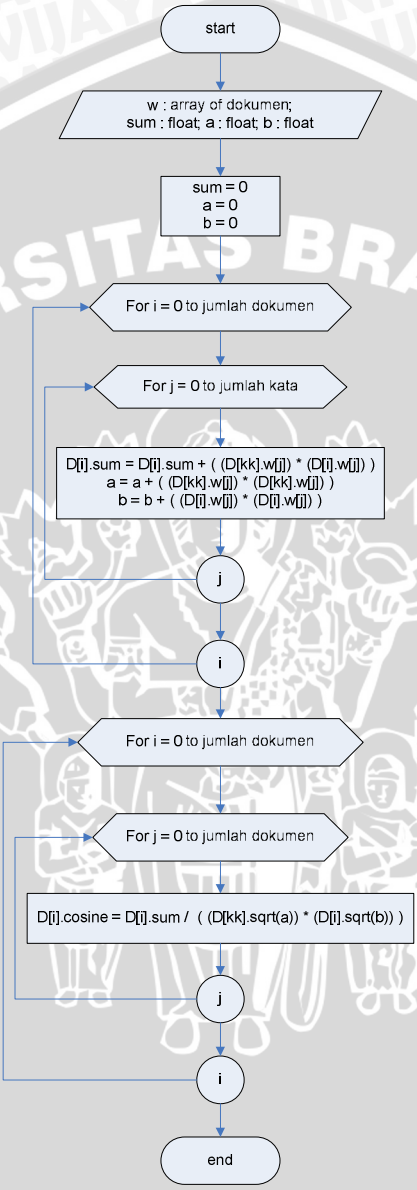
Gambar 3.12 *flowchart* TF IDF



Gambar 3.13 *flowchart* menghitung tf

Dari hasil matriks pembobotan TF-IDF maka akan dihitung similaritas antar dokumen dengan menggunakan *Vector Space Model* seperti yang digambarkan pada gambar 3.14

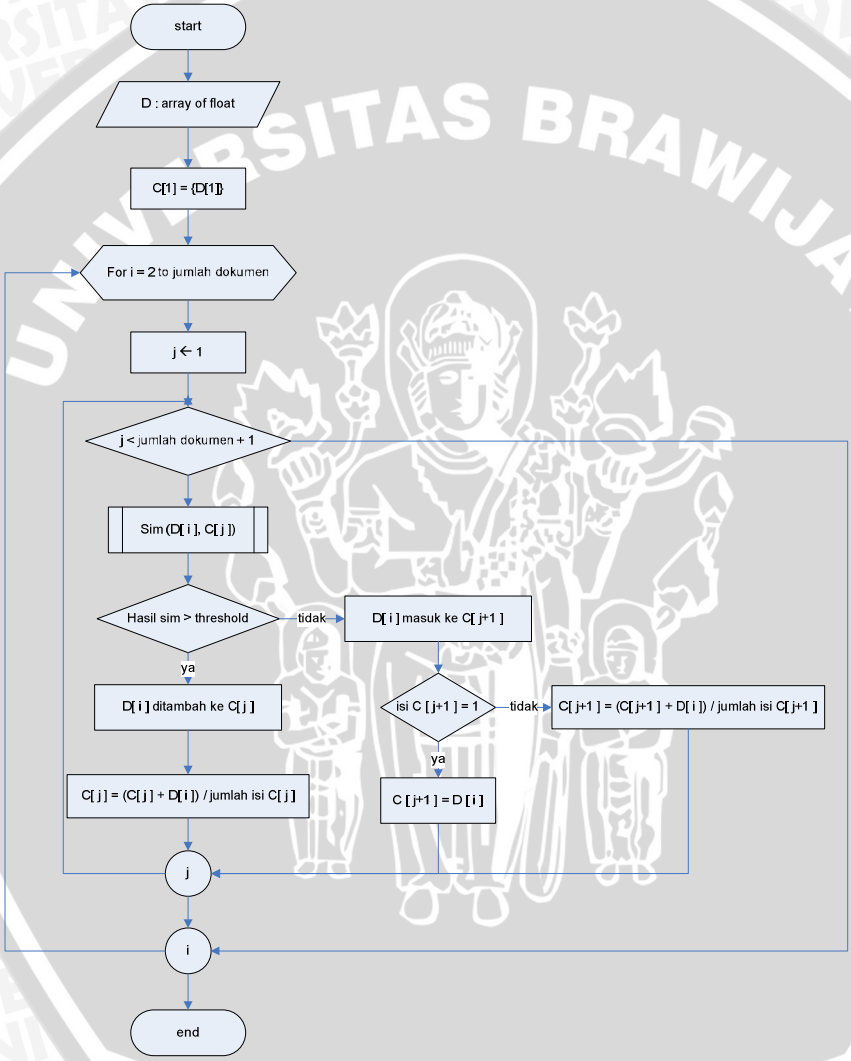
3.2.3. *Vector Space Model*



Gambar 3.14 *flowchart Vector Space Model*

Proses similaritas ini diperlukan dalam proses perhitungan *clustering* dengan menggunakan *Single Pass clustering*. Proses *clustering* ini digambarkan pada gambar 3.15

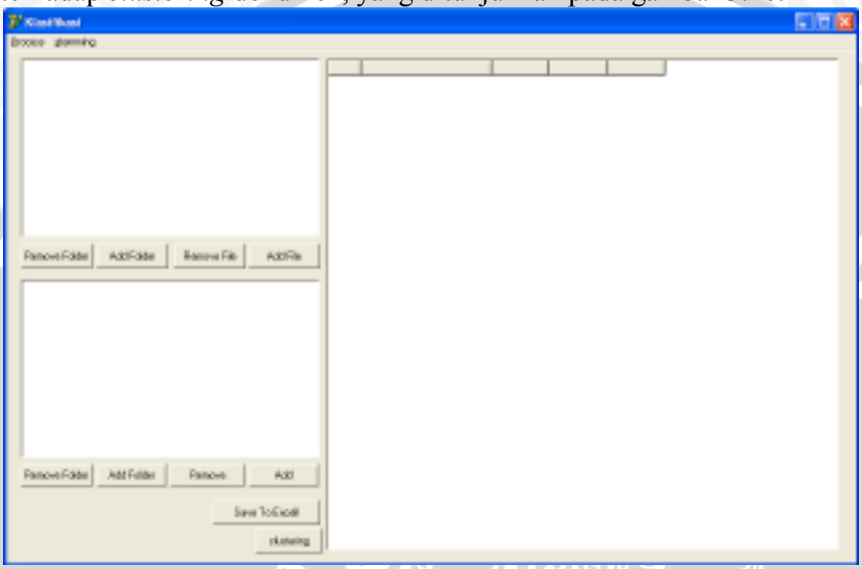
3.2.4. *Single Pass Clustering*



Gambar 3.15 *flowchart single pass*

3.3. Perancangan *user interface*

Pada subbab ini akan dijelaskan mengenai perancangan *user interface* dari sistem yang akan membantu menguji pengaruh *stemming* terhadap *clustering* dokumen, yang ditunjukkan pada gambar 3.16.



Gambar 3.16. perancangan *prototype user interface*

Fitur-fitur yang terdapat dalam *interface* ini adalah:

1. *Add folder* dan *add file* pada data latih: Fitur untuk memasukkan dokumen latih yang dipilih dari direktori *file*
2. *Add folder* dan *add file* pada data uji: Fitur untuk memasukkan dokumen uji yang dipilih dari direktori *file*
3. *Stemming* :Fitur untuk memilih *stemming* yang digunakan
4. *Process* : Fitur untuk memproses perhitungan
5. *Save to* excel: Fitur untuk menyimpan hasil perhitungan ke format excel
6. *Clustering*: Fitur untuk meng-*cluster* dokumen

3.4. Contoh Perhitungan

Misal diberikan dua buah dokumen latih dengan kategori yang berbeda dan tiga dokumen uji. Dokumen latih dan dokumen uji diambil dari *okezone.com*.

Dokumen latih 1

Harga si Emas Hitam Kian Anjlok!
Jum'at, 22 Januari 2010 - 07:57 wib
Candra Setya Santoso - Okezone

NEW YORK - Harga minyak mentah kian melemah ke level terendahnya dalam sebulan, pada perdagangan Kamis (21/1/2010) waktu setempat, seusai pemerintah Amerika Serikat (AS) mengajukan kebijakan baru di sektor perbankan.

Seperti dilansir dari AFP, Jumat (22/1/2010), proposal kebijakan perbankan yang diajukan Presiden Barack Obama memberi tekanan terhadap bursa AS dan para trader pun berpikir imbas apa yang akan terjadi pada pasar komoditas.

Rencananya, Presiden Obama untuk memperkecil dana-dana spekulasi oleh bank-bank komersil bisa membuat harga minyak menjadi mahal bagi para trader.

"Kalau kita bisa mereduksi dampak perbankan pada pasar komoditas, maka akan terjadi volatilitas yang minim pada harga bahan energi dan makanan," ungkap analis Mike masters.

Sementara itu, Goldman Sachs dan bank-bank besar lainnya telah membantu aliran dana spekulatif miliaran dolar AS ke pasar minyak dan gas alam, selama beberapa tahun terakhir.

Harga minyak untuk kontrak Maret turun USD1,66 menjadi USD76,08 per barel pada New York Mercantile Exchange (Nymex). Pada awal perdagangan, minyak sempat berada di level USD75,66, level terendah sejak 23 Desember tahun lalu. Sedangkan di London, harga minyak jenis Brent untuk kontrak Maret turun USD1,74 ke USD74,58 per barel pada ICE Futures.

Harga bahan-bahan energi anjlok di pagi hari usai Energy Information Administration melaporkan bahwa permintaan gasolin dan bahan bakar jet melemah selama beberapa pekan terakhir. Suplai gas alam turun lebih dari ekspektasi, menjadi 2,6 triliun kaki kubik dan saat ini lebih rendah dibanding rata-rata lima tahun.

Departemen Tenaga Kerja mengatakan, klaim insentif pengangguran naik 36 ribu pada pekan lalu menjadi 482 ribu.
(css)

Dokumen latih 2

Usus Sehat Berkat Apel
Kamis, 21 Januari 2010 - 15:38 wib
Adhini Amaliafitri - Okezone

BUAH apel diketahui mengandung pektin yang berfungsi melawan lemak dan menurunkan kolesterol. Serat di dalamnya juga membantu proses pencernaan. Belakangan, apel disinyalir mampu menyehatkan usus.

Berdasarkan berita yang dikutip Health Day, Kamis (21/2/2010), sebuah studi teranyar menemukan fakta bahwa pektin di dalam apel dapat meningkatkan bakteri baik dalam sistem pencernaan. Departemen Mikrobiologis University of Denmark's National Food Institute melakukan sebuah penelitian untuk menguji efek dari mengonsumsi apel. Mereka memberi makan beberapa ekor tikus yang sedang diet dengan buah apel dan berbagai macam produk berbasis apel, seperti jus apel dan puree apel (apel yang telah dihaluskan).

Kemudian, para peneliti memeriksa bakteri di dalam usus tikus untuk melihat apakah mengonsumsi apel dapat meningkatkan bakteri baik di dalamnya. Bakteri baik tersebut bermanfaat bagi kesehatan, juga mengurangi risiko terjangkit beberapa penyakit pencernaan.

"Dalam penelitian, kami menemukan bahwa tikus yang makan apel dengan kandungan pektin tinggi, jumlah bakteri baik di dalam ususnya meningkat hingga kesehatan ususnya juga meningkat," kata Andrea Wilcks, salah seorang peneliti pada jurnal BMC Microbiology edisi 20 Januari.

"Sepertinya, ketika apel dimakan teratur selama periode tertentu, bakteri baik ini membantu menghasilkan asam lemak rantai pendek yang memberikan kondisi ph seimbang untuk memastikan keseimbangan mikroorganisme yang menguntungkan. Mereka (bakteri baik) ini juga memproduksi zat kimia yang disebut butyrate yang merupakan bahan bakar penting untuk sel-sel dinding usus," kata Wilcke.

Penulis kajian menegaskan bahwa lebih banyak penelitian dibutuhkan untuk menentukan apakah penemuan pada tikus tersebut juga berlaku efektif untuk manusia.(ftr)

Dokumen uji 1

BCA Klaim Sudah Kembalikan 90% Dana Nasabah
Jum'at, 22 Januari 2010 - 17:29 wib
Andina Meryani - Okezone

JAKARTA - PT Bank Central Asia Tbk (BBCA) mengklaim sudah mengembalikan 90 persen dana nasabah yang mengalami kerugian akibat pembobolan ATM.

Adapun jumlah kerugian yang dilaporkan dari 200 nasabah yang mengalami pembobolan ATM tersebut mencapai sekira Rp5 miliar.

"Kami sudah mengganti sekira 90 persennya. Kami minta nasabah untuk tenang karena pasti akan kami ganti kalau memang terbukti akibat fraud," ujar Wakil Direktur BCA, Jahja Setiaatmadja, seusai konferensi pers

bersama Himbara, di Gedung Thamrin Bank Indonesia, Jakarta, Jumat (22/1/2010).

