Pembangunan Daftar *Stopword* Menggunakan Pendektan Term-Based Random Sampling Untuk Kategorisasi Dokumen Berbahasa Inggris

Anju Vikhers [1], Moch. Arif Bijaksana, Ir.,MTECH [2], Yanuar Firdaus A. W., ST., MT[3]

[1,2,3]Departemen Teknik Informatika Institut Teknologi Telkom, Bandung

[1]vikhers@gmail.com, [2]mab@stttelkom.ac.id, [3]yfa@stttelkom.ac.id

Abstrak

Kata-kata dalam suatu dokumen yang sering muncul tapi kurang berarti dalam proses kategorisasi disebut sebagai *stopword.* Untuk kata-kata yang dikategorisasikan ke dalam *stopwords* dianggap tidak memiliki kontribusi dalam proses kategorisasi, seharusnya dihapus sewaktu pengindeksan sebelum proses kategorisasi dilakukan. Bagaimanapun, penggunaan satu daftar *stopword* untuk koleksi dokumen yang berbeda-beda bisa mengurangi performansi dari peng-kategorisasian.

Pada tugas akhir ini digunakan pendekatan *Term-Based Random Sampling* menghasilkan daftar *stopword* secara otomatis untuk dokumen yang diberikan. Pendekatan ini, menentukan seberapa besar informasi yang dimiliki suatu kata (*term*). Dengan ini akan bisa ditentukan suatu daftar *stopword* secara otomatis. Dalam tugas akhir ini digunakan koleksi dokumen Reuter. Untuk daftar *stopword* yang dihasilkan akan di evaluasi dengan melakukan kategorisasi pada dokumen yang menggunakan daftar *stopword* yang dihasilkan.

Pendekatan ini juga nanti akan dievaluasi dengan membandingkan hasil performansi kategorisasi yang dihasilkan dengan pre-proses pembuangan daftar *stopword* menggunakan daftar *stopword* yang dihasilkan dengan menggunakan pendekatan ini, hasil performansi kategorisasi menggunakan daftar *stopword* Salton and Buckley I dan Salton and Buckley II, Google *stopword*, *defalutl English Stopword* dan hasil performansi kategorisasi tanpa menggunakan pre-proses pembuangan *stopword.*

Dari hasil evaluasi yang dilakukan, daftar *stopword* yang lebih efektif bisa diperoleh dengan menggunakan pendekatan Term-Based Random Sampling. Dengan akurasi pengkategorisasi sebesar 88.24%.

Keyword
Kategorisasi, *term, stopword*

Abstract

Words in a document that frequently occuring but meaningless in categorization are called as stopwords. Words that categorize as stopwords do not contribute for categorization, they should be remove during indexing before categorization process. However, using a single fixed stopword list across diffrent document collection could be decrimental the performansi of categorization.

On this final project, Term-Based Random sampling is used as approach to provide stopword list automatically for document collection that is processed. This approach, define how informative a term is. So, it's enable us to define a stopword list automatically. The *stopword* list that is produced will evaluate by categorization step for document that use this stopword list.

This approach will evaluate by comparing the result of categorization that produce by use preprocessing using stopword list that produce by this approach with the performansi using stopword Salton and Buckley I, Salton and Buckley II, Google's stopword, default English Stopword, and also camparing with categorization's performansi without using stopword.

From the evaluation, better stopword list is coming from stopword list that produce by using Term-Based Random Sampling Approach. The accuration is 88,24%.

Keyword
Categorization, *term, stopword*

1. Pendahuluan

1.1 Latar Belakang

Adanya berita elektronik berbahasa Inggris merupakan salah satu penyajian informasi pada *digital library* yang mempunyai peranan sangat penting terutama dalam meningkatkan kegunaan dari sekumpulan dokumen. Dokumen-dokumen tersebut tentunya mengandung berbagai informasi yang berharga yang dapat dimanfaatkan oleh banyak pihak. Suatu dokumen merupakan sumber informasi yang memiliki nilai yang berharga. Sebenarnya secara eksplisit, bisa saja mengetahui informasi pada suatu dokumen dengan mudah, salah satu contohnya dengan melihat judul yang ada pada bagian dokumen tersebut. Namun dokumen-dokumen yang ada mempunyai nilai informasi yang bersifat implisit dan pada umumnya tidak pernah diperhatikan secara seksama. Sedangkan informasi inilah yang sebenarnya merupakan pengetahuan yang mempunyai informasi sangat berguna yang perlu kita peroleh.

Text mining merupakan bagian dari data mining yang memfokuskan bidangnya pada pengambilan informasi berharga dari basis data yang berupa *text*. Kategorisasi adalah salah satu teknik dari *text mining* yang bertujuan untuk menentukan topik dari suatu artikel atau *text document* berdasarkan atribut kelasnya, kategorisasi bersifat *supervised*. Untuk memperoleh hasil kategorisasi yang baik maka diperlukan suatu preproses *stopwords removal* yaitu penghapusan kata-kata yang sering muncul tapi tidak memiliki kontribusi terhadap informasi suatu dokumen.

Dengan terus bertambahnya jumlah dokumen pada *digital library*, pengkategorian secara manual tentu saja akan menjadi suatu masalah baru untuk pengguna. Hal tersebut akan memakan banyak waktu dan menimbulkan kejenuhan. Oleh karena itu, akan digunakan teknik *text mining* yaitu kategorisasi untuk mengkategorikan dokumen berbahasa Inggris. Sebelum proses kategorisasi dilakukan, perlu dilakukan proses *stopwords removal* terlebih dahulu pada dokumen. Maka diperlukan daftar *stopword* yang baik yang digunakan pada proses *stopword removal* ini, agar hasil pengkategorisasian yang dihasilkan bisa lebih baik.

