Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TINJAUAN PUSTAKA

2.1 *Text Mining*

Teks di ciptakan bukan untuk di gunakan oleh mesin, tapi untuk dikonsumsi manusia langsung. Karena itu, pada umumnya "*Natural Language Processor*" digunakan untuk memproses teks yang tidak terstruktur. Hearst (1999) mempertanyakan penggunaan kata '*mining*' di *data mining* dan *text mining*. Kata '*mining*' memberikan arti dimana fakta-fakta atau relasi-relasi baru dihasilkan dari proses me-'mining' data. Dia mengklaim bahwa aktivitas data mining lebih memfokuskan pada penemuan *trend* dan *pattern* yang sebenarnya sudah ada (Hearst, 1999). Sedangkan ahli *text mining* yang lain beranggapan bahwa *text mining* adalah proses penemuan kembali relasi dan fakta yang terkubur didalam teks, dan tidak harus baru (Mladenic et al., 2001).

Seperti halnya *data mining*, *text mining* adalah proses penemuan akan informasi atau tren baru yang sebelumnya tidak terungkap, dengan memproses dan menganalisa data dalam jumlah besar (Adiwijaya, 2006). Dalam menganalisa sebagian atau keseluruhan teks yang tidak terstruktur (*unstructured text*), *text mining* mencoba untuk mengasosiasikan satu bagian teks dengan yang lainnya berdasarkan aturan-aturan tertentu. Hasil yang di harapkan adalah informasi baru atau "*insight*" yang tidak terungkap jelas sebelumnya. Seperti di sebutkan sebelumnya yang sedikit condong pada definisi *text mining* oleh Hearst, *text mining* telah mengadopsi teknik yang di gunakan di bidang *natural language processing* dan *computational linguistics*.

Implementasi Algoritma ..., Andre Anggiharto, FTI UMN, 2013

Walaupun teknik di *computational linguistics* bisa dibilang maju dan cukup akurat untuk mengekstrak informasi, akan tetapi tujuan *text mining* bukan hanya mengekstrak informasi. Melainkan untuk menemukan *pattern* dan informasi baru yang belum terungkapkan (Craten et al., 1998), yang sulit ditemukan tanpa analisa yang dalam. Walau kemampuan komputer untuk mencapai kemampuan untuk memproses teks seperti manusia sangat sulit, bila tidak mustahil, telah banyak teknik-teknik baru di *computational linguistics* yang bisa membantu *text mining* untuk mencerna teks lebih jauh lagi. *Text mining* lebih memfokuskan pada relasi dan *co-existence* dari satu dokumen dengan yang lainnya. Walaupun *text mining* lebih dari *information retrieval, text mining* telah mengadopsi *information retrieval* untuk menyaring dan mengurangi jumlah informasi untuk diproses selanjutnya.

Text mining memiliki definisi lain yaitu 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 antara dokumen (Mooney, 2006) dan proses untuk menemukan informasi baru, yang belum pernah diketahui, secara otomatis oleh komputer dari sumber-sumber tertulis yang berbeda (Weiguo et al., 2005).

Istilah informasi yang belum pernah diketahui sebelumnya ada 2 macam, yaitu (Weiguo et al., 2005).:

a. *Strict definition*, yaitu informasi yang belum pernah diketahui sebelumnya, bahkan oleh penulisnya sekalipun. Contohnya adalah menemukan metode baru

untuk pertumbuhan rambut, yang merupakan efek samping dari prosedur lain yang ada

b. *Lenient definition*, yaitu menemukan informasi yang sudah ada di teks. Contohnya menemukan nama suatu produk dari halaman web.