Dia pun mengakui kemungkinan besar kerugian yang diderita nasabah tersebut masih dapat bertambah karena kemungkinan masih ada nasabah yang belum melapor. Oleh karena itu, dia meminta kepada nasabahnya untuk segera menelpon call center BCA jika terdapat hal yang mencurigakan.

Bagi nasabah yang merasa dirugikan BCA berjanji untuk mengganti kerugian dalam tempo maksimal 3x24 jam setelah pelaporan.

"Bagi para nasabah ada keyakinan bahwa kerugian yang bukan karena kelalaian para nasabah pasti diganti. Kita mohon waktu 3x24 jam tapi kalau dalam pelaksanaan 1x24 jam ya bisa kita selesaikan," tandasnya.
(ade)

Dokumen uji 2

BI Perintahkan Bank Ganti Kerugian Nasabah
Jum'at, 22 Januari 2010 - 09:31 wib
Ade Hapsari Lestarini - Okezone

JAKARTA - Bank Indonesia (BI) telah memerintahkan bank untuk mengganti kerugian nasabah segera setelah proses verifikasi kerugian selesai dilakukan. Sehubungan dengan berkurangnya saldo rekening milik beberapa nasabah di bank tertentu oleh pihak yang tidak berhak.

Hal tersebut dikatakan Kepala Biro BI Difi A Johansyah, dalam siaran persnya, seperti dikutip dari situs resmi BI, di Jakarta, Jumat (22/1/2010).

Selain itu, pihak BI mengungkapkan bila dari laporan terakhir bank sampai dengan kemarin, tidak terdapat perubahan yang signifikan atas potensi kerugian material pada hari sebelumnya yaitu sekira Rp5 miliar dari empat bank.

Hal ini mengindikasikan bahwa bank telah mampu mengatasi permasalahan tersebut dan dengan demikian masyarakat tetap aman menggunakan kartu ATM sebagaimana biasa.

Selanjutnya, bank diminta mengganti kartu ATM nasabah yang dicurigai telah dicuri datanya dan melakukan edukasi kepada nasabah dan masyarakat mengenai keamanan penggunaan kartu ATM.

BI sendiri telah meminta bank untuk memeriksa dan menditeksi serta meningkatkan keamanan mesin-mesin ATM dan Electronic Data Capture (EDC) baik secara sistem maupun secara fisik atau lokasinya. Di samping itu, BI juga mengimbau agar nasabah selalu meneliti dan memperhatikan kondisi saat menggunakan mesin ATM maupun EDC.
(ade)

Dokumen uji 3

Olahraga Ampuh Jaga Emosi Wanita
Senin, 18 Januari 2010 - 17:56 wib
Dewi Arta - Okezone

MENJELANG menstruasi, emosi wanita sering kali tidak stabil. Tanpa sebab, wanita tiba-tiba marah dan lebih senang menyendiri.

Menurut dr Boy Abidin, perubahan emosi wanita yang tidak stabil menjelang menstruasi adalah wajar. Saat itu wanita mengalami perubahan peningkatan hormonal. Fase ini disebut pre-menstruasi syndrome, yang biasanya terjadi 7-10 hari sebelum menstruasi.

”Untuk mengendalikan emosi, bisa dengan olahraga secara teratur. Jadi perubahan hormonalnya tidak terlalu fluktuatif. Intinya menjadi pola hidup sehat,” saran genekolog yang aktif memberikan penyuluhan seputar perilaku seksual pada remaja.

Sependapat dengan dr Boy, psikolog Ratih Ibrahim menambahkan, emosi wanita bisa diatur dengan mengonsumsi nutrisi yang benar. Misalnya, mengonsumsi makanan 4 sehat 5 sempurna, dan suplemen kesehatan.

”Olahraga merupakan kunci utama menjaga emosi wanita,” tukasnya. (tty)

Setelah dokumen latih maupun uji melalu proses *preprocessing* maka akan dapat dihitung bobot masing *term* dengan menggunakan metode pembobotan TF-IDF dimana langkah awal menghitung TF (*Term Frequency*) yaitu jumlah kata dalam suatu dokumen, kemudian DF (*Document Frequency*) yaitu jumlah dokumen yang mengandung kata yang dimaksud, lalu IDF (*Inverse Document Frequency*) yaitu log dari hasil pembagian jumlah dokumen dan DF, dan yang terakhir menghitung W (*weight*) seperti pada persamaan 2.1

Tabel 3.1 perhitungan manual pembobotan

token	tf				df	D/df	IDF	W			
	kk	D1	D2	D3				kk	D1	D2	D3
jum	1	1	1	1	3	1	0	0	0	0	0
at	1	1	1	1	3	1	0	0	0	0	0
22	2	2	2	1	3	1	0	0	0	0	0
januari	1	1	1	1	3	1	0	0	0	0	0
2010	3	2	2	1	3	1	0	0	0	0	0

wib	1	1	1	1	3	1	0	0	0	0	0
okezone	1	1	1	1	3	1	0	0	0	0	0
rendah	3	0	0	1	1	3	0,477	1,431	0	0	0,477
1	2	2	1	0	2	1,5	0,176	0,352	0,352	0,176	0
waktu	1	1	0	1	2	1,5	0,176	0,176	0,176	0	0,176
usai	2	1	0	0	1	3	0,477	0,954	0,477	0	0
perintah	1	0	2	0	1	3	0,477	0,477	0	0,954	0
bank	7	2	9	0	2	1,5	0,176	1,233	0,352	1,585	0
jumat	1	1	1	0	2	1,5	0,176	0,176	0,176	0,176	0
beri	1	0	0	1	1	3	0,477	0,477	0	0	0,477
jadi	6	0	0	2	1	3	0,477	2,863	0	0	0,954
pasar	3	0	0	2	1	3	0,477	1,431	0	0	0,954
untuk	3	3	2	5	3	1	0	0	0	0	0
dana	3	2	0	0	1	3	0,477	1,431	0,954	0	0
bias	2	0	0	2	1	3	0,477	0,954	0	0	0,954
bahan	4	0	0	1	1	3	0,477	1,908	0	0	0,477
makan	1	0	0	2	1	3	0,477	0,477	0	0	0,954
ungkap	1	0	1	0	1	3	0,477	0,477	0	0,477	0
besar	1	1	0	1	2	1,5	0,176	0,176	0,176	0	0,176
bantu	1	0	0	1	1	3	0,477	0,477	0	0	0,477
miliar	1	1	1	0	2	1,5	0,176	0,176	0,176	0,176	0
alam	2	2	0	0	1	3	0,477	0,954	0,954	0	0
tahun	3	0	0	1	1	3	0,477	1,431	0	0	0,477
akhir	2	0	1	0	1	3	0,477	0,954	0	0,477	0
turun	3	0	0	1	1	3	0,477	1,431	0	0	0,477
hari	1	0	1	0	1	3	0,477	0,477	0	0,477	0
lapor	1	3	1	0	2	1,5	0,176	0,176	0,528	0,176	0
minta	1	2	2	0	2	1,5	0,176	0,176	0,352	0,352	0
kata	1	0	1	0	1	3	0,477	0,477	0	0,477	0
klaim	1	2	0	0	1	3	0,477	0,477	0,954	0	0
ribu	2	0	0	1	1	3	0,477	0,954	0	0	0,477
bca	0	4	0	0	1	3	0,477	0	1,908	0	0
kembali	0	3	0	0	1	3	0,477	0	1,431	0	0
90	0	3	0	0	1	3	0,477	0	1,431	0	0
nasabah	0	10	6	0	2	1,5	0,176	0	1,761	1,057	0
17	0	1	0	0	1	3	0,477	0	0,477	0	0

29	0	1	0	0	1	3	0,477	0	0,477	0	0
andina	0	1	0	0	1	3	0,477	0	0,477	0	0
meryani	0	1	0	0	1	3	0,477	0	0,477	0	0
jakarta	0	2	2	0	2	1,5	0,176	0	0,352	0,352	0
pt	0	1	0	0	1	3	0,477	0	0,477	0	0
central	0	1	0	0	1	3	0,477	0	0,477	0	0
asia	0	1	0	3	2	1,5	0,176	0	0,176	0	0,528
tbk	0	1	0	0	1	3	0,477	0	0,477	0	0
bbca	0	1	0	0	1	3	0,477	0	0,477	0	0
persen	0	2	0	2	2	1,5	0,176	0	0,352	0	0,352
rugi	0	6	4	0	2	1,5	0,176	0	1,057	0,704	0
akibat	0	2	0	1	2	1,5	0,176	0	0,352	0	0,176
bobol	0	2	0	0	1	3	0,477	0	0,954	0	0
atm	0	2	5	0	2	1,5	0,176	0	0,352	0,88	0
jumlah	0	1	0	0	1	3	0,477	0	0,477	0	0
200	0	1	0	0	1	3	0,477	0	0,477	0	0
capai	0	1	0	0	1	3	0,477	0	0,477	0	0
kira	0	2	1	0	2	1,5	0,176	0	0,352	0,176	0
rp5	0	1	1	0	2	1,5	0,176	0	0,176	0,176	0
ganti	0	4	3	0	2	1,5	0,176	0	0,704	0,528	0
tenang	0	1	0	0	1	3	0,477	0	0,477	0	0
bukti	0	1	0	2	2	1,5	0,176	0	0,176	0	0,352
fraud	0	1	0	0	1	3	0,477	0	0,477	0	0
ujar	0	1	0	0	1	3	0,477	0	0,477	0	0
wakil	0	1	0	0	1	3	0,477	0	0,477	0	0
direktur	0	1	0	0	1	3	0,477	0	0,477	0	0
jahja	0	1	0	0	1	3	0,477	0	0,477	0	0
setiaatmadja	0	1	0	0	1	3	0,477	0	0,477	0	0
konferensi	0	1	0	0	1	3	0,477	0	0,477	0	0
pers	0	1	1	0	2	1,5	0,176	0	0,176	0,176	0
himbara	0	1	0	0	1	3	0,477	0	0,477	0	0
gedung	0	1	0	0	1	3	0,477	0	0,477	0	0
thamrin	0	1	0	0	1	3	0,477	0	0,477	0	0
indonesia	0	1	1	1	3	1	0	0	0	0	0
aku	0	1	0	0	1	3	0,477	0	0,477	0	0
mungkin	0	2	0	0	1	3	0,477	0	0,954	0	0

derita	0	1	0	3	2	1,5	0,176	0	0,176	0	0,528
tambah	0	1	0	2	2	1,5	0,176	0	0,176	0	0,352
telpon	0	1	0	0	1	3	0,477	0	0,477	0	0
call	0	1	0	0	1	3	0,477	0	0,477	0	0
center	0	1	0	0	1	3	0,477	0	0,477	0	0
dapat	0	1	1	2	3	1	0	0	0	0	0
curiga	0	1	1	0	2	1,5	0,176	0	0,176	0,176	0
rasa	0	1	0	1	2	1,5	0,176	0	0,176	0	0,176
janji	0	1	0	0	1	3	0,477	0	0,477	0	0
tempo	0	1	0	0	1	3	0,477	0	0,477	0	0
maksimal	0	1	0	0	1	3	0,477	0	0,477	0	0
3	0	2	0	0	1	3	0,477	0	0,954	0	0
24	0	3	0	0	1	3	0,477	0	1,431	0	0
jam	0	3	0	0	1	3	0,477	0	1,431	0	0
yakin	0	1	0	0	1	3	0,477	0	0,477	0	0
lalai	0	1	0	0	1	3	0,477	0	0,477	0	0
mohon	0	1	0	0	1	3	0,477	0	0,477	0	0
laksana	0	1	0	0	1	3	0,477	0	0,477	0	0
ya	0	1	0	0	1	3	0,477	0	0,477	0	0
selesai	0	1	0	0	1	3	0,477	0	0,477	0	0
tandas	0	1	0	0	1	3	0,477	0	0,477	0	0
ade	0	1	2	0	2	1,5	0,176	0	0,176	0,352	0
bi	0	0	7	0	1	3	0,477	0	0	3,34	0
09	0	0	1	0	1	3	0,477	0	0	0,477	0
31	0	0	1	0	1	3	0,477	0	0	0,477	0
hapsari	0	0	1	0	1	3	0,477	0	0	0,477	0
lestarini	0	0	1	0	1	3	0,477	0	0	0,477	0
proses	0	0	1	0	1	3	0,477	0	0	0,477	0
verifikasi	0	0	1	0	1	3	0,477	0	0	0,477	0
laku	0	0	2	0	1	3	0,477	0	0	0,954	0
hubung	0	0	1	0	1	3	0,477	0	0	0,477	0
kurang	0	0	1	3	2	1,5	0,176	0	0	0,176	0,528
saldo	0	0	1	0	1	3	0,477	0	0	0,477	0
rekening	0	0	1	0	1	3	0,477	0	0	0,477	0
milik	0	0	1	2	2	1,5	0,176	0	0	0,176	0,352
pihak	0	0	2	0	1	3	0,477	0	0	0,954	0