Pada tugas akhir ini akan dibahas tentang pembangunan daftar stopword yang akan digunakan dalam proses *stopword removal* yang akan mempengaruhi hasil kategorisasi. Salah satu pendekatan yang digunakan untuk memperoleh daftar stopword adalah pendekatan *Term Based Random Sampling*. Pendekatan *Term-Based Random Sampling* ini bisa mengukur seberapa pentingnya suatu *term* dalam suatu dokumen yaitu dengan menghitung bobot. Bobot dari suatu *term* bisa dihitung dengan menggunakan *Kullback-Leibler Divergence Measure*. Semakin kecil bobot suatu *term* maka semakin cocok *term* tersebut disebut sebagai *stopword*. Dengan menggunakan pendekatan *Term-Based Random Sampling* ini maka bisa diperoleh suatu daftar *stopword* yang lebih efektif.

1.2 Perumusan masalah

Berdasarkan latar belakang, permasalahan yang dijadikan objek penelitian dan pengembangan tugas akhir ini adalah sebagai berikut :

1. Bagaimana menghasilkan daftar stopword pada dokumen yang digunakan dengan menggunakan pendekatan *Term-Based Random Sampling* untuk kategorisasi dokumen berbahasa Inggris.
2. Bagaimana mengaplikasikan pre-proses dengan *stopwords removal* dengan daftar *stopword* yang diperoleh dengan pendekatan *Term-Based Random Sampling* dan daftar *stopword* bahasa Inggris yang sudah didefenisikan.
3. Bagaimana mengukur dan menganalisis performansi kategorisasi setelah dilakukan *stopwords removal* dengan *Term-Basd Random Sampling* untuk kategorisasi dokumen berbahasa Inggris berdasarkan F-Measure.

Batasan masalah dalam tugas akhir ini adalah :

1. Metode kategorisasi yang digunakan adalah Naïve Bayes yang ada pada Software WEKA.
2. Dokumen yang digunakan adalah dokumen berbahasa Inggris yang bersifat offline yaitu dokumen Reuters dengan jumlah 200 dokumen.
3. Dokumen yang akan digunakan dilakukan proses stemming terlebih dahulu.

1.3 Tujuan

Berdasarkan rumusan masalah diatas, tujuan yang ingin dicapai dari tugas akhir ini adalah :

1. *Mengelola* Menghasilkan daftar *stopword* pada dokumen yang digunakan dengan menggunakan pendekatan *Term-Based Random Sampling* untuk kategorisasi dokumen berbahasa Inggris.
2. Mengaplikasikan pre-proses dengan *stopwords removal* dengan daftar *stopword* yang diperoleh dengan pendekatan Term-Based Random Sampling dan daftar

stopword bahasa Inggris yang sudah didefenisikan.

3. Mengukur dan menganalisis performansi kategorisasi setelah dilakukan *stopwords* removal untuk kategorisasi dokumen berbahasa Inggris berdasarkan *F-Measure.*

2. Dasar Teori

2.1 Data Mining

Data mining adalah proses mengekstrak pengetahuan yang berguna dari jumlah data yang besar. *Data mining* bertujuan untuk menemukan pola dalam basis data yang berkapasitas besar untuk mendukung dalam pengambilan keputusan. *Data mining* menganalisis data menggunakan tool untuk menemukan pola dan aturan dalam himpunan data. Tool *data mining* diharapkan mampu mengenal pola ini dalam data dengan input minimal dari user.[6]

2.2 Fungsionalitas Data Mining

Data mining dapat diklasifikasikan sesuai dengan fungsi yang dilakukan yaitu :

1. *Characterization*

   Memberikan ringkasan mengenai karakteristik objek-objek dalam suatu kelas tertentu. Contoh : menentukan karakteristik pelanggan potensial.

2. *Discrimination*

   Membandingkan karakteristik objek-objek pada kelas tertentu dengan kelas lainnya. Contoh : membandingkan pelanggan potensial dengan yang tidak.

3. *Association*

   Mempelajari frekuensi item-item yang terjadi secara bersamaan dalam transaksi database. Contoh : beli (x, roti) $\rightarrow$ beli (x, mentega).

4. *Classification*

   Mengklasifikasikan data ke dalam kelas yang diberikan berdasarkan nilai atribut (*supervised classification*). Kelas-kelas klasifikasi sudah didefinisikan dahulu. Tipe datanya biasanya bersifat kategorik. Contoh : mengklasifikasikan produk berdasar respon terhadap iklan : baik, biasa, tidak ada respon.

5. *Prediction*

   Meramalkan nilai atribut yang hilang atau tidak diketahui berdasarkan informasi lain. Tipe datanya bersifat kontinyu. Contoh : meramalkan apakah suatu produk akan laku atau tidak berdasarkan data yang ada.

6. *Clustering*

   Berbeda dengan *classification, clustering* (klasterisasi) merupakan *unsupervised classification*. Dalam klasterisasi, kelas-kelas tidak didefinisikan terlebih dahulu. Contoh : aplikasi pengelompokan dokumen untuk mesin pencarian.

7. *Outlier Analysis*

   Mengidentifikasi dan menjelaskan noise dan *outlier. Outlier* adalah objek data yang tidak memenuhi model dan persyaratan secara umum, yang berbeda dan inkonsisten dengan data set yang ada. Contoh aplikasi : *fraud detection*.

8. *Evolution analysis*

   Menggambarkan dan memodelkan regulasi atau tren untuk obyek yang berubah setiap saat. Mungkin melibatkan semua fungsionalias data mining yang lain.

2.3 Teks Mining

Teks mining merupakan pencarian pola yang menarik atau pola yang berguna pada sebuah informasi tekstual yang tidak terstruktur (teks natural language), atau bisa juga didefinisikan sebagai proses

dari menganalisa teks untuk mengekstraksi informasi dengan tujuan tertentu.

Kebanyakan pendekatan teks mining menggunakan algoritma mining pada atribut yang dihubungkan pada setiap dokumen. Atribut tersebut bisa berupa ekstraksi keyword dari dokumen atau hanya daftar kata dalam dokumen yang bersangkutan.

2.4 *Text Analysis*

> Pada *text analysis* sangat penting untuk dapat mengekstraksi feature dengan "benar" supaya didapatkan ketepatan dan hasil yang berguna. Pada tahap ini akan dihilangkan ambiguitas/kerancuan dari teks, dihitung frekuensi kata-kata atau frase yang menggambarkan isi dari masing-masing dokumen. Hanya kata-kata tertentu atau frase yang sesuai dengan aturan grammar yang akan diekstraksi.