Banyak juga ahli riset yang mengkategorikan *document categorization* sebagai *text mining*. Walau kategorisasi dokumen dapat memberikan label dan kesimpulan yang akurat pada dokumen-dokumen tertentu, ini tidak menghasilkan fakta-fakta atau relasi yang baru. Tetapi bilamana label-label atau kesimpulan-kesimpulan yang di hasilkan di analisa dan di korelasikan lebih lanjut, ini bisa menghasilkan fakta dan relasi baru antara *group-group* dokumen yang berbeda. Tahapan dalam proses *text mining* terdiri dari *tokenizing*, *filtering*, *stemming* (Mooney, 2006).

2.2 *Tokenizing*

Tokenizing adalah tahap pemotongan *string* input berdasarkan tiap kata yang menyusunnya (Mooney, 2006). Pengertian lain t*okenizing* adalah sebuah proses untuk memilah isi dokumen teks sehingga menjadi satuan kata-kata. Menurut (Weiss *et al.* 2005), proses ini cukup rumit untuk sebuah program komputer karena beberapa karakter dapat dijadikan sebagai pembatas (*delimiter)* dari token-token itu sendiri. Pembatas dari token tersebut antara lain spasi, tab, dan baris baru, sedangkan karakter ( ) < > ! ? ” . , terkadang dianggap sebagai pembatas dan juga bukan pembatas tergantung pada kondisi pemakaiannya.

2.3 *Filtering*

Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil *token*. Biasanya dilakukan dengan cara menggunakan *wordlist* (menyimpan kata penting) atau *stoplist* (membuang kata yang kurang penting) dari hasil *token* yang telah diperoleh dari proses *tokenizing* (Mooney, 2006).

Menurut Talla (2002, p21), *wordlist* adalah daftar kata-kata yang tidak dipakai didalam pemrosesan bahasa alami. Hasil penelitian sebelumnya menyatakan bahwa penggunaan *wordlist* meningkatkan kemampuan pemrosesan bahasa alami.

Menurut Soumen (2003,p48), kebanyakan bahasa resmi di berbagai Negara memiliki kata fungsi dan kata sambung seperti artikel dan preposisi yang hampir selalu muncul pada dokumen teks. Biasanya kata-kata ini tidak memiliki arti yang lebih didalam memenuhi kebutuhan seorang pengguna di dalam mencari informasi. Kata-kata tersebut disebut sebagai wordlist.

2.4 *Stemming*

Stemming adalah tahap mencari *root* kata dari tiap kata hasil *filtering* (Mooney, 2006). Menurut Talla (2003, p7), *Stemming* merupakan suatu proses untuk menemukan kata dasar dari sebuah kata. Dengan menghilangkan semua imbuhan (*affixes)* baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi bahasa yang baik dan benar.

2.5 *Term Frequency* (TF)

Langkah pertama untuk menentukan identifikasi dokumen adalah membuat daftar kata-kata dalam suatu dokumen kemudian menghitung frekuensi kemunculannya. Cara ini pertama kali dilakukan oleh Hans Peter Luhn (1958) dan disebut *term frequency* (tf). *Term frequency* (tf) dapat digunakan untuk mengukur *term-weighting* (pembobotan kata) yang didasarkan pada jumlah frekuensi kata dalam sebuah dokumen.

2.6 *Inverse Document Frequency* (IDF)

IDF adalah besaran yang pertama kali didefinisikan oleh Karen Spark-Jones (1998) yaitu *log* dari jumlah dokumen keseluruhan dibagi dengan jumlah dokumen yang ada *term*/kata yang dicari. Jadi *term* yang umum yang terdapat dalam semua dokumen akan mempunyai nilai IDF yang rendah dan sebaliknya term yang hanya terdapat dalam satu dokumen akan mempunyai nilai IDF tinggi. Faktor *Term frequency* saja belum cukup memberikan indikasi bahwa *term* yang dihasilkan memiliki kedudukan yang sesuai dalam sebuah dokumen atau teks *query*. Hal ini dapat dilihat ketika terdapat *term* yang memiliki frekuensi yang tinggi terkonsentrasi pada sebuah atau sebagain kecil dokumen, tidak terdapat pada sebagian dokumen lainnya, maka akan berpengaruh pada ketepatan hasil pencarian. Oleh karena itu *inverse document frequency* (idf) memperhitungkan faktor-faktor yang menyangkut penyebaran suatu *term* dalam sekumpulan dokumen,didefinisikan dengan rumus yaitu

$$IDF = \log\left(\frac{D}{df}\right)$$

Keterangan :