berhak	0	0	1	0	1	3	0,477	0	0	0,477	0
kepala	0	0	1	0	1	3	0,477	0	0	0,477	0
biro	0	0	1	0	1	3	0,477	0	0	0,477	0
difi	0	0	1	0	1	3	0,477	0	0	0,477	0
a	0	0	1	0	1	3	0,477	0	0	0,477	0
johansyah	0	0	1	0	1	3	0,477	0	0	0,477	0
siar	0	0	1	0	1	3	0,477	0	0	0,477	0
kutip	0	0	1	0	1	3	0,477	0	0	0,477	0
situs	0	0	1	0	1	3	0,477	0	0	0,477	0
resmi	0	0	1	0	1	3	0,477	0	0	0,477	0
kemarin	0	0	1	0	1	3	0,477	0	0	0,477	0
ubah	0	0	1	0	1	3	0,477	0	0	0,477	0
signifikan	0	0	1	0	1	3	0,477	0	0	0,477	0
atas	0	0	2	0	1	3	0,477	0	0	0,954	0
potensi	0	0	1	0	1	3	0,477	0	0	0,477	0
material	0	0	1	0	1	3	0,477	0	0	0,477	0
empat	0	0	1	0	1	3	0,477	0	0	0,477	0
indikasi	0	0	1	0	1	3	0,477	0	0	0,477	0
masalah	0	0	1	0	1	3	0,477	0	0	0,477	0
masyarakat	0	0	2	1	2	1,5	0,176	0	0	0,352	0,176
tetap	0	0	1	0	1	3	0,477	0	0	0,477	0
aman	0	0	3	0	1	3	0,477	0	0	1,431	0
guna	0	0	3	0	1	3	0,477	0	0	1,431	0
kartu	0	0	3	0	1	3	0,477	0	0	1,431	0
lanjut	0	0	1	0	1	3	0,477	0	0	0,477	0
curi	0	0	1	0	1	3	0,477	0	0	0,477	0
data	0	0	2	0	1	3	0,477	0	0	0,954	0
edukasi	0	0	1	0	1	3	0,477	0	0	0,477	0
kena	0	0	1	0	1	3	0,477	0	0	0,477	0
periksa	0	0	1	0	1	3	0,477	0	0	0,477	0
diteksi	0	0	1	0	1	3	0,477	0	0	0,477	0
tingkat	0	0	1	3	2	1,5	0,176	0	0	0,176	0,528
mesin	0	0	3	0	1	3	0,477	0	0	1,431	0
electronic	0	0	1	0	1	3	0,477	0	0	0,477	0
capture	0	0	1	0	1	3	0,477	0	0	0,477	0
edc	0	0	2	0	1	3	0,477	0	0	0,954	0

baik	0	0	1	1	2	1,5	0,176	0	0	0,176	0,176
cara	0	0	2	2	2	1,5	0,176	0	0	0,352	0,352
sistem	0	0	1	1	2	1,5	0,176	0	0	0,176	0,176
fisik	0	0	1	0	1	3	0,477	0	0	0,477	0
lokasi	0	0	1	0	1	3	0,477	0	0	0,477	0
samping	0	0	1	0	1	3	0,477	0	0	0,477	0
imbau	0	0	1	0	1	3	0,477	0	0	0,477	0
teliti	0	0	1	0	1	3	0,477	0	0	0,477	0
perhati	0	0	1	0	1	3	0,477	0	0	0,477	0
kondisi	0	0	1	0	1	3	0,477	0	0	0,477	0
juta	0	0	0	1	1	3	0,477	0	0	0	0,477
manfaat	0	0	0	6	1	3	0,477	0	0	0	2,863
sehat	0	0	0	4	1	3	0,477	0	0	0	1,908
10	0	0	0	1	1	3	0,477	0	0	0	0,477
dewi	0	0	0	1	1	3	0,477	0	0	0	0,477
arta	0	0	0	1	1	3	0,477	0	0	0	0,477
salah	0	0	0	1	1	3	0,477	0	0	0	0,477
satu	0	0	0	2	1	3	0,477	0	0	0	0,954
konsumsi	0	0	0	2	1	3	0,477	0	0	0	0,954
gandrung	0	0	0	1	1	3	0,477	0	0	0	0,477
enak	0	0	0	1	1	3	0,477	0	0	0	0,477
penganan	0	0	0	1	1	3	0,477	0	0	0	0,477
utama	0	0	0	1	1	3	0,477	0	0	0	0,477
tampil	0	0	0	2	1	3	0,477	0	0	0	0,954
sederhana	0	0	0	1	1	3	0,477	0	0	0	0,477
lembut	0	0	0	1	1	3	0,477	0	0	0	0,477
hadir	0	0	0	1	1	3	0,477	0	0	0	0,477
bagai	0	0	0	3	1	3	0,477	0	0	0	1,431
warna	0	0	0	2	1	3	0,477	0	0	0	0,954
mulai	0	0	0	1	1	3	0,477	0	0	0	0,477
putih	0	0	0	1	1	3	0,477	0	0	0	0,477
kuning	0	0	0	1	1	3	0,477	0	0	0	0,477
cokelat	0	0	0	1	1	3	0,477	0	0	0	0,477
luar	0	0	0	1	1	3	0,477	0	0	0	0,477
tarik	0	0	0	1	1	3	0,477	0	0	0	0,477
kandung	0	0	0	4	1	3	0,477	0	0	0	1,908

nutrisi	0	0	0	1	1	3	0,477	0	0	0	0,477
masuk	0	0	0	1	1	3	0,477	0	0	0	0,477
kaya	0	0	0	3	1	3	0,477	0	0	0	1,431
protein	0	0	0	2	1	3	0,477	0	0	0	0,954
kalori	0	0	0	1	1	3	0,477	0	0	0	0,477
bebas	0	0	0	1	1	3	0,477	0	0	0	0,477
kolesterol	0	0	0	4	1	3	0,477	0	0	0	1,908
asal	0	0	0	1	1	3	0,477	0	0	0	0,477
china	0	0	0	1	1	3	0,477	0	0	0	0,477
umum	0	0	0	1	1	3	0,477	0	0	0	0,477
ikut	0	0	0	1	1	3	0,477	0	0	0	0,477
serta	0	0	0	1	1	3	0,477	0	0	0	0,477
menu	0	0	0	1	1	3	0,477	0	0	0	0,477
masak	0	0	0	1	1	3	0,477	0	0	0	0,477
lunak	0	0	0	1	1	3	0,477	0	0	0	0,477
campur	0	0	0	1	1	3	0,477	0	0	0	0,477
sentuh	0	0	0	1	1	3	0,477	0	0	0	0,477
tepat	0	0	0	1	1	3	0,477	0	0	0	0,477
hidang	0	0	0	1	1	3	0,477	0	0	0	0,477
sebut	0	0	0	1	1	3	0,477	0	0	0	0,477
keju	0	0	0	1	1	3	0,477	0	0	0	0,477
kedelai	0	0	0	1	1	3	0,477	0	0	0	0,477
medis	0	0	0	1	1	3	0,477	0	0	0	0,477
mampu	0	0	0	1	1	3	0,477	0	0	0	0,477
kadar	0	0	0	1	1	3	0,477	0	0	0	0,477
30	0	0	0	1	1	3	0,477	0	0	0	0,477
ldl	0	0	0	1	1	3	0,477	0	0	0	0,477
jahat	0	0	0	1	1	3	0,477	0	0	0	0,477
35	0	0	0	1	1	3	0,477	0	0	0	0,477
40	0	0	0	1	1	3	0,477	0	0	0	0,477
beku	0	0	0	1	1	3	0,477	0	0	0	0,477
darah	0	0	0	1	1	3	0,477	0	0	0	0,477
hdl	0	0	0	1	1	3	0,477	0	0	0	0,477
sakit	0	0	0	3	1	3	0,477	0	0	0	1,431
jantung	0	0	0	2	1	3	0,477	0	0	0	0,954
diabetes	0	0	0	1	1	3	0,477	0	0	0	0,477

wanita	0	0	0	1	1	3	0,477	0	0	0	0,477
cegah	0	0	0	1	1	3	0,477	0	0	0	0,477
gejala	0	0	0	2	1	3	0,477	0	0	0	0,954
nyaman	0	0	0	1	1	3	0,477	0	0	0	0,477
menopause	0	0	0	1	1	3	0,477	0	0	0	0,477
imbang	0	0	0	1	1	3	0,477	0	0	0	0,477
estrogen	0	0	0	1	1	3	0,477	0	0	0	0,477
kalsium	0	0	0	1	1	3	0,477	0	0	0	0,477
efektif	0	0	0	1	1	3	0,477	0	0	0	0,477
rheumatoid	0	0	0	1	1	3	0,477	0	0	0	0,477
arthritis	0	0	0	1	1	3	0,477	0	0	0	0,477
tubuh	0	0	0	2	1	3	0,477	0	0	0	0,954
serang	0	0	0	1	1	3	0,477	0	0	0	0,477
kebal	0	0	0	1	1	3	0,477	0	0	0	0,477
radang	0	0	0	1	1	3	0,477	0	0	0	0,477
cukup	0	0	0	1	1	3	0,477	0	0	0	0,477
sendi	0	0	0	1	1	3	0,477	0	0	0	0,477
beli	0	0	0	1	1	3	0,477	0	0	0	0,477
toko	0	0	0	1	1	3	0,477	0	0	0	0,477
sedia	0	0	0	1	1	3	0,477	0	0	0	0,477
ukur	0	0	0	1	1	3	0,477	0	0	0	0,477
potong	0	0	0	1	1	3	0,477	0	0	0	0,477
salad	0	0	0	1	1	3	0,477	0	0	0	0,477
nsa	0	0	0	1	1	3	0,477	0	0	0	0,477

Dari hasil perhitungan bobot tersebut maka dapat dihitung nilai similaritas masing –masing dokumen uji terhadap dokumen latih dengan menggunakan persamaan 2.2

Yang pertama dihitung terlebih dulu akar dari jumlah bobot kk kuadrat begitu juga D1, D2, D3 :

$$sqrt\ (kk) = sqrt \sum_{j=1}^{n} kk_j^2$$
$$= \sqrt{30{,}968}$$
$$= 5{,}565$$

$$sqrt\ (D1) = sqrt \sum_{j=1}^{n} D1_j^2$$
$$= \sqrt{32{,}112}$$
$$= 5{,}667$$

$$sqrt\ (D2) = sqrt \sum_{j=1}^{n} D2_j^2$$
$$= \sqrt{41{,}714}$$
$$= 6{,}459$$

$$sqrt\ (D3) = sqrt \sum_{j=1}^{n} D3_j^2$$
$$= \sqrt{55{,}905}$$
$$= 7{,}477$$

kemudian menghitung jumlah kk dot D1, D2, D3:

$$sum\ (kk\ dot\ D1) = \sum_{j=1}^{n} kk_j D_{1,j}$$
$$= 4{,}024$$

$$sum\ (kk\ dot\ D2) = \sum_{j=1}^{n} kk_j D_{2,j}$$
$$= 3{,}764$$

$$sum\ (kk\ dot\ D3) = \sum_{j=1}^{n} kk_j D_{3,j}$$
$$= 9{,}395$$

Similaritasnya:

$$cosine\ (D1) = sum\ (kk\ dot\ D1)/(sqrt\ (kk) * sqrt(D1))$$
$$= 4{,}024\ /\ (5{,}565\ *\ 5{,}667)$$
$$= 4{,}024/\ 31{,}537$$
$$= 0{,}128$$

$$cosine\ (D2) = sum\ (kk\ dot\ D2)/(sqrt\ (kk) * sqrt(D2))$$
$$= 3{,}764/\ (5{,}565 * 6{,}459)$$
$$= 3{,}764/\ 35{,}944$$
$$= 0{,}105$$

$$cosine\ (D3) = sum\ (kk\ dot\ D3)/(sqrt\ (kk) * sqrt(D3))$$
$$= 9{,}395/\ (5{,}565 * 7{,}477)$$
$$= 9{,}395/\ 41{,}609$$
$$= 0{,}226$$

Semua perhitungan ini adalah perhitungan berdasar dokumen latih satu diperlakukan juga untuk dokumen latih yang kedua, sehingga akan di peroleh matriks similaritas seperti ini:

Tabel 3.2 Matriks similaritas

	D1	D2	D3
sim1	0,129904456	0,105901684	0,228337242
sim2	0,025035318	0,131037896	0,374484435

Dari hasil matrix di atas maka akan dilakukan *clustering* dengan metode *single pass clustering* dengan menggunaka *threshold* dimulai dari 0 sampai mendapatkan jumlah *cluster* sebanyak jumlah dokumen uji. Sehingga akan didapatkan nilai *threshold* dengan hasil *precision* dan *recall* yang terbaik.