Pada tahap ini akan dilakukan *parsing* atau segmentasi dokumen itu sehingga diperoleh daftar kata-kata yang ada di dalamnya. Daftar kata itu kemudian disaring dengan membuang kata-kata yang ada di daftar *stopword*. Kata-kata yang tersisa itu kemudian dihilangkan imbuhan-imbuhannya melalui proses *stemming* sehingga didapatkan daftar kata dasar yang dapat mewakili dokumen tersebut. Selain daftar kata juga akan didapatkan frekuensi dari kemunculan masing-masing kata/frase dalam suatu dokumen.

2.4.1 Kata henti (*Stopword)*

Merupakan kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Contoh stopwords adalah "at","to","an" dan seterusnya. Stopword juga didefenisikan sebagai sejumlah kata-kata yang dihilangkan sewaktu proses pengindeksan secara otomatis karena jika kata-kata ini tidak dihilangkan akan menghasilkan pengindeksan yang kurang baik. Atau ada juga yang menyebutkan bahwa *stopword* ini sebagai kata negatif atau noise.[8]

Jika kata henti atau stopword ini tidak dihilangkan maka proses kategorisasi bisa saja menghasilakan kelas suatu dokumen yang tidak tepat karena menggunakan term yang tidak relevan. Kata henti atau stopword ini juga akan mengambil bagian yang besar dalam suatu dokumen teks. Tapi dengan menghindarkan atau dengan melakukan pengapusan pada saat tahap pengindeksan akan mempercepat proses, menghemat tempat penyimpanan data dan meningkatkan akurasi dalam pengkategorisasian.[8]

Stopwords removal adalah sebuah proses untuk menghilangkan kata yang 'tidak relevan' pada hasil *parsing* sebuah dokumen teks dengan cara membandingkannya dengan *Stoplist* (*stopword list*) yang ada Contoh dari stopword misalnya, kata sambung, artikel dan preposisi.

Pada umumnya untuk membangun daftar *stopword* dilakukan dengan melihat frekuensi kemunculan dari setiap kata yang ada pada dokumen. Dan untuk kata yang memiliki frekuensi yang tinggi dengan nilai threshold tertentu maka diperoleh daftar *stopword.* Tapi, dengan meilihat sebatas dari frekuensi kata tidak lah cukup untuk bisa mendapatkan daftar *stopword* yang baik. Hal ini bisa di ilustrasikan oleh Bucley dengan kalimat "The head and president of an American computer system company based in Washington said she expected to make a million systems by the end of the year". Semua kata dalam kalimat tersebut, dalam lebih dari 10% dari dokumen yang ada di koleksi TREC dan menghilangkan kata-kata ini maka tidak akan meninggalkan satu katapun nantinya untuk diproses. Jadi untuk menghasilkan daftar stopword yang baik, maka disampingkan berdasarkan frekuensi dari term tapi juga melihat tingkat kepentingan dari kata tersebut dalam dokumen.[7]

Pada tugas akhir ini yang menjadi fokus pengerjaan adalah preproses untuk menghilangkan kata-kata yang tidak penting dalam suatu dokumen. Jadi dilakukan pembangunan daftar *stopword* yang cocok untuk dokumen yang akan diproses.

2.4.2 Pembangunan Daftar Stopword Secara Umum

Untuk memperoleh daftar stopword maka akan dilihat frekuensi kemunculan dari term-term yang ada dalam sebuah dokumen teks. Berikut langkah-langkah pembangunan daftar stopwordnya :

1. Semua kata-kata yang ada dalam koleksi dokumen di-extrak atau diambil semua.
2. Kemudian masing-masing kata-kata ini akan diurutkan berdasarkan frekuensi kemunculannya
3. Kata-kata yang paling sering muncul biasanya akan dianggap sebagai kata henti (stopword)

- Dari proses diatas akan dibentuk sejumlah kata-kata yang termasuk kata henti yang akan dibuang dalam proses berikutnya

2.4.3 Pendekatan *Term Based Random Sampling*

Salah satu algoritma yang digunakan dalam membangun daftar stopword adalah : **"***Term Based Random Sampling Approach***".**

1. Pendekatan ini didasarkan pada tingkat seberapa pentingnya suatu term dalam suatu dokumen.
2. Dengan pendekatan ini kita dapat menentukan apakah suatu *term* tertentu merupakan suatu stopword berdasarkan tingkat kepentingannya dalam dokumen.
3. Semakin kecil tingkat kepentingan suatu term atau kata maka term tersebut akan cenderung dikategorikan sebagai *stopword*.
4. Tingkat kepentingan suatu term dapat diakses dengan menggunakan Kullback-Leibler *divergence measure*. Berikut adalah rumus yang digunakan :

$$\omega(t) = Px * \log_2 \frac{Px}{Pc}$$

5. Idenya adalah menemukan term-term yang memiliki arti yang sama dengan istilah yang diberikan, kemudian menemukan semua dokumen-dokumen yang mengandung term-term yang diberikan dan dokumen ini digunakan sebagai dokumen sample untuk proses berikutnya dalam pembangunan daftar stopword.
6. Kemudian meng-ekstrak kata-kata yang memiliki informasi yang paling sedikit dari sampel dengan mengukur divergence dari distribusi kata-kata yang diberikan dengan himpunan dokumen sample dari distribusinya.
7. Dengan Kullback-Leibler *divergence measure*, maka bisa ditentukan tingkat kepentingan masing-masing term.

$$\omega(t) = Px * \log_2 \frac{Px}{Pc}$$

Untuk formula di atas, $Px = \frac{tfx}{lx}$ dan $Pc = \frac{F}{tokenc}$, dimana tfx adalah frekuensi dari query term dalam himpunan dokumen sampel, sedangakan lx merupakan panjang dari himpunan dokumen sampel, F adalah frekuensi dari wuery term di dalam keseluruhan koleksi dokumen, $tokenc$ adalah jumlah total token dalam keseluruhan koleksi dokumen.[8]

2.4.4 Algoritma dari Term Based Random Sampling Approach

Ulang Y kali, dimana Y adalah sebuah parameter:

1. Secara acak pilih sebuah terms dalam file kamus, kita sebut sebagai ω_{random}
2. Retrieve/ambil semua dokumen dalam corpus/kitab yang terdiri dari ω_{random}
3. Dengan menggunakan Kullback_Leibler divergence measure untuk meng-*assign* berat ke setiap term pada dokumen sampel yang diperoleh. Berat yang di-*assign* akan memberikan indikasi tentang seberapa pentingnya suatu istilah dalam suatu dokumen.
4. Bagi masing-masing berat term dengan maksimum berat dari semua term. Sebagai hasilnya nanti, semua berat dikontrol/berada dalam range [1,0]. Dengan kata lain, menormalisasikan masing-masing term dengan berat maksimum.
5. Urutkan term-term tersebut secara ascending sesuai dengan berat masing-masing term. Semakin kurang informatifnya suatu istilah, maka kurang berguna term tersebut dalam suatu dokumen dan akan semakin cocok masuk kedalam daftar stopword.
6. Extrak ranking paling atas X, dimana X adalah sebuah parameter.

Dengan memiliki sebuah panjang array X * Y. masing-masing elemen dalam array berasosiasi dengan sebuah berat.

7. Menyusutkan array dengan merging elemen-elemen mengandung istilah yang sama dan mengambil rata-rata berat dari term. Misalnya : jika term "retrieval" terjadi 3 kali dalam array dan beratnya adalah 0.5, 0.4, 0.3 secara terurut, kita gabungkan 3 elemen ini secara bersamaan ke dalam "one single one" dan berat dari istilah "retrieval" menjadi :
   0.5 + 0.4 + 0.3/3 = 0.4
8. Rank array yang disusutkan secara ascending bergantung berat term. Dengan kata lain urutkan array secara ascending.
9. Extrak L top-ranked terms sebagai daftar stopword. L adalah suatu parameter.

Misalkan dokumen yang digunakan (koleksi dokumen dalam berekstensi .txt)

Doc ID	Doc Text
1	I bought a new mat.
2	A cat sat on the mat.
3	The cat is white and the mat is blue

Koleksi dokumen di-parsing, untuk masing masing kata di konversi menjadi huruf kecil, dilakukan proses *stemming*, dan penghilangan tanda-tanda baca. Hasilnya adalah

idberita	term	freq
1	a	1
1	cat	1
1	sat	1
1	on	1
1	the	1
1	mat	1
2	i	1
2	bought	1
2	a	1
2	new	1
2	mat	1
3	the	2
3	cat	1
3	is	2
3	white	1
3	and	1
3	mat	1
3	blue	1

Ulangi sebanyak Y kali, dimana Y adalah parameter :

1. Secara acak pilih 1 term dari kamus, disebut sebagai ωrandom
2. Ambil semua dokumen yang mengandung ωrandom dalam korpus

Gunakan Kullbak Leibler divergence measure untuk memberikan nilai/bobot untuk masing-masing term yang ada di dokumen yang terambil (Misalkan Y=2) Misalkan ωrandom adalah **cat** (untuk Y=1), maka dokumen-dokumen yang mengandung ωrandom = cat adalah

Doc ID	Doc Text
2	A cat sat on the mat.
3	The cat is white and the mat is blue

Maka bobot untuk masing-masing term yang ada dalam dokumen dengan id = 2 dan 3 yaitu

Doc id	term	bobot
2	a	0.2748318729
2	cat	1.0084952223
2	sat	0.5042476111
2	on	0.5042476111
2	the	1.5127428334
2	mat	0.7400960144
3	is	1.0084952223
3	white	0.5042476111
3	and	0.5042476111
3	blue	0.5042476111

Misalkan ωrandom adalah **and** (untuk Y=2), maka dokumen-dokumen yang mengandung ωrandom = and adalah

Doc ID	Doc Text
3	The cat is white and the mat is blue

Maka bobot untuk masing-masing term yang ada dalam dokumen dengan id = 3 yaitu

Doc Id	term	bobot
3	the	1.0465782492
3	cat	0.4081784312
3	is	1.3710570692
3	white	0.6855285346
3	and	0.6855285346
3	mat	0.2459390212
3	blue	0.6855285346

Hasil akhir dari pengulangan ini adalah array yang berasosasi dengan bobot

Y/X										
1	0.27	1.00	0.50	0.50	1.51	0.74	1.00	0.50	0.50	0.50
2	1.04	0.40	1.37	0.68	0.68	0.245	0.68			

Kemudian dilakukan proses Normalisasi, sehingga nilainya dikontrol antar [0,1]

Y/X										
1	0.18	0.66	0.33	0.33	1	0.48	0.66	0.33	0.33	0.33
2	0.76	0.29	1	0.5	0.5	0.17	0.5			

Lakukan proses sorting

Y/X									
1	0.18	0.33	0.33	0.33	0.33	0.33	0.48	0.66	
2	0.17	0.29	0.5	0.5	0.5	0.76	1		

Ambil X teratas : sehingga hasilnya adalah array berukuran X*Y. misalkan X=6, maka hasilnya adalah

Y/X						
1	0.18	0.33	0.33	0.33	0.33	0.33
2	0.17	0.29	0.5	0.5	0.5	0.76

Lakukan proses merging, dimana setiap term yang sama akan dimerge

0.17	0.18	0.29	0.33	0.33	0.34	0.4	0.4	0.76

Ambil L teratas, misalkan L = 5
Maka term-term yang berasosasi terhadap bobot diatas sebagai daftar *stopword* adalah
Mat, a, cat, sat, on

2.5 Kategorisasi

Kategorisasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk memperkirakan kelas dari suatu objek yang labelnya tidak diketahui.