Tabel 3.3 *precision* dan *recall*

percobaan	threshold	correct	wrong	miss	precision	recall	jumlah *cluster*
1	0,005	2	1	1	0,667	0,667	1
2	0,01	2	1	1	0,667	0,667	1
3	0,015	2	1	1	0,667	0,667	1
4	0,02	2	1	1	0,667	0,667	2
5	0,025	2	1	1	0,667	0,667	2
6	0,03	2	1	1	0,667	0,667	2
7	0,035	2	1	1	0,667	0,667	2
8	0,04	2	1	1	0,667	0,667	2
9	0,045	2	1	1	0,667	0,667	2

10	0,05	2	1	1	0,667	0,667	2
11	0,055	2	1	1	0,667	0,667	2
12	0,06	2	1	1	0,667	0,667	2
13	0,065	2	1	1	0,667	0,667	2
14	0,07	2	1	1	0,667	0,667	2
15	0,075	2	1	1	0,667	0,667	3

Keterangan :
Correct : hasil dari sistem yang sesuai dengan data asli
Wrong : hasil dari sistem yang tidak sesuai dengan data asli
Miss : data asli yang tidak terdapat dalam hasil sistem

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

4.1. Lingkungan implementasi

Lingkungan implementasi yang akan dijelaskan dalam sub-bab ini meliputi lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1. Lingkungan implementasi perangkat keras

Perangkat keras yang digunakan dalam pembuatan sistem pemilahan arikel berita dalam skripsi ini meliputi:

1. Processor Genuine Intel(R) CPU T2300 @ 1.66 gHz (2 CPUs)
2. Memory RAM 1016 MB
3. Harddisk 148,9 GB

4.1.2. Lingkungan implementasi perangkat lunak

Perangkat lunak yang digunakan dalam pembuatan sistem pemilahan artikel berita dalam skripsi ini meliputi:

1. Sistem operasi Microsoft Windows XP Professional
2. Borland Delphi 7
3. Text editor notepad

4.2. Implementasi program

Berdasarkan analisa dan perancangan yang dijabarkan pada bab 3, maka pada sub-bab ini akan dijelaskan implementasi dari perancangan proses yang telah dijabarkan pada bab 3.

4.2.1. Implementasi *preprocessing*

Dalam implementasi *preprocessing* ini terdapat beberapa proses meliputi *case folding*, *tokenizing*, *filtering* dan *stemming*. *Preprocessing* ini dilakukan pada dokumen latih dan dokumen uji. Dalam implementasi ini tahap-tahap *preprocessing case folding*, *tokenizing*, *filtering* berada dalam satu prosedur, yaitu prosedur `Check_Document`.

4.2.1.1. *Case folding*

Case folding merupakan tahap awal dalam *preprocessing*. Proses *case folding* bertujuan untuk merubah semua karakter huruf

menjadi huruf kecil dan karakter selain huruf dan angka diubah menjadi karakter spasi. Proses *case folding* ditunjukkan pada kode program 4.1.

```
Tmp_Doc := Test_Document.Text;
  for i := 1 to Length(Tmp_Doc) do
  begin
    if ( ord(Tmp_Doc[i]) in [0..47]) or ( ord(Tmp_Doc[i])
in [58..64]) or ( ord(Tmp_Doc[i]) in [91..96]) or (
ord(Tmp_Doc[i]) in [123..127]) then
    begin
      Tmp_Doc[i] := ' ';
    end;
  end;
  Test_Document.Text := Tmp_Doc;
 Lowercase(Test_Document.Text);
```

Kode program 4.1. kode program *case folding*

4.2.1.2. *Tokenizing*

Tokenizing merupakan proses memecah kata per kata dari dokumen teks atau dapat disebut dengan *parsing.* Tahap pemecahan teks menjadi potongan kata per kata yang ditunjukkan pada kode program 4.2.

```
      Dump_Document := TStringList.Create;
        while Length(Test_Document.Text)>0 do
        begin

            if pos(' ',Test_Document.Text) <> 0 then
            begin
              Tmp_Word  :=  Copy(Test_Document.Text,1,pos('
      ',Test_Document.Text));
            end else
            begin
              Tmp_Word := Test_Document.Text;
              Test_Document.Text := '';
            end;
            Tmp_Word := Trim(Tmp_Word);

            Tmp_Doc := Test_Document.Text;
            Delete(Tmp_Doc,1,pos(' ',Test_Document.Text));
            Test_Document.Text := Tmp_Doc;

            if (Length(Tmp_Word)>0) then
            begin
              Dump_Document.Add(Tmp_Word);
```

```
        end;
    end;
    Test_Document.Text := Dump_Document.Text;
```

Kode program 4.2. kode program *tokenizing*

4.2.1.3. *Filtering*

Filtering merupakan proses penghilangan kata-kata yang tidak penting yang dicocokkan dengan daftar kata-kata tidak penting atau *stopword* berdasarkan jurnal F. Tala. Proses *filtering* ini ditunjukkan pada kode program 4.3.

```
Stop_Words := TStringList.Create;
  Stop_Words.LoadFromFile('D:\Verly\Stop
Words.txt');
  for i := 1 to Test_Document.Count do
  begin
    Tmp_Word := Test_Document.Strings[i-1];
    for j := 1 to Stop_Words.Count do
    begin
      if Tmp_Word = Stop_Words.Strings[j-1] then
      begin
          Test_Document.Strings[i-1] := 'xxxxxxxx';
      end;
    end;
  end;

Dump_Document.Text := '';
  for i := 1 to Test_Document.Count do
  begin
    if Test_Document.Strings[i-1] <> 'xxxxxxxx' then
    begin
      Dump_Document.Add(Test_Document.Strings[i-1]);
    end;
  end;
 Test_Document.Text  := Dump_Document.Text;
```

Kode program 4.3. kode program *filtering*

4.2.1.4. *Stemming*

Stemming merupakan proses pemotongan imbuhan kata untuk memperoleh kata dasar dari sebuah kata berimbuhan. Kata dasar yang digunakan sebagai acuan diambil dari KBBI. Dalam proses ini terdapat tahap pencocokan dengan kamus kata dasar yang dalam implementasi

ini ada dalam fungsi `Check_Dictionary` yang ditunjukkan dalam kode program 4.4.

```
function  TForm1.Check_Dictionary(Tmp_Word:  string;
Dictionary: TStringList):boolean;
var Exist : Boolean;
    j     : integer;
begin

    Exist := False;
    for j := 1 to Dictionary.Count do
    begin
      if (Tmp_Word = Dictionary.Strings[j-1]) then
      begin
        Exist := True;
        break;
      end;
    end;

    result := Exist;
  end;
```

Kode program 4.4. kode program fungsi pengecekan kamus

Proses *stemming* yang diimplementasikan dalam skripsi ini terdapat dua jenis *stemming*, yaitu *stemming* berdasarkan penelitian Arifin dan Setiono yang berada pada prosedur `stemming_a`. serta *stemming* berdasarkan penelitian Nazief dan Adriani yang berada pada prosedur `stemming_NP`.

4.2.2. Implementasi pembobotan

Pembobotan yang diimplementasikan dalam skripsi ini yaitu pembobotan dengan menggunakan TF-IDF. Proses pembobotan ini dilakukan terhadap dokumen latih dan dokumen uji. Proses pembobotan diawali dengan menghitung TF(*Term Frequency*), yaitu menghitung jumlah kemunculan kata dalam sebuah dokumen yang dalam implementasi ini ada dalam prosedur `freq` yang ditunjukkan dalam kode program 4.5.

```
procedure TForm1.freq(NamaFile: string;ResultColumn:integer
);
var
tmp_word:string;
i,index, index2, Tmp_I:integer;
```

```
Exist:boolean;
Tmp_V:Variant;
begin

  Index           := 0;
  for i := 2 to tbPerhitungan.RowCount do
  begin
    try
      Index2 := StrToInt(tbPerhitungan.Cells[0,i-1].value);
      Index := Index2;
    except
    end;
  end;


  for i := 1 to Test_Document.Count do
  begin
    Tmp_Word  := Test_Document.Strings[i-1];
    Exist     := false;
    Index2    := 0; // IndexScanning

    repeat
        Inc(Index2);
        try
          Tmp_V := tbPerhitungan.Cells[1,Index2].value;
        except
          Tmp_V := '';
        end;
        if Tmp_Word = Tmp_V then
        begin
          Exist := true;
          break;
        end;
    until   (Tmp_V   =   '')   or   (Exist)   or   (Index2   =
tbPerhitungan.RowCount-1);

    if not Exist then
    begin
      tbPerhitungan.RowCount := tbPerhitungan.RowCount + 1;
      Inc(Index);
      tbPerhitungan.Cells[0,Index].Value := Index;
      tbPerhitungan.Cells[1,Index].Value := Tmp_Word;
      tbPerhitungan.Cells[ResultColumn,Index].Value := 1;
    end else
    begin
      try
        Tmp_I                                              :=
```

```
tbPerhitungan.Cells[ResultColumn,Index2].Value;
      except
        Tmp_I := 0;
      end;
      tbPerhitungan.Cells[ResultColumn,Index2].Value      :=
Tmp_I + 1;
    end;
  end;

end;
```

Kode program 4.5. kode program menghitung frekuensi

Kemudian akan dihitung juga jumlah total dokumen uji yang ditunjukkan pada kode program 4.6.

```
      v_df := 0;
          for j := 1 to ListUji.Count do
          begin
            if     tbPerhitungan.Cells[ListUji.Count+3-j,i-
      1].Value > 0 then
            begin
              inc(v_df);
            end;
          end;
    tbPerhitungan.Cells[ListUji.Count+3,i-1].Value := v_df;
```

Kode program 4.6. kode program jumlah dokumen uji

Setelah itu akan dilakukan penghitungan D/DF(*Document Frequency*) dimana DF adalah jumlah dokumen yang memiliki kata yang dimaksud yang ditunjukkan pada kode program 4.7.

```
Try
      tbPerhitungan.Cells[ListUji.Count+4,i-1].Value      :=
ListUji.count / v_df;
      tbPerhitungan.Cells[ListUji.Count+4,i-1].Format     :=
'0.0#';
    except
      tbPerhitungan.Cells[ListUji.Count+4,i-1].Value := 0;
    end;
```

Kode program 4.7. kode program D/DF

Serta akan dihitung juga IDF yang merupakan hasil log dari D/DF yang ditunjukkan pada kode program 4.8.

```
Try
      tbPerhitungan.Cells[ListUji.Count+5,i-1].Value      :=
Log10(tbPerhitungan.Cells[ListUji.Count+4,i-1].Value);
    except
      tbPerhitungan.Cells[ListUji.Count+5,i-1].Value := 0;
    end;

    tbPerhitungan.Cells[ListUji.Count+5,i-1].Format       :=
'0.0##';
```

Kode program 4.8. kode program menghitung IDF

Sehingga dapat dihitung nilai bobot masing-masing dokumen seperti yang ditunjukkan pada kode program 4.9.

```
for j := 1 to ListUji.Count+1 do
    begin
      // w kk dan w D1..Dn
      tbPerhitungan.Cells[ListUji.Count+5+j,i-1].Value    :=
(tbPerhitungan.Cells[ListUji.Count+5,i-1].Value)           *
(tbPerhitungan.Cells[1+j,i-1].Value);
      tbPerhitungan.Cells[ListUji.Count+5+j,i-1].Format   :=
'0.0##';
```

Kode program 4.9. kode program menghitung bobot

4.2.3. Implementasi similaritas

Similaritas merupakan proses untuk memperoleh nilai similaritas atau kesamaan atau kemiripan antar dokumen uji dengan dokumen latih yang dalam implementasi ini akan menggundakan metode VSM(*Vector Space Model*). Proses perhitungan nilai similaritas ini diawali dengan perhitungan kuadrat dari nilai bobot yang ditunjukkan pada kode program 4.10., sehingga akan dapat dihitung hasil akar dari jumlah nilai kuadrat tersebut yang ditunjukkan pada kode program 4.11.

```
tbPerhitungan.Cells[ListUji.Count+1+ListUji.Count+5+j,i-
1].Value  :=  power(tbPerhitungan.Cells[ListUji.Count+5+j,i-
1].Value,2);

tbPerhitungan.Cells[ListUji.Count+1+ListUji.Count+5+j,i-
1].Format := '0.0##';
```

Kode program 4.10. kode program menghitung kuadrat dari nilai bobot

```
for j :=  1 to ListUji.Count+1 do
  begin
    Sum_Total := 0;
    for i := 2 to tbPerhitungan.RowCount do
    begin
      Sum_Total          :=          Sum_Total          +
tbPerhitungan.Cells[ListUji.Count*2+6+j,i-1].Value;
    end;

tbPerhitungan.Cells[ListUji.Count*2+6+j,tbPerhitungan.rowco
unt-1 ].Value := sqrt(Sum_Total);

tbPerhitungan.Cells[ListUji.Count*2+6+j,tbPerhitungan.rowco
unt-1 ].Format := '0.0##';
  end;
```

Kode program 4.11. kode program menghitung akar dari hasil kuadrat

Kemudian dihitung juga hasil perkalian nilai bobot latih dan nilai bobot uji yang ditunjukkan pada kode program 4.12., dan untuk kemudian akan dihitung jumlah dari hasil perkalian tersebut untuk masing-masing dokumen uji yang ditunjukkan pada kode program 4.13.

```
for j := 1 to ListUji.Count do
    begin
      tbPerhitungan.Cells[ListUji.Count*3+7+j,i-1].Value :=
tbPerhitungan.Cells[ListUji.Count+6,i-1].Value           *
tbPerhitungan.Cells[ListUji.Count+6+j,i-1].Value;
      tbPerhitungan.Cells[ListUji.Count*3+7+j,i-1].Format
:= '0.0##';
    end;
  end;
```