Pada klasifikasi terdapat dua proses, yaitu:

1. Proses *Learning* atau *Training*
   Proses ini bertujuan untuk membangun model atau fungsi yang akan digunakan untuk memprediksi label atau kelas dari objek yang belum diketahui label atau kelasnya. Pada proses ini digunakan objek

yang telah diketahui label atau kelas datanya yang disebut *training set*. Performansi model dapat diukur dengan menguji model dengan data *training* itu sendiri disebut performansi *training.*

2. Proses *Testing*.
Pada proses ini akan dilakukan tes dengan menggunakan objek yang telah diketahui label atau kelas yang disebut *test set*, dengan tujuan mengetahui keakurasian dan *error-rate* dari model yang telah dibentuk. Objek yang digunakan pada proses ini berbeda dengan objek yang digunakan pada proses learning. Keakurasian objek yang digunakan pada proses tes ini dilakukan dengan cara membandingkan label atau kelas yang telah diketahui dengan label atau kelas hasil prediksi dari model atau fungsi.

2.5.1 Kategorisasi Teks

Klasifikasi teks merupakan salah satu contoh penggunaan metode klasifikasi. Pada klasifikasi teks akan dilakukan pengklasifikasian suatu dokumen, kedalam satu kelas kategori. Proses pengklasifikasian dokumen juga harus melalui proses *training* terlebih dahulu. Sebelum dilakukan *training*, dilakukan *preprocessing* terlebih dahulu. *Preprocessing* adalah proses yang dilakukan pada data *training* agar data tersebut sesuai dengan masukan yang diminta oleh algoritma *learning.* Terdiri dari pembuangan *stopword,* pembobotan kata*,* dan pemilihan atribut, yaitu kata-kata yang paling sering muncul dalam seluruh data *training*, dilakukan dengan memakai metode *feature selection*. Baru kemudian atribut-atribut ini menjadi masukan untuk algoritma *learning* sehingga dapat dihasilkan model *learning* yang dapat mengklasifikasi data.

2.5.2 Evaluasi

Keefektifan untuk klasifikasi teks, didefinisikan sebagai pengukuran kemampuan sistem untuk mengkategorikan dokumen ke dalam kelompok yang sesuai. Model tabel kontingensi adalah pengukuran yang paling banyak digunakan untuk mengukur keefektifan.

2.5.3 Tabel Kontingensi

Didefinisikan sebagai tabel statistik yang memperlihatkan frekuensi yang diobservasi dari elemen data yang diklasifikasi berdasar dua variabel, baris mengindikasikan satu variabel dan kolom mengindikasikan variabel yang satunya. Digunakan dua pengukuran penting untuk keefektifan sistem dari jenis tabel ini, yaitu recall dan precision.

Contoh tabel kontingensi yang sederhana dari sistem pengambilan keputusan biner dapat dilihat pada Tabel 2.1 Sistem membuat n keputusan biner, dimana masing-masing memiliki satu jawaban yang tepat, baik Yes atau No. Setiap jawaban menerangkan jumlah keputusan tiap tipe. Misalnya a adalah jumlah dimana sistem menghasilkan Yes, dan Yes merupakan jawaban yang tepat. Dan c adalah jumlah ketika sistem menjawab Yes, padahal jawaban tepatnya adalah No.

Tabel 2-1 : Tabel kontigensi untuk kumpulan keputusan biner

		PREDICTED CLASS	
		Class = Yes	**Class = No**
ACTUAL CLASS	**Class = Yes**	A	b
	Class = No	C	d

Berikut ini dua pengukuran penting untuk keefektifan yang didapat dari tabel kontingensi:

- ***Recall*** dihitung untuk mengevaluasi seberapa besar *coverage* suatu model dalam memprediksi suatu kelas tertentu. *Recall* didapat dari jumlah kategori yang tepat diklasifikasi dibagi total jumlah kategori benar yang seharusnya :

$$R = \frac{a}{a + b} \quad (2.1)$$

- ***Precision*** dihitung untuk mengevaluasi seberapa baik ketepatan model dapat memprediksi suatu kelas. *Presicion* didapat dari jumlah kategori yang tepat diklasifikasi dibagi jumlah kategori seluruhnya yang harus diklasifikasi :

$$P = \frac{a}{a + c} \quad (2.2)$$

Pengukuran yang efektif dapat melakukan kesalahan apabila pemeriksaan recall dan precision dilakukan sendiri. Precision mungkin dikorbankan untuk mendapat recall yang tinggi. Untuk menyimpulkan dan membuat pengukuran yang composite, digunakan *F-measure* sebagai kriteria evaluasi. *F-Measure* merupakan matriks gabungan kombinasi terbaik antara *recall* dan *presicion* :

$$F\text{-}Measure = \frac{2a}{2a + b} \quad (2.3)$$

2.5.4 Estimasi Akurasi

Ukuran lain yang umumnya digunakan pada mesin learning adalah akurasi (A) dan *error* (E) tidak digunakan secara luas pada klasifikasi teks. Akurasi digunakan untuk menghitung seberapa akurat model yang telah dihasilkan dapat memprediksi data. Sesuai tabel kontingensi pada tabel 2.1, akurasi dan error didefinisikan sebagai berikut :

$$A = \frac{a+d}{a+b+c+d} \tag{2.4}$$

$$E = \frac{b+c}{a+b+c+d} = 1 - A \tag{2.5}$$

3. Perancangan Sistem

3.1 Deskripsi Sistem

Pada bab ini akan dibahas mengenai analisa dan perancangan perangkat lunak untuk pembangunan daftar *stopword* secara otomatis yang akan digunakan untuk proses kategorisasi dokumen. Perancangan implementasinya berupa diagram aliran data (DAD). Diagram aliran data dipakai sebagai alat bantu pengembangan sistem yang memiliki level-level sesuai dengan tingkat kedalamannya. Diagram aliran data dipakai karena mempunyai sifat yang dapat menjamin kejelasan sistem yang digambarkan, kelengkapan penggambaran dan menghindari keambiguan.

3.2 Analisa sistem

Pada tugas akhir ini akan dibangun sebuah perangkat lunak yang akan digunakan untuk pembangunan daftar *stopword* secara otomatis dan akan digunakan juga Weka sebagai *tool* Data Mining untuk proses pengkategorisasian dokumen, guna menguji tingkat keakuratan dari daftar stopword yang dibangun dengan melihat nilai F-measure.