Kode program 4.12. kode program menghitung perkalian bobot latih dan tiap uji

```
for j :=  1 to ListUji.Count do
  begin
    Sum_Total := 0;
    for i := 2 to tbPerhitungan.RowCount do
    begin
      Sum_Total          :=          Sum_Total          +
tbPerhitungan.Cells[ListUji.Count*3+7+j,i-1].Value;
    end;

tbPerhitungan.Cells[ListUji.Count*3+7+j,tbPerhitungan.rowco
unt-1 ].Value := Sum_Total;
```

```
tbPerhitungan.Cells[ListUji.Count*3+7+j,tbPerhitungan.rowco
unt-1 ].Format := '0.0##';
  end;
```

Kode program 4.13. kode program menghitung akar hasil perkalian

sehingga dari perhitungan-perhitungan tersebut dapat dihitung nilai similaritas yang ditunjukkan pada kode program 4.14.

```
for i := 1 to ListUji.Count do
  begin
    tbPerhitungan.Cells[1,j+i-1].Value := 'D'+inttostr(i);
    tbPerhitungan.Cells[2,j+i-1].Value                    :=
tbPerhitungan.Cells[tbPerhitungan.ColCount-ListUji.Count+i-
1,s].Value / ( tbPerhitungan.Cells[tbPerhitungan.ColCount-
1-ListUji.Count*2,s].Value                                *
tbPerhitungan.Cells[tbPerhitungan.ColCount-ListUji.Count*2-
1+i,s].Value);
    tbPerhitungan.Cells[2,j+i-1].Format := '0.0##';
  end;
```

Kode program 4.14. kode program menghitung similaritas

4.2.4. Implementasi *single pass clustering*

untuk tahap *single pass* dibuat *class object* baru yaitu *class* `cluster` yang di dalamnya terdapat prosedur `Add_Dokumen` untuk menghitung nilai cluster yang merupakan nilai similaritas dokumen yang termasuk di dalam cluster tersebut yang ditunjukkan pada kode program 4.15. dan *class* `List_Cluster` yang di dalamnya terdapat prosedur `Compare_Cluster` untuk melakukan perhitungan perbandingan atau similaritas antara nilai isi *cluster* dan nilai similaritas dokumen yang akan di *cluster* dimana hasil perhitungan tersebut akan dibandingkan dengan *threshold* sehingga dapat diputuskan bahwa dokumen tersebut masuk ke *cluster* pembanding tadi atau *cluster* baru, yang ditunjukkan pada kode program 4.16.

```
procedure                    TCluster.Add_Dokumen(No_Dokumen:
integer;Data_Dokumen : array of real);
var tmp : real;
  i,j : integer;
begin
  Inc(Jumlah_Dokumen);
  List_No_Dokumen[Jumlah_Dokumen] := No_Dokumen;

  for i := 1 to Jumlah_Kategori do
```

```
  begin
    List_Nilai_Dokumen[Jumlah_Dokumen,i] := Data_Dokumen[i-
1];
  end;

  // Menghitung nilai cluster
  for i := 1 to Jumlah_Kategori do
  begin
    Tmp := 0;
    for j := 1 to Jumlah_Dokumen do
    begin
      Tmp := Tmp + List_Nilai_Dokumen[j,i];
    end;
    Nilai_Cluster[i] := Tmp / Jumlah_Dokumen;
  end;
end;
```

Kode program 4.15. prosedur Add_Dokumen

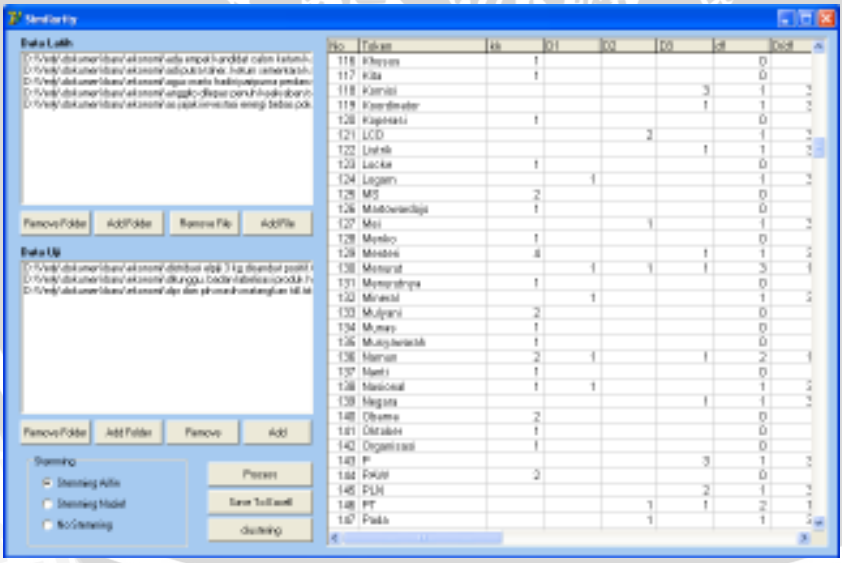
```
function  TList_Cluster.Compare_Cluster(No_Cluster :
integer; Data_Dokumen : array of real;Threshold :
real):real;
var Total : real;
i : integer;
begin
  if No_CLuster <= Jumlah_Cluster then
  begin
    Total := 0;

    for i := 1 to Jumlah_Kategori do
    begin
      Total := Total +
(Clusters[No_Cluster].Nilai_Cluster[i] * Data_Dokumen[i-
1]);
    end;


    if Total >= Threshold then
    begin
      result := Total;
    end else
    begin
      result := -1;
    end;

  end else result := -1;
end;
```

Kode program 4.16. prosedur Compare_Cluster

4.3. Implementasi *interface*

4.3.1. *Interface similarity*

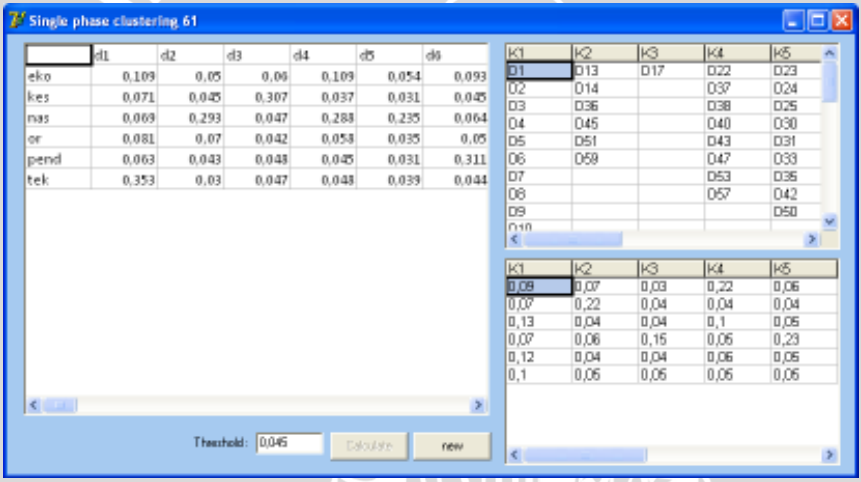
Interface similarity ditunjukkan pada gambar 4.1., *interface* ini merupakan tampilan awal saat aplikasi dijalankan. Dalam interface ini pengguna memasukkan dokumen-dokumen latih yang berektensi .txt, dokumen dapat dimasukkan langsung satu *folder* dengan menekan tombol *add folder* ataupun memasukkan dokumen satu persatu dengan menekan tombol *add file*. Kemudian pengguna juga memasukkan dokumen-dokumen uji yang juga berektensi .txt dengan cara yang sama dengan cara memasukkan dokumen latih. Kemudian pengguna memilih apakah akan menggunakan *stemming* Arifin atau *stemming* Nazief atau tanpa menggunakan *stemming*. Kemudian klik tombol *process* untuk memproses sehingga didapat nilai similaritas tiap dokumen uji terhadap latih yang ditampilkan pada tabel. Pada tabel tersebut ditampilkan mulai dari daftar kata, frekuensi kata dan tahap-tahap perhitungan pembobotan hingga similaritas. Untuk menyimpan tabel keseluruhan ke dalam format excel dilakukan dengan menekan tombol *save to excel*. Untuk menuju proses *clustering* maka dilakukan dengan menekan tombol *clustering* untuk menuju *interface single pass clustering*.



Gambar 4.1. *interface similarity*

4.3.2. *Interface single pass clustering*

Interface single pass clustering ditunjukkan pada gambar 4.2. *interface* akan muncul ketika tombol *clustering* pada *interface* awal telah ditekan. Pada *interface* ini pengguna memasukkan terlebih dulu nilai threshold dan kemudian menekan tombol *calculate,* kemudian akan muncul dialog untuk memilih file excel yang berisi kumpulan nilai similaritas dokumen uji terhadap semua latih per kategori. Setelah file dipilih maka akan muncul pada tabel sebelah kiri adalah tabel masukan dari file excel, dan pada tabel sebelah kanan atas akan muncul hasil *clustering* dokumen uji, sedangkan tabel kanan bawah merupakan tabel nilai *cluster*. Tombol *new* untuk melakukan *clustering* dokumen kembali.



Gambar 4.2. *interface single pass clustering*

4.4. Pembahasan dan analisa hasil percobaan sistem

4.4.1. Analisa uji *stemming*

Pada uji *stemming* ini, dilakukan pengujian terhadap sejumlah kata berimbuhan. Kemudian kata berimbuhan tersebut akan melalui proses *stemming* Arifin dan *stemming* Nazief yang telah diterapkan dalam skripsi ini.

Hasil pengujian tersebut ditunjukkan pada lampiran 2, dimana daftar kata-kata yang telah diberi kombinasi awalan dan akhiran tersebut

ditunjukkan pada kolom kata berimbuhan (kolom 3). Hasil kata dasar setelah diproses dengan *stemming* Arifin ditunjukkan pada kolom Arifin (kolom 4). Sedangkan hasil kata dasar setelah melalui proses *stemming* Nazief ditunjukkan pada kolom Nazief (kolom 5).

Pada tabel hasil proses *stemming* Arifin dan *stemming* Nazief tersebut dapat dilihat bahwa pada imbuhan awalan ”di-” dengan akhiran “-kan” dan “-kan,-nya” kata “diberikan” dan “diberikannya” pada *stemming* Arifin gagal diproses dimana kata dasar menjadi kata “ikan” karena setelah dilakukan pemotongan awalan “di-“, kata “ber” dianggap sebagai awalan sehinggak kata “ber” terpotong dan kata “ikan” terdapat dalam kamus maka kata tersebut yang dikeluarkan menjadi kata dasar. Sedangkan pada *stemming Nazief* berhasil diproses karena memotong akhiran terlebih dulu dan setelah pemotongan akhiran, kata “beri” terdapat dalam kamus. Selain itu, untuk kata “dicarikan”, “dicarikannya”, “disamakan” dan kata “disamakannya” pada *stemming* Arifin menjadi kata “carik” dan “samak”, hal ini disebabkan karena setelah dipotong awalan “di-” dan akhiran “-an” kata tersebut terdapat dalam kamus, sedangkan dengan *stemming* Nazief berhasil karena dalam algoritma Nazief terdapat proses pengecekan jika akhiran “-an” dan huruf sebelumnya adalah “k” maka akan dianggap imbuhan akhiran “-kan” dan dilakukan pemotongan akhiran “-kan”.

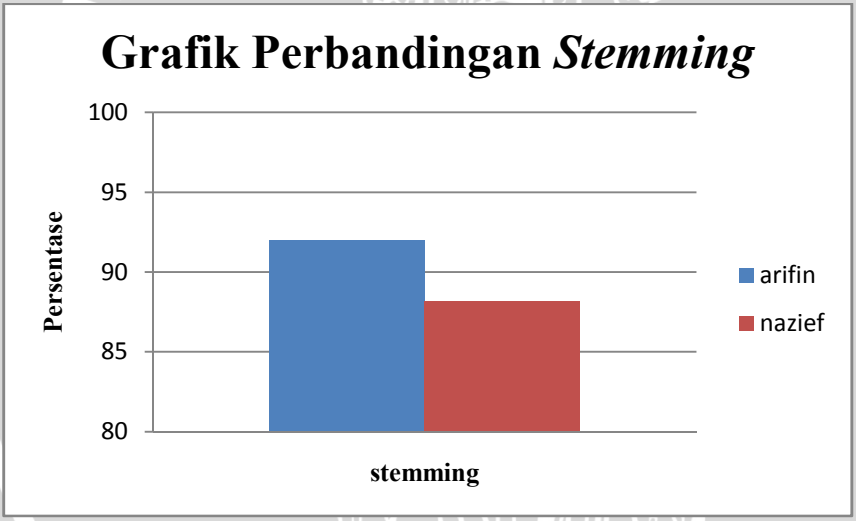
Pada imbuhan awalan “ke-” dengan akhiran “-an” dan “-an,-nya” serta akhiran “-an,-nya,-lah”, *stemming* Nazief gagal mengolah kata “kenaikan”, “kenaikannya” dan “kenaikannyalah” karena setelah tahap pemotongan akhiran “-an” terdapat huruf “k” sehingga dianggap imbuhan akhiran “-kan” sehingga kata menjadi “kenai” dimana kata tersebut tidak terdapat di kamus, sehingga kata yang dikeluarkan adalah kata awal yaitu “kenaikan”. Sedangkan pada *stemming* Arifin, kata tersebut berhasil diproses menjadi “naik” karena algoritma arifin memotong imbuhan awalan terlebih dulu yaitu “ke-” dan kemudian memotong akhiran “-lah” dan atau “-nya” setelah itu memotng akhiran “-an”.

Selain itu, pada uji *stemming* ini juga dilakukan pengujian terhadap 1000 kata berimbuhan yang terdapat pada dokumen uji yang digunakan dalam skripsi ini, guna mengetahui perbandingan tingkat keakuratan *stemming* Arifin dan *stemming* Nazief. Hasil perbandingan ditunjukkan pada tabel 4.1.

Tabel 4.1. tabel perbandingan uji *stemming*

stemming	kata yang benar	kata yang salah	persentase
Arifin	920	80	92
Nazief	882	118	88,2

Tabel 4.1. merupakan tabel hasil uji *stemming* dari seribu kata berimbuhan yang terdapat pada dokumen uji. Pada tabel tersebut dapat dilihat bahwa kata berimbuhan yang berhasil diproses dan benar dengan menggunakan *stemming* Arifin sebanyak 920 kata, sedangkan dengan menggunakan *stemming* Nazief sebanyak 882 kata. *Stemming* Arifin lebih banyak dapat memproses kata berimbuhan dibanding *stemming* Nazief.



Gambar 4.3. grafik perbandingan *stemming*

Grafik perbandingan *stemming* yang ditunjukkan pada gambar 4.3. merupakan hasil setelah dilakukan uji coba *stemming* dengan menggunakan seribu kata berimbuhan yang diambil dari dokumen uji, didapatkan hasil bahwa dengan menggunakan *stemming* Nazief lebih baik dibanding dengan menggunakan *stemming* Arifin walaupun

persentasenya tidak terlalu jauh perbedaannya. Dimana dengan menggunakan *stemming* Arifin diperoleh hasil 92% sedangkan dengan menggunakan *stemming* Nazief diperoleh hasil 898,2%. Hal ini dikarenakan dari seribu kata berimbuhan yang diujikan, jumlah kata berimbuhan yang dapat diproses dengan *stemming* Arifin lebih banyak dibanding dengan kata berimbuhan yang dapat diproses oleh *stemming* Nazief.

4.4.2. Analisa *precision* dan *recall*

Dalam percobaan ini ada dua jenis dokumen sebagai masukan, yaitu dokumen latih dan uji yang masing-masing mempunyai enam jenis kategori berita, yaitu ekonomi, kesehatan, nasional, olah raga, pendidikan dan teknologi. Perincian jumlah dokumen latih dan uji berdasarkan jenis kategorinya ditunjukkan pada tabel 4.2. Dokumen uji telah diketahui terlebih dahulu jenis kategorinya bertujuan untuk pengecekan keakuratan hasil *cluster* yang dilakukan oleh sistem. Dalam percobaan ini dokumen uji akan diacak sebelum dimasukkan ke dalam sistem.

Tabel 4.2. tabel dokumen latih dan dokumen uji

	dokumen latih	**dokumen uji**
ekonomi	60	10
kesehatan	56	11
nasional	60	10
olahraga	59	11
pendidikan	57	10
teknologi	66	9
jumlah dokumen	358	61

Hasil similaritas dokumen uji terhadap dokumen latih dengan menggunakan *stemming* Arifin ditunjukkan pada lampiran 3, sedangkan tabel hasil similaritas dokumen uji terhadap dokumen latih dengan menggunakan *stemming* Nazief ditunjukkan pada lampiran 4, dan merupakan tabel hasil similaritas dokumen uji terhadap dokumen latih tanpa menggunakan *stemming* ditunjukkan pada lampiran 5. Dimana C1 merupakan *cluster* pertama yaitu kategori ekonomi, C2 merupakan

cluster kedua yaitu kategori kesehatan, C3 merupakan *cluster* ketiga yaitu kategori nasional, C4 merupakan *cluster* keempat yaitu kategori olah raga, C5 merupakan *cluster* kelima yaitu kategori pendidikan, dan C6 merupakan *cluster* keenam yaitu kategori teknologi.

Dari tabel pada lampiran 3, lampiran 4 dan lampiran 5, dapat dilihat bahwa nilai similaritas dokumen uji terhadap dokumen latih rata-rata mengalami kenaikan ketika menggunakan *stemming*, baik itu menggunakan *stemming* Arifin ataupun menggunakan *stemming* Nazief.

Dari nilai similaritas pada tabel-tabel tersebut nantinya akan digunakan untuk mengevaluasi pengujian sistem. Parameter evaluasi yang digunakan pada sistem ini yaitu *precision* dan *recall* yang menilai tingkat keberhasilan sistem dalam meng-*cluster* dokumen uji ke dalam kategori yang sesuai. Hasil *precision* dan *recall* yang diperoleh ditunjukkan pada Tabel 4.4.

Pada tabel 4.4. dapat dilihat bahwa dengan menggunakan *stemming* Arifin nilai *precision* dan *recall* terbaik berada pada *threshold* 0,04 yaitu 0,360656 (36 %), sedangakan dengan menggunakan *stemming* Arifin nilai *precision* dan *recall* terbaik berada pada *threshold* 0,045 yaitu 0,344262 (34 %) dan jika tanpa menggunakan *stemming*, nilai *precision* dan *recall* terbaik berada pada *threshold* 0,03 yaitu 0,163934 (16 %). Dari perolehan tersebut dapat diketahui bahwa dengan menggunakan *stemming* dapat meningkatkan *precision* dan *recall clustering* dokumen dengan menggunakan *single pass*.

Hal tersebut disebabkan karena pada percobaan tanpa *stemming*, sebenarnya banyak kata yang mempunyai kata dasar yang sama tetapi memiliki partikel atau imbuhan kata yang berbeda. Sedangkan dengan menggunakan *stemming* partikel atau imbuhan tersebut dihilangkan sehingga menghasilkan banyak kata yang sama dimana hal tersebut akan meningkatkan frekuensi kata. Dimana frekuensi kata berpengaruh similaritas, jika frekuensi kata meningkat maka nilai similaritas akan meningkat juga. Dimana nilai similaritas berpengaruh pada hasil perhitungan *precision* dan *recall*, jika nilai similaritas dokumen uji terhadap dokumen latih meningkat, maka nilai *precision* dan *recall* juga akan meningkat.

Tabel 4.4. tabel hasil evaluasi sistem

percobaan	threshold	stemming Arifin			stemming Nazief			tanpa stemming		
		precision	recall	jumlah cluster	precision	recall	jumlah cluster	precision	recall	jumlah cluster
1	0	0,163934	0,163934	1	0,163934	0,163934	1	0,163934	0,163934	1
2	0,005	0,163934	0,163934	1	0,163934	0,163934	1	0,163934	0,163934	1
3	0,01	0,163934	0,163934	1	0,163934	0,163934	1	0,163934	0,163934	1
4	0,015	0,163934	0,163934	1	0,163934	0,163934	1	0,163934	0,163934	1
5	0,02	0,163934	0,163934	1	0,163934	0,163934	1	0,163934	0,163934	1
6	0,025	0,163934	0,163934	1	0,163934	0,163934	1	0,163934	0,163934	4
7	0,03	0,163934	0,163934	6	0,196721	0,196721	2	0,163934	0,163934	5
8	0,035	0,163934	0,163934	6	0,163934	0,163934	7	0,016393	0,016393	8
9	0,04	0,360656	0,360656	6	0,163934	0,163934	7	0,081967	0,081967	11
10	0,045	0,163934	0,163934	10	0,344262	0,344262	7	0	0	14
11	0,05	0	0	13	0,098361	0,098361	10	0	0	14
12	0,055	0,04918	0,04918	14	0,081967	0,081967	12	0	0	19
13	0,06	0	0	18	0,016393	0,016393	13	0	0	23
14	0,065	0	0	19	0,131148	0,131148	17	0	0	27
15	0,07	0	0	22	0,131148	0,131148	20	0,131148	0,131148	31
16	0,075	0	0	27	0,114754	0,114754	24	0,131148	0,131148	34
17	0,08	0,131148	0,131148	31	0,147541	0,147541	26	0,131148	0,131148	36
18	0,085	0,131148	0,131148	33	0,147541	0,147541	28	0,131148	0,131148	38
19	0,09	0,131148	0,131148	35	0,147541	0,147541	30	0	0	43
20	0,095	0,131148	0,131148	40	0,147541	0,147541	33	0	0	45
21	0,1	0	0	43	0,147541	0,147541	36	0	0	47
22	0,105	0	0	44	0,147541	0,147541	38	0	0	47


UNIVERSITAS BRAWIJAYA

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari percobaan-percobaan yang telah dilakukan dapat disimpulkan bahwa:

1. Penggunaan *stemming* pada tahap *preprocessing* dalam meng-*cluster* dokumen dapat meningkatkan *precision* dan *recall clustering* dokumen.
2. Untuk beberapa kasus dalam Bahasa Indonesia, terutama dalam skripsi ini, algoritma *stemming* Arifin lebih baik dari algoritma *stemming* Nazief.
3. Dalam skripsi ini, tingkat keberhasilan *stemming* mengolah kata berimbuhan menjadi kata dasar yaitu 92% dengan menggunakan *stemming* Arifin dan 88,2% dengan menggunakan *stemming* Nazief.

5.2. Saran

Untuk penelitian lebih lanjut, disarankan melakukan pengujian dengan menambah algoritma *stemming* yang lain agar tampak pengaruh *stemming* terhadap *clustering* dokumen. Kemudian, untuk meningkatkan hasil *stemming* dapat dilakukan pemilihan kata dasar dalam kamus seperti hanya mendaftar kata dasar yang dapat diberi imbuhan saja.


UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Arifin, Agus Zainal., Setiono, Ari Novan. 2002. *Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering*. Teknik Elektro, Institut Teknologi Sepuluh Nopember: Surabaya.

Asian, Jelita., Williams, Hugh. E., & Tahaghoghi, S.M.M. 2005. Proceedings of the Twenty-eighth Australasian conference on Computer Science - Volume 38: *stemming Indonesian*. Australian Computer Society, Inc: Australia.

Capon, Trevor Bench., Soda, Giovanni., & Tjoa, A Min. 1999. *Database and Expert Sistems Applications*: 10th International Conference, DEXA'99. Springer: NewYork.

Departemen pendidikan nasional. 2001. *Kamus Besar Bahasa Indonesia*. Balai pustaka: Jakarta.

Departemen pendidikan nasional. 2001. *Kamus Besar Bahasa Indonesia Daring*. http://pusatbahasa.depdiknas.go.id/kbbi/index.php

Feldman, Ronen., Sanger, James. 2007. *The Text Mining Handbook*. Cambridge University Press: England.

Hovy, E. (2003). Text *Summarization*. Dalam R. Mitkov, *The Oxford Handbook of Computational Linguistics* (hal. 583-589). Oxford: Oxford University Press.

Jain, A. K., Murty, M. N., & Flynn, P. J. 1999. *Data Clustering* : A review in ACM computing surveys, vol.32 no.3

Manning, Christoper D., Raghavan, Prabhakar., & Scutze, Hinrich. 2007. *An introduction of information retrieval*. Cambridge university press: Cambridge.

Money, Raymond J. 2006. *Machine Learning Text Categorization*. University of texas: Austin.

Rosell, Magnus.(2005). *Improving Clustering of Swedish Newspaper Articles using Stemming and Compound Splitting*. Royal Institute of technology: Sweden.

Tala, Fazdillah Z. 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. http://www.illc.uva.nl/Publications/ResearchReports/MoL-2003-02.text.pdf. akses pada 19-11-2009.

Yogatama, Dani. 2008. *Studi Penggunaan Stemming untuk Meningkatkan performansi Sistem Temu Balik Informasi*. Central Library Institute Technology Bandung: Bandung.