Metode yang digunakan untuk pembangunan daftar *stopword* adalah *Term-Based Random Sampling.* Algoritma *Term-Based Random Sampling* ini didasarkan pada seberapa informatifnya suatu term dalam suatu dokumen. Dengan ini bisa ditentukan tingkat kepentingan dari term-term yang ada dalam dokumen, term dengan tingkat kepentingan yang kecil maka akan cenderung dikategorikan ke dalam daftar *stopword*. Untuk menghitung tingkat kepentingan dari term-term yang ada, maka digunakan *Kullback-Leibler divergence measure* untuk menghitung nilainya.

Pada aplikasi ini akan dilakukan proses stemming terlebih dahulu yaitu pengembalian kata-kata yang ada dalam dokumen kedalam bentuk dasarnya. Kemudian dilakukan pembangunan daftar stopword dengan melihat bobot masing-masing term dan juga memperhatikan nilai thresholdnya.

4. Implementasi dan Pengujian

4.1 Implementasi

4.1.1 Deskripsi Perangkat Lunak

Secara umum sistem yang akan dibangun adalah pemberian bobot pada tiap kata (*term*) untuk semua dokumen teks yang ada dan menemukan kata-kata yang memiliki tingkat kepentingan yang lebih penting atau lebih informatif dalam dokumen dengan melakukan proses *sampling* pada dokumen yang ada, dan kemudian nantinya akan dianalisa berdasarakan bobot yang ada (feature weight) untuk masing-masing kata dalam tiap-tiap sampel yang diperoleh. Dan hasil dari bobot-bobt kata yang diperoleh akan digunakan untuk menentukan kata-kata (terms) yang akan dikategorikan sebagai *stopword* dengan *threshold* tertentu agar memperoleh daftar *stopword* yang optimal. Daftar *stopword* tersebut akan digunakan dalam proses data mining yaitu kategorisasi sebagai langkah *preprocessing* yaitu proses pembuangan *stopword* dan menggunakan *classifier* yang terdapat pada tool WEKA.

Sistem ini dibangun menjadi sebuah tool atau alat bantu untuk menghasilkan daftar *stopword* untuk mem-preproses dataset dalam hal pembuangan *stopword.* Sehingga dataset yang akan di-mining bisa menghasilkan hasil yang lebih baik. Diharapakan sistem ini dapat digunakan untuk menghasilkan daftar stopword yang bisa dipercaya dan cocok dengan dataset yang akan di-mining. Sistem yang dibangun menggunakan PHP dengan *user interface* yang *user friendly* sehingga dari sistem akan dengan mudah mengoperasikan sistem. Data berupa dokumen teks yang diambil dari dokumen Reuters diletakkan dalam direktori tertentu. Setelah dilakukan proses pembobotan maka dataset yang telah terbentuk akan disimpan dala suatu database agar mudah dan fleksibel dalam melakukan analisa dan proses kategorisasi.

Data yang digunakan dalam proses pembangunan daftar *stopword* merupakan data *non-structure* disimpan ke dalam file bertipe teks, sehingga aplikasi yang akan dibangun diharapkan dapat melakukan preprocessing yaitu dengan melakukan segmentasi teks dengan menghitung kemunculan kata dalam tiap dokumen yang

menghasilkan data numerik pada atribut-atributnya. Analisis akan dilakukan terhadap hasil kategorisasi dari data yang telah dihilangkan kata-kata yang sering muncul dari data tersebut dan kata tersebut tidak berguna (*stopword removal*) dengan menghitung keakuratan data yang dihasilkan berupa nilai F-Measure.

4.1.2 Input dan Output

Secara umum input dalam perangkat lunak ini adalah data metah berupa file yang ber-ekstensi txt yang berasal dari website berita berbahasa Inggris. Pengguna akan menginputkan jumlah dokumen yang akan diproses dimana sebelumnya data tersebut didownload berupa dakumen Reuter. Sedangkan yang menjadi output pada perngkat lunak tersebut adalah file database Mysql, berupa kata-kata (*terms*) yang menjadi/dikategorikan ke dalam *stopword* yang sesuai dengan dataset yang diproses yang digunakan nantinya untuk pre-proses pada kategorisasi untuk melihat apakah akuras meningkat atau tidak

4.2 Skenario Pengujian

Pengujian dilakukan untuk melihat seberapa baik daftar *stopword* yang dibangun yang diperoleh dari sistem yang diimplemetasikan. Proses evaluasi dari pengujian dilihat dari nilai F-Measure yang menunjukkan seberapa baiknya proses kategorisasi.

Proses training menggunakan dataset Reuter dengan jumlah total 80 dokumen dan 4 kelas. Sebelumnya dataset Reuter dibobotin terlebih dahulu dengan pembobotan TF/IDF dan masing-masing dokumen telah di stemming menggunakan algoritma Porter. Proses testing menggunakan data sebanyak 20% dari data training yang diambil secara random. Proses training dan testing dilakukan terhadap data sebelum dan sesudah dilakukan proses pembuangan daftar *stopword* menggunakan tool WEKA.

4.2.1 Dokumen Uji

Dataset yang akan diujikan diambil dari dataset UCI yaitu berita berbahasa Inggris Reuter. Untuk dataset reuter ini, terdiri dari 200 dokumen, 20 kelas tapi untuk dokumen uji yang digunakan hanya 4 kelas yang terdiri dari 80 dokumen.

Dataset yang akan digunakan berjumlah 80 data untuk dokumen Reuter. Terdapat beberapa kategori pelabelan dari dataset yang diambil yang dijelaskan dalam tabel sebagai berikut:

Tabel 4-1 Dataset dan kategori dari dokumen Reuter

No	Dataset	Kategori
1	Reuter	Corn, Money, Ship, Trade
2	Reuter	Acq, Crude, Earn, Grain

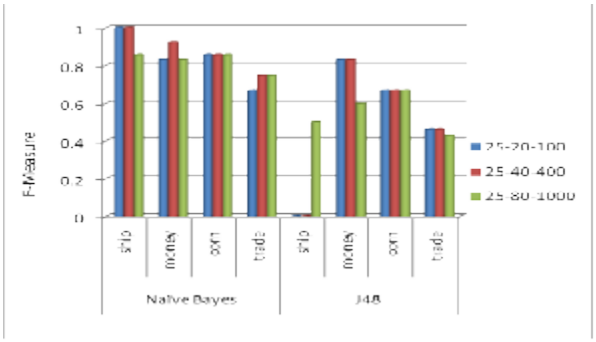
4.2.2 Pengujian Daftar *Stopword*

Dataset yang ada dilakukan pembuangan daftar *stopword* dengan sistem yang ada yaitu dengan daftar *stopword* yang telah didefenisikan Salton And Buckley I dan II, Google *Stopword*, default English *Stopword* dan daftar *stopword* hasil dari sitem yang dibangun dan juga dataset yang tidak dilakukan pembuangan daftar *stopword*.

Pengujian dataset yang telah ada dimaksudkan untuk membandingkan seberapa baik daftar *stopword* yang dihasilkan dari sistem yang dibangun. Proses untuk evaluasi seberapa baik daftar *stopword* yang dihasilkan sistem dilakukan dengan pendekatan datamining yaitu proses kategorisasi. Algoritma kategorisasi yang digunakan telah ada dalam tool WEKA. Proses pengukuran akan memperhatikan hasil akurasi dan F-measure yang dihasilkan.

4.3 Analisis

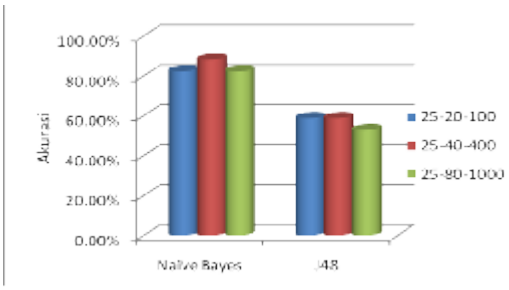
4.3.1 Analisis F-Measure dan Akurasi



Gambar 4-1 Distribusi nilai F-Measure dengan Naïve Bayes dan J48 menggunakan stopword Term-Based Random Sampling dengan parameter Y:X:L 25-20-100, 25-40-400, 25-80-1000

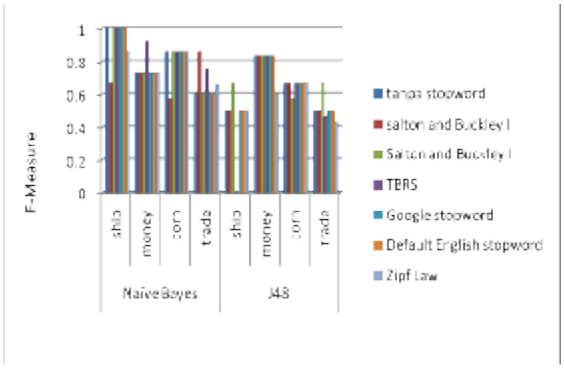
Berdasarkan distribusi nilai F-Measure Gambar 4-1, dapat diketahui bahwa persebaran nilai F-Measure untuk masing-masing kelas hampir merata untuk kategorisasi menggunakan Naïve Bayes Sedangkan untuk kategorisasi yang menggunakan J48 ada beberapa nilai F-Measure yang secara signifikan menunjukkan perbedaan yang jauh yaitu

pada kelas ship. Untuk kelas corn, untuk masing-masing dataset yang menggunakan daftar stopword yang berbeda menghasilkan nilai F-Measure yang sama persis yaitu pada nilai 0.857 untuk yang menggunakan Naïve Bayes sedangkan yang menggunakan J48 dengan nilai F-Measure 0.667. Nilai F-Measure paling tinggi itu ada pada kelas ship yaitu dataset yang menggunakan daftar *stopword* dengan parameter 25-20-100 dan 25-40-400 pada kategorisasi menggunakan Naïve Bayes tapi berbeda dengan kategorisasi yang menggunakan J48, dimana menunjukkan nilai F-Measure yang sangat-sangat kecil. Sedangkan untuk kelas money, nilai F-Measure tertinggi ada pada dataset yang menggunakan daftar *stopword* dengan parameter 25-40-400 yaitu pada nilai F-Measure 0.923 untuk kategorisasi menggunakan Naïve Bayes. Untuk kelas trade, nilai F-Measure antara dataset yang menggunakan daftar *stopword* dengan parameter 25-80-1000 sama dengan yang berparameter yaitu 0.75. Tapi secara keseluruhan nilai F-Measure yang paling konstan dan baik yaitu pada dataset yang preproses penghilangan daftar *stopword*-nya menggunakan daftar *stopword* yang dihasilkan sistem dengan parameter 25-40-400.



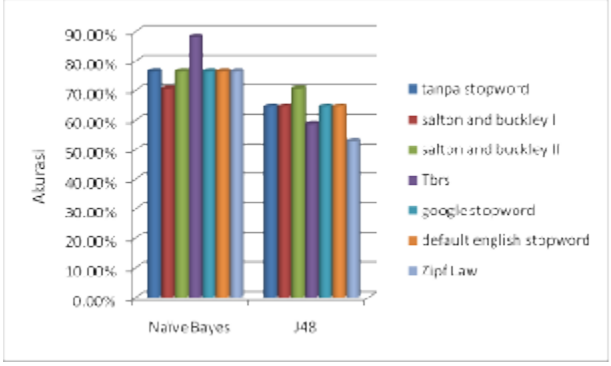
Gambar 4-2 Distribusi nilai Akurasi Reuter dengan metode kategorisasi Naïve Bayes dan J48 menggunakan stopword Term-Based Random Sampling dengan parameter Y:X:L 25-20-100, 25-40-400, 25-80-1000