LAMPIRAN

Lampiran 1. Daftar *Stop Word*

ada	bagai	bersama-sama	Di	ialah
adalah	bagaikan	betulkah	Dia	ibarat
adanya	bagaimana	biasa	Dialah	ingin
adapun	bagaimanakah	biasanya	Diantara	inginkah
agak	bagaimanapun	bila	diantaranya	inginkan
agaknya	bagi	bilakah	dikarenakan	ini
agar	bahkan	bisa	dini	inikah
akan	bahwa	bisakah	diri	inilah
akankah	bahwasanya	boleh	dirinya	itu
akhirnya	banyak	bolehkah	disini	itukah
aku	beberapa	bolehlah	disinilah	itulah
akulah	begini	buat	dong	jangan
amat	beginian	bukan	dulu	jangankan
amatlah	beginikah	bukankah	enggak	janganlah
anda	beginilah	bukanlah	enggaknya	jika
andalah	begitu	bukannya	entah	jikalau
antar	begitukah	cuma	entahlah	juga
antara	begitulah	dahulu	hal	justru
antaranya	begitupun	dalam	hampir	kala
apa	belum	dan	hanya	kalau
apaan	belumlah	dapat	hanyalah	kalaulah
apabila	berapa	dari	harus	kalaupun
apakah	berapakah	daripada	haruslah	kalian
apalagi	berapalah	dekat	harusnya	ialah
apatah	berapapun	demi	hendak	ibarat
atau	berkali-kali	demikian	hendaklah	ingin
ataukah	bermacam	demikianlah	hendaknya	inginkah
ataupun	bermacam-macam	dengan	hingga	inginkan
	bersama	depan	ia	ini

Lampiran 2. Tabel uji *stemming*

Keterangan :

: kata dasar yang tidak seharusnya

imbuhan		kata berimbuhan	Arifin	nazief
awalan	akhiran			
di-		diberi	Beri	beri
		diterima	Terima	terima
		dicari	Cari	cari
		diharap	Harap	harap
		dihantam	Hantam	hantam
di-	-kan	diberikan	Ikan	beri
		dicarikan	Carik	cari
		diharapkan	Harap	harap
		dipukulkan	Pukul	pukul
		disamakan	samak	sama
di-	-kan, -nya	diberikannya	ikan	beri
		dicarikannya	carik	cari
		diharapkannya	harap	harap
		dipukulkannya	pukul	pukul
		disamakannya	samak	sama
di-, per-		dipersulit	sulit	sulit
		dipercepat	cepat	cepat
		diperluas	luas	luas
		diperpanjang	panjang	panjang
		diperalat	alat	ralat
di-, per-	kan-	dipermainkan	main	main
		dipertemukan	temu	temu
		dipersatukan	satu	satu

		dipertanyakan	tanya	tanya
		dipermalukan	malu	malu
di-, per-	-kan, -nya	dipermainkannya	main	main
		dipertemukannya	temu	temu
		dipersatukannya	satu	satu
		dipertanyakannya	tanya	tanya
		dipermalukannya	malu	malu
di-, ber-	-kan	diberlakukan	laku	laku
		diberhentikan	henti	henti
di-, ber-	-kan, -nya	diberlakukannya	laku	laku
		diberhentikannya	henti	henti
ke-		kekasih	kasih	kasih
		kehendak	hendak	hendak
		ketabrak	tabrak	tabrak
		kepergok	pergok	pergok
		ketemu	ketemu	ketemu
ke-	-nya	kekasihnya	kasih	kasih
		kehendaknya	hendak	hendak
		keduanya	dua	dua
		ketiganya	tiga	tiga
		kelimanya	lima	lima
ke-	-nya, -lah	kekasihnyalah	kasih	kasih
		kehendaknyalah	hendak	hendak
		keduanyalah	dua	dua
		ketiganyalah	tiga	tiga
		kelimanyalah	lima	lima
ke-	-ku	kekasihku	kasih	kasih
		kehendakku	hendak	hendak
ke-	-ku,-lah	kekasihkulah	kasih	kasih
		kehendakkulah	hendak	hendak

ke-	-mu	kekasihmu	kasih	kasih
		kehendakmu	hendak	hendak
ke-	-mu,-lah	kekasihmulah	kasih	kasih
		kehendakmulah	hendak	hendak
ke-	-an	kenaikan	naik	kenaikan
		kepunyaan	punya	punya
		kemarahan	marah	marah
		keamanan	aman	aman
		kenakalan	nakal	nakal
ke-	-an,-nya	kenaikannya	naik	kenaikannya
		kepunyaannya	punya	punya
		kemarahannya	marah	marah
		keamanannya	aman	aman
		kenakalannya	nakal	nakal
ke-	-an,-nya,-lah	kenaikannyalah	naik	kenaikannyalah
		kepunyaannyalah	punya	punya
		kemarahannyalah	marah	marah
		keamanannyalah	aman	aman
		kenakalannyalah	nakal	nakal
ke-	-kah	kemanakah	mana	mana
ke-,ber-	-an	kebersamaan	sama	sama
ke-,ber-	-an,-nya	kebersamaannya	sama	sama
ke-,ber-	-an,-nya,-lah	kebersamaannyalah	sama	sama
ke-	-an,-ku,-pun	kepergiankupun	gi	pergi
se-		sekamar	kamar	kamar
		sepandai	pandai	pandai
		setinggi	setinggi	setinggi
		setelah	telah	setelah
		sepergi	pergi	pergi
		seizin	izin	izin

se-	-nya	sepandainya	pandai	pandai
		setingginya	tinggi	setinggi
		setelahnya	telah	telah
		seperginya	gi	pergi
		seizinnya	izin	izin
se-	-nya,-lah	seizinnyalah	izin	izin
se-,per-	-an	sepermainan	main	main
se-,per-	-an,-nya	sepermainannya	main	main
te-		terasa	rasa	rasa
		terangsang	rangsang	rangsang
ter-		terambil	ambil	ambil
		tertekan	tekan	tekan
		terlambat	lambat	lambat
		tertinggi	tinggi	tinggi
		teriris	iris	iris
		terganti	ganti	ganti
be-		berambut	rambut	rambut
		berasa	rasa	rasa
		berantai	rantai	beranta
		berebut	rebut	rebut
		bekerja	kerja	kerja
ber-		beribu	ribu	ribu
		bertamu	tamu	ta
		berilmu	ilmu	beril
		berkata	kata	kata
		berjanji	janji	janji
ber-,pe-	-an	berperasaan	asa	rasa
ber-	-nya,-lah	bersamanyalah	sama	sama
ber-	-mu,-lah	bersamamulah	sama	sama
ber-	-pun	bersamapun	sama	sama

bel-		belajar	ajar	ajar
me-		memakan	makan	makan
		memasak	masak	masak
		merusak	rusak	rusak
		merampok	rampok	rampok
		mewarna	warna	warna
		merasa	rasa	rasa
me-	-i	menaiki	tik	naik
		mewarnai	warna	warna
meng-		menggambar	gambar	gambar
		mengikis	kikis	kikis
		mengharap	harap	harap
		mengupas	kupas	kupas
		mengirim	kirim	kirim
mem-,per-		memperpanjang	panjang	memperpanjang
mem-,per-	-kan,-nya	mempertanyakannya	tanya	mempertanyakannya
mem-,per-	-kan	mempertontonkan	tonton	mempertontonkan
mem-,per-	-kan,-nya,-lah	mempermainkannyalah	main	mempermainkannyalah
mem-,ber-	-kan,-nya	memberlakukannya	laku	memberlakukannya
mem-,ber-	-kan	memberdayakan	daya	memberdayakan
mem-,ber-	-kan,-nya,-lah	memberdayakannyalah	daya	memberdayakannyalah
men-		menjadi	menjadi	jadi
		menangis	tangis	tangis
		menampar	tampar	tampar
		menulis	tulis	tulis
		menari	tari	tari
pel-		pelajar	ajar	ajar
pe-		perasa	rasa	rasa

		pemilik	milik	milik
		pewarna	warna	warna
		pelukis	lukis	lukis
		pemuda	pemuda	pemuda
peng-		pengganggu	ganggu	ganggu
		pengurang	kurang	urang
		penghapus	hapus	hapus
		pengirim	kirim	pengirim
		pengajar	ajar	ajar
peny-		penyapu	sapu	sapu
		penyisir	sisir	sisir
		penyendiri	sendiri	sendiri
		penyayang	sayang	sayang
		penyambut	sambut	sambut
pem-		pemanah	manah	manah
		pemakai	pakai	pakai
		pemberi	beri	beri
		pemahat	pahat	pahat
		pembeli	beli	beli
pem- ,ber-	-an	pemberlakuan	laku	pemberlakuan
		pemberdayaan	daya	pemberdayaan
pem-	-an	pembelian	belian	belian
pen-		penanam	tanam	tanam
		pendaki	aki	pendak
		penari	tari	tari
		penerjemah	terjemah	terjemah
		penerbit	terbit	terbit
per-	-an	pertemanan	teman	teman
		perjuangan	juang	juang
		permainan	main	main

		pergaulan	gaul	gaul
		persamaan	sama	sama
meng-	-kan	mengerjakan	kerja	kerja
meng-	-i	mengendarai	kendara	kendara
	-an	makanan	makan	makan
		lipatan	lipat	lipat
		imbuhan	imbuh	imbuh
		ucapan	ucap	ucap
		buruan	buru	buru
	-an,-nya	peranannya	peran	peran
		makanannya	makan	makan
		lipatannya	lipat	lipat
		mainannya	main	main
		ucapannya	ucap	ucap
	-an,-nya,-lah	peranannyalah	peran	peran
		makanannyalah	makan	makan
		lipatannyalah	lipat	lipat
		mainannyalah	main	main
		ucapannyalah	ucap	ucap
	-an,-nya,-pun	peranannyapun	peran	peran
		makanannyapun	makan	makan
		lipatannyapun	lipat	lipat
		mainannyapun	main	main
		ucapannyapun	ucap	ucap
	-an,-mu	perananmu	peran	peran
		makananmu	makan	makan
		lipatanmu	lipat	lipat
		mainanmu	main	main
		ucapanmu	ucap	ucap
	-an,-mu,-lah	perananmulah	peran	peran

		makananmulah	makan	makan
		lipatanmulah	lipat	lipat
		mainanmulah	main	main
		ucapanmulah	ucap	ucap
	-an,-mu,-pun	perananmupun	peran	peran
		makananmupun	makan	makan
		lipatanmupun	lipat	lipat
		mainanmupun	main	main
		ucapanmupun	ucap	ucap
	-an,-ku	perananku	peran	peran
		makananku	makan	makan
		lipatanku	lipat	lipat
		mainanku	main	main
		ucapanku	ucap	ucap
	-an,-ku,-lah	peranankulah	peran	peran
		makanankulah	makan	makan
		lipatankulah	lipat	lipat
		mainankulah	main	main
		ucapankulah	ucap	ucap
	-an,-ku,-pun	peranankupun	peran	peran
		makanankupun	makan	makan
		lipatankupun	lipat	lipat
		mainankupun	main	main
		ucapankupun	ucap	ucap
	-kan	terangkan	terangkan	terang
		tinggikan	tinggi	tinggi
		naikkan	naik	naik
		benarkan	benarkan	benar
		singkirkan	singkir	singkir
	-kan,-lah	terangkanlah	terangkanlah	terang

		tinggikanlah	tinggi	tinggi
		naikkanlah	naik	naik
		benarkanlah	benarkanlah	benar
		singkirkanlah	singkir	singkir
	-i	percayai	caya	percaya
		terangi	terang	terang
		cintai	cinta	cinta
		kasihi	kasih	kasih
		sayangi	sayang	sayang
	-i,-lah	percayailah	caya	percaya
		terangilah	terang	terang
		cintailah	cinta	cinta
		kasihilah	kasih	kasih
		sayangilah	sayang	sayang
	-kah	adakah	ada	ada
		kapankah	kapan	kapan
		bagaimanakah	bagaimana	bagaimana
		siapakah	siapa	siapa
		berapakah	apakah	berapa
	-lah	kamulah	kamu	kamu
		terimalah	terima	terima
		simpanlah	simpan	simpan
		dirinyalah	diri	diri
		berikanlah	ikan	berik
	-pun	diapun	dia	dia
		sayapun	saya	saya
		kamupun	kamu	kamu
		merekapun	reka	mereka
		kalianpun	kalian	kalian
	-ku	diriku	diri	diri
		rumahku	rumah	rumah

		uangku	uang	uang
		milikku	milik	milik
		tamanku	taman	taman
	-ku,-lah	dirikulah	diri	diri
		rumahkulah	rumah	rumah
		uangkulah	uang	uang
		milikkulah	milik	milik
		tamankulah	taman	taman
	-ku,-pun	dirikupun	diri	diri
		rumahkupun	rumah	rumah
		uangkupun	uang	uang
		milikkupun	milik	milik
		tamankupun	taman	taman
	-mu	dirimu	diri	diri
		rumahmu	rumah	rumah
		uangmu	uang	uang
		milikmu	milik	milik
		tamanmu	taman	taman
	-mu,-pun	dirimupun	diri	diri
		rumahmupun	rumah	rumah
		uangmupun	uang	uang
		milikmupun	milik	milik
		tamanmupun	taman	taman
	-mu,-lah	dirimulah	diri	diri
		rumahmulah	rumah	rumah
		uangmulah	uang	uang
		milikmulah	milik	milik
		tamanmulah	taman	taman
	-nya	dirinya	diri	diri
		rumahnya	rumah	rumah
		uangnya	uang	uang

		miliknya	milik	milik
		tamannya	taman	taman
	-nya,-lah	dirinyalah	diri	diri
		rumahnyalah	rumah	rumah
		uangnyalah	uang	uang
		miliknyalah	milik	milik
		tamannyalah	taman	taman
	-nya,-pun	dirinyapun	diri	diri
		rumahnyapun	rumah	rumah
		uangnyapun	uang	uang
		miliknyapun	milik	milik
		tamannyapun	taman	taman

Lampiran 3. Tabel nilai similaritas dokumen uji terhadap dokumen latih dengan menggunakan *stemming* Arifin

	C1	C2	C3	C4	C5	C6
D1	0,109	0,071	0,069	0,081	0,063	0,353
D2	0,05	0,045	0,293	0,07	0,043	0,03
D3	0,06	0,307	0,047	0,042	0,048	0,047
D4	0,109	0,037	0,288	0,058	0,045	0,048
D5	0,054	0,031	0,235	0,035	0,031	0,039
D6	0,093	0,045	0,064	0,05	0,311	0,044
D7	0,063	0,262	0,034	0,051	0,046	0,051
D8	0,084	0,078	0,093	0,075	0,506	0,056
D9	0,055	0,054	0,069	0,043	0,049	0,339
D10	0,058	0,044	0,051	0,347	0,041	0,078
D11	0,079	0,038	0,228	0,041	0,037	0,029
D12	0,06	0,04	0,039	0,069	0,043	0,213
D13	0,062	0,214	0,03	0,04	0,036	0,036
D14	0,062	0,228	0,068	0,044	0,045	0,044
D15	0,229	0,048	0,084	0,074	0,063	0,075
D16	0,08	0,055	0,08	0,046	0,272	0,045
D17	0,03	0,035	0,035	0,151	0,042	0,047
D18	0,104	0,051	0,055	0,063	0,051	0,32
D19	0,167	0,143	0,064	0,063	0,062	0,076
D20	0,1	0,053	0,08	0,079	0,295	0,069
D21	0,137	0,055	0,173	0,07	0,065	0,05
D22	0,156	0,071	0,065	0,046	0,039	0,045
D23	0,061	0,037	0,049	0,213	0,054	0,049
D24	0,059	0,036	0,05	0,215	0,053	0,048
D25	0,059	0,06	0,059	0,345	0,043	0,075
D26	0,082	0,067	0,087	0,062	0,318	0,056
D27	0,043	0,138	0,034	0,025	0,036	0,03
D28	0,044	0,139	0,035	0,025	0,036	0,03

D29	0,182	0,067	0,093	0,057	0,066	0,076
D30	0,058	0,056	0,057	0,283	0,041	0,062
D31	0,081	0,045	0,06	0,179	0,047	0,059
D32	0,072	0,057	0,061	0,083	0,151	0,076
D33	0,054	0,029	0,045	0,169	0,042	0,047
D34	0,062	0,041	0,048	0,058	0,038	0,14
D35	0,046	0,018	0,028	0,271	0,031	0,042
D36	0,084	0,239	0,06	0,087	0,06	0,068
D37	0,284	0,053	0,061	0,046	0,059	0,058
D38	0,152	0,033	0,095	0,044	0,064	0,025
D39	0,07	0,051	0,068	0,053	0,231	0,06
D40	0,25	0,064	0,098	0,082	0,086	0,092
D41	0,12	0,045	0,042	0,078	0,039	0,311
D42	0,056	0,037	0,048	0,211	0,04	0,055
D43	0,338	0,034	0,122	0,054	0,069	0,041
D44	0,063	0,031	0,323	0,058	0,04	0,048
D45	0,086	0,245	0,04	0,042	0,046	0,044
D46	0,067	0,041	0,416	0,056	0,041	0,035
D47	0,141	0,031	0,23	0,04	0,044	0,046
D48	0,093	0,041	0,07	0,053	0,372	0,052
D49	0,068	0,034	0,312	0,072	0,055	0,042
D50	0,102	0,041	0,078	0,175	0,102	0,053
D51	0,067	0,225	0,046	0,105	0,051	0,054
D52	0,079	0,049	0,07	0,044	0,054	0,252
D53	0,15	0,04	0,098	0,064	0,113	0,046
D54	0,086	0,044	0,064	0,05	0,251	0,041
D55	0,097	0,082	0,177	0,061	0,051	0,079
D56	0,04	0,027	0,026	0,033	0,03	0,182
D57	0,276	0,025	0,034	0,034	0,028	0,025
D58	0,112	0,044	0,341	0,041	0,051	0,046
D59	0,031	0,166	0,023	0,028	0,022	0,039

D60	0,039	0,154	0,025	0,033	0,032	0,025
D61	0,047	0,032	0,033	0,047	0,031	0,342

Lampiran 4. Tabel nilai similaritas dokumen uji terhadap dokumen latih dengan menggunakan stemming Nazief

	C1	C2	C3	C4	C5	C6
D1	0,117	0,073	0,07	0,079	0,07	0,357
D2	0,054	0,053	0,291	0,074	0,047	0,037
D3	0,104	0,317	0,057	0,06	0,06	0,079
D4	0,108	0,042	0,306	0,058	0,051	0,049
D5	0,058	0,029	0,223	0,037	0,039	0,037
D6	0,095	0,045	0,063	0,051	0,326	0,048
D7	0,06	0,289	0,036	0,052	0,048	0,053
D8	0,089	0,086	0,098	0,083	0,544	0,062
D9	0,063	0,065	0,071	0,044	0,051	0,342
D10	0,058	0,044	0,056	0,346	0,047	0,081
D11	0,09	0,041	0,245	0,044	0,044	0,032
D12	0,064	0,042	0,04	0,069	0,044	0,215
D13	0,062	0,224	0,032	0,036	0,035	0,032
D14	0,06	0,223	0,079	0,04	0,05	0,047
D15	0,23	0,06	0,091	0,075	0,066	0,079
D16	0,096	0,061	0,084	0,058	0,302	0,057
D17	0,035	0,04	0,037	0,167	0,044	0,052
D18	0,106	0,054	0,056	0,062	0,056	0,33
D19	0,172	0,161	0,064	0,061	0,069	0,082
D20	0,104	0,058	0,088	0,079	0,308	0,071
D21	0,15	0,062	0,181	0,077	0,075	0,057
D22	0,176	0,065	0,081	0,045	0,045	0,047
D23	0,06	0,042	0,051	0,228	0,053	0,045
D24	0,06	0,042	0,052	0,243	0,053	0,046

D25	0,059	0,062	0,057	0,349	0,045	0,078
D26	0,087	0,08	0,086	0,06	0,335	0,066
D27	0,045	0,154	0,035	0,024	0,039	0,032
D28	0,046	0,155	0,036	0,023	0,04	0,033
D29	0,193	0,065	0,098	0,061	0,07	0,079
D30	0,067	0,064	0,062	0,281	0,047	0,067
D31	0,087	0,049	0,069	0,176	0,05	0,061
D32	0,086	0,064	0,078	0,093	0,21	0,083
D33	0,062	0,03	0,045	0,184	0,043	0,049
D34	0,069	0,045	0,048	0,076	0,039	0,152
D35	0,049	0,02	0,028	0,294	0,033	0,043
D36	0,094	0,242	0,069	0,098	0,066	0,069
D37	0,287	0,065	0,064	0,046	0,058	0,056
D38	0,176	0,044	0,122	0,06	0,092	0,038
D39	0,077	0,054	0,071	0,057	0,244	0,069
D40	0,272	0,063	0,102	0,082	0,09	0,093
D41	0,127	0,051	0,04	0,079	0,041	0,318
D42	0,063	0,038	0,053	0,211	0,043	0,063
D43	0,329	0,04	0,12	0,056	0,071	0,047
D44	0,069	0,037	0,34	0,063	0,049	0,053
D45	0,098	0,26	0,046	0,048	0,054	0,05
D46	0,07	0,042	0,417	0,059	0,042	0,037
D47	0,146	0,038	0,238	0,043	0,056	0,048
D48	0,102	0,047	0,083	0,056	0,379	0,051
D49	0,073	0,039	0,318	0,077	0,066	0,045
D50	0,101	0,044	0,081	0,168	0,106	0,053
D51	0,071	0,244	0,046	0,116	0,055	0,057
D52	0,076	0,053	0,075	0,043	0,047	0,252
D53	0,161	0,041	0,104	0,064	0,131	0,047
D54	0,083	0,041	0,068	0,047	0,276	0,04
D55	0,09	0,078	0,186	0,059	0,052	0,078

D56	0,039	0,025	0,03	0,03	0,032	0,181
D57	0,297	0,025	0,033	0,032	0,027	0,021
D58	0,125	0,045	0,366	0,053	0,05	0,045
D59	0,031	0,17	0,031	0,028	0,023	0,041
D60	0,039	0,159	0,024	0,034	0,033	0,025
D61	0,046	0,035	0,033	0,046	0,03	0,347

Lampiran 5. Tabel nilai similaritas dokumen uji terhadap dokumen latih tanpa menggunakan *stemming*

	C1	**C2**	**C3**	**C4**	**C5**	**C6**
D1	0,089	0,066	0,055	0,074	0,057	0,332
D2	0,044	0,041	0,278	0,058	0,031	0,02
D3	0,051	0,318	0,034	0,037	0,04	0,043
D4	0,097	0,039	0,279	0,053	0,04	0,045
D5	0,046	0,032	0,215	0,031	0,029	0,029
D6	0,087	0,05	0,061	0,051	0,282	0,038
D7	0,059	0,219	0,029	0,045	0,043	0,042
D8	0,075	0,077	0,071	0,062	0,491	0,043
D9	0,041	0,036	0,052	0,035	0,034	0,276
D10	0,053	0,041	0,041	0,34	0,036	0,065
D11	0,062	0,037	0,211	0,032	0,033	0,022
D12	0,047	0,044	0,034	0,059	0,035	0,191
D13	0,058	0,199	0,025	0,036	0,038	0,03
D14	0,057	0,24	0,04	0,028	0,042	0,032
D15	0,206	0,044	0,08	0,064	0,055	0,059
D16	0,059	0,045	0,063	0,038	0,248	0,035
D17	0,023	0,033	0,029	0,148	0,04	0,041
D18	0,099	0,052	0,047	0,062	0,047	0,301
D19	0,169	0,111	0,064	0,051	0,062	0,046
D20	0,097	0,054	0,056	0,075	0,278	0,055

D21	0,117	0,045	0,161	0,056	0,053	0,036
D22	0,136	0,069	0,055	0,038	0,028	0,036
D23	0,053	0,038	0,045	0,199	0,046	0,039
D24	0,052	0,038	0,047	0,202	0,046	0,038
D25	0,043	0,059	0,044	0,314	0,035	0,057
D26	0,069	0,07	0,066	0,053	0,304	0,046
D27	0,032	0,138	0,023	0,02	0,028	0,022
D28	0,032	0,139	0,023	0,021	0,028	0,022
D29	0,175	0,067	0,087	0,039	0,049	0,071
D30	0,053	0,042	0,047	0,258	0,036	0,052
D31	0,07	0,04	0,047	0,165	0,036	0,048
D32	0,072	0,064	0,06	0,082	0,152	0,063
D33	0,046	0,03	0,043	0,172	0,039	0,042
D34	0,058	0,039	0,046	0,053	0,032	0,134
D35	0,04	0,019	0,024	0,274	0,024	0,032
D36	0,074	0,247	0,052	0,072	0,049	0,06
D37	0,276	0,053	0,047	0,039	0,049	0,045
D38	0,145	0,026	0,082	0,037	0,045	0,02
D39	0,063	0,051	0,062	0,048	0,227	0,049
D40	0,228	0,061	0,077	0,062	0,072	0,072
D41	0,12	0,048	0,034	0,071	0,034	0,31
D42	0,048	0,036	0,043	0,198	0,032	0,049
D43	0,349	0,037	0,123	0,054	0,071	0,039
D44	0,05	0,031	0,297	0,052	0,037	0,037
D45	0,059	0,232	0,034	0,028	0,041	0,036
D46	0,062	0,044	0,412	0,055	0,036	0,026
D47	0,144	0,031	0,224	0,035	0,044	0,035
D48	0,084	0,041	0,064	0,047	0,36	0,034
D49	0,061	0,036	0,313	0,064	0,053	0,032
D50	0,092	0,043	0,071	0,155	0,084	0,043
D51	0,058	0,212	0,041	0,081	0,045	0,045

D52	0,066	0,044	0,062	0,038	0,045	0,223
D53	0,121	0,03	0,075	0,051	0,085	0,031
D54	0,078	0,045	0,052	0,039	0,246	0,036
D55	0,08	0,074	0,159	0,044	0,038	0,063
D56	0,036	0,031	0,021	0,026	0,019	0,182
D57	0,261	0,026	0,029	0,028	0,023	0,018
D58	0,095	0,043	0,327	0,034	0,049	0,036
D59	0,025	0,19	0,02	0,026	0,021	0,037
D60	0,035	0,171	0,023	0,033	0,03	0,021
D61	0,043	0,033	0,023	0,041	0,026	0,351