Berdasarkan Gambar 4-2 yaitu grafik yang menunjukkan perbandingan akurasi dari masing-masing hasil kategorisasi dataset dengan daftar *stopword* yang berbeda-beda. Untuk dataset yang menggunakan daftar *stopword* dengan perbandingan parameter Y:X:L yaitu 25-20-100 dan 25-80-1000 memiliki hasil akurasi kategorisasi yaitu 82.35% dengan menggunakan metode kategorisasi Naïve Bayes dan untuk kategorisasi yang menggunakan metode J48 menghasilkan nilai akurasi 58.82% dan 52,94%. Sedangkan nilai akurasi tertinggi yaitu 88.24% untuk Naïve Bayes dihasilkan dari hasil pengkategorisasian dengan daftar *stopword* dengan perbandingan parameter Y:X:L yaitu 25-40-400 dan untuk akurasi terbaik yang menggunakan metode J48 yaitu 58.82% yaitu pada perbandingan parameter 25-20-100 dan 25-40-400



Gambar 4-5 Distribusi nilai F-Measure koleksi dokumen Reuter dengan metode kategorisasi Naïve Bayes dan J48 menggunakan daftar terdefenisi, stopword yang dihasilkan dengan menggunakan Zipf Law, hasil Term-Based Random Sampling dan tanpa stopword

Berdasarkan Gambar 4-5, yang menunjukkan distribusi F-Measure hasil kategorisasi dari dokumen Reuter dengan daftar *stopword* yang terdefenisi, Term-Based Random Sampling Zipf Law, dan tanpa *stopword* , untuk pengkategorian dengan metode Naïve Bayes pada kelas ship menunjukkan nilai F-Measure yang tinggi dan hanpir merata. Untuk daftar *stopword* Salton and Bucley I menghasilkan akurasi yang paling rendah dalam kelas ship yaitu 0.667 kemudian Zipf Law dengan nilai F-Measure 0.857. Untuk kelas money hampir merata yaitu untuk daftar *stopword* salton and Bucley I dan II, Google stopword, default English stopwrd dan tanpa stopword menghasilkan nilai F-Measure yang sama dengan nilai 0.727. Sedangkan daftar *stopword* TBRS dengan nilai F-Measure 0.923. Untuk kelas corn F-Measure terendah juga dihasilkan dari penggunaan daftar stopword Salton and Bucley I dan untuk daftar stopword yang lain menghasilkan nilai F-Measure yang sama yaitu 0.857. Untuk kelas trade F-Measure terbaik dihasilkan dari penggunaan daftar stopword Salton and Bucley I yaitu 0.857. Sedangkan untuk kategorisasi dengan metode J48, untuk kelas money nilai F-Measurenya sama yaitu 0.833 kecuali untuk daftar *stopword* Zipf Law dengan F-Measure 0.6. Untuk kelas ship dan trade, nilai F-Measure tertinggi yaitu penggunaan daftar *stopword* salton and Bucley II yaitu 0.667.



Gambar 4-6 Distribusi nilaii kategorisasi menggunakan Naïve Bayes dan J48 menggunakan daftar terdefenisi, stopword yang dihasilkan dengan menggunakan Zipf Law, hasil Term-Based Random Sampling dan tanpa stopword

Berdasarkan Gambar 4-6, grafik yang menunjukkan akurasi dari proses pengkategorisasian yang menggunakan daftar terdefenisi, stopword yang dihasilkan dengan menggunakan Zipf Law, hasil Term-Based Random Sampling dan tanpa stopword. Bisa diambil kesimpulan, untuk pengkategorisasian dengan menggunakan Naïve Bayes menghasilkan akurasi terbaik yaitu 88.24% dan juga terbaik jika deibandingkan dengan akurasi yang menggunakan metode kategorisasi J48. Tapi untuk akurasi tertinggi untuk pengkategorian dengan menggunakan J48 yaitu pengkategorian yang menggunakan daftar stopwr Salton and Buckley I. Dan untutk pengkategorian yang menggunakan Naïve Bayes, akurasinya cenderung merata dengan akurasi yang cukup efektif. Begitu juga pengkategorian yang menggunakan J48, hanya saja pada saat menggunakan daftar stopword Zipf Law, nilai akurasinya yaitu 52.94%.

Lampiran

[1] C. Fox. Lexical analysis and stoplists. In Information Retrieval - Data Structures & Algorithms, pages 102{130. Prentice-Hall, 1992.

[2] D. Hawking. Overview of the TREC2002. In Proceedings of the Nineth Text REtrieval Conference (TREC 9), pages 87-94, Gaithersburg, MD, 2000.

[3] E. M. Voorhees. Overview of TREC2002. In Proceedings of the Eleventh Text REtrieval conference (TREC2002), pages 1{16, Gaithersburg, MD, 2002.

[4] G. Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems (TOIS), 20(4):357{389, 2002.

[5] Jiawei, Han. and Michelle, Kamber.(2001). *Data Mining: Concepts and Techniques*. San Francisco : Morgan Kaufmann Publisher.

[6] H. Zipf. Human Behaviours and the Principle of Least Effort. Addison-Wesley, Cambridge, MA, 1949.

[7] Effective Technique for Indonesian Text Retrieval. Jelita Asian B.comp. Sc.(Hons), School of Computer Science and Information Technology, Science, engineering, and Technology Portfolio, RMIT University, Melbourne, Victoria, Australia. 2007.

[8] Http://www.coffeecup.com/help/articles/default-english-stopwords/?PHPSESSID=188516aae0ea25b13fe018e2faf238ce

[9] Http://www.ranks.nl/resources/stopwords.html

[10] Rachel TszWai Lo, Ben He, Iadh Ounis. Automatically Building a Stopword List for an Information Retrieval System. Department of Computing Science University of Glasgow 17 Lilybank Gardens Glasgow, UK

[11] R. K. Belew. Finding Out About. Cambridge University Press, 2000.

[12] S. Charkrabarti. Mining the Web: Discovering knowledge from hypertext. Morgan Kaufmann, 2003.

[13] T. M. Cover and J. A. Thomas. Elements of Information Theory. John Wiley, 1991.

[14] W. B. Croft. Combining approaches to information retrieval. In Advances in Information Retrieval - Recent Research from the Center for Intelligent Information, pages 1{28. Kluwer Academic Publishers, 2000.

[15] W. Francis. Frequency Analysis of English Usage: Lexicon and Grammar. Houghton Mi²in, 1982.