D = total dokumen

df = banyak dokumen yang mengandung kata yang dicari

2.7 *Term Frequency – Inverse Document Frequency* (TF-IDF)

Pembedaan kata (*term discrimination*) dalam sebuah dokumen harus mampu mengidentifikasikan dengan baik isi dari dokumen tersebut terhadap dokumen lainnya. Sehingga sebuah kata akan memiliki bobot hubungan tinggi bila kata tersebut memiliki *term frequency* yang tinggi dalam dokumen tetapi frekuensi yang rendah pada koleksi dokumen secara keseluruhan. Oleh karena itu, perhitungan bobot kata dilakukan dengan mengalikan *term frequency* pada suatu dokumen dengan *inverse document frequency* (Gerard et al., 1998), dimana didefiniskan dengan rumus yaitu

$$W_{dt} = tf_{dt} * IDF_t$$

Keterangan :

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

tf = banyaknya kata yang dicari pada sebuah dokumen

IDF = *Inverse Document Frequency*

Metode tf-idf merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen (Sulistyo et al., 2008). Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata didalam sebuah dokumen dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata didalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut didalam dokumen tersebut. Jumlah frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi didalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen

2.8 *Vector Space Model*

Vector space model adalah suatu model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query*. Pada model ini, *query* dan dokumen dianggap sebagai vektor-vektor pada ruang n-dimensi, dimana n adalah jumlah dari seluruh *term* yang ada dalam *leksikon. Leksikon* adalah daftar semua *term* yang ada dalam indeks. Salah satu cara untuk mengatasi hal tersebut dalam model *vector space* adalah dengan cara melakukan perluasan vektor. Proses perluasan dapat dilakukan pada vektor *query*, vektor dokumen, atau pada kedua vektor tersebut, dimana didefinisikan dengan rumus yaitu

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Keterangan :

A = Nilai bobot kata pada kata kunci

B = Nilai bobot kata pada dokumen

2.9 PHP

PHP adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memrogram situs *web* dinamis. PHP dapat digunakan untuk membangun sebuah CMS. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998,

perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang *PHP: Hypertext Preprocessing*. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2.10 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang

mendirikan MySQL AB adalah: David Axmark,Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL(General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan

sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

2.11 *Library of Congress*

Perpustakaan Kongres Amerika Serikat (*Library of Congress*) adalah perpustakaan nasional Amerika Serikat dan pusat riset Kongres Amerika Serikat. Perpustakaan ini menempati 3 buah gedung di Washington, D.C.. Perpustakaan terbesar di dunia dari segi luas rak buku dan total koleksi buku. Katalog perpustakaan ini mendaftar lebih dari 32 juta judul bahan pustaka yang ditulis dalam 470 bahasa. Perpustakaan juga menyimpan koleksi 61 juta manuskrip, dan koleksi buku langka terbesar di Amerika Utara, termasuk naskah Deklarasi Kemerdekaan Amerika Serikat dan Kitab Gutenberg (satu dari 4 salinan belum dalam keadaan sempurna yang ada).

Selain itu, perpustakaan menyimpan lebih dari 1 juta judul terbitan pemerintah Amerika Serikat, 1 juta terbitan surat kabar dari seluruh dunia selama 3 abad terakhir, 33.000 volume surat kabar yang dijilid, 500.000 gulung mikrofilm, lebih dari 6.000 judul buku komik, dan koleksi literatur hukum terbesar di dunia. Koleksi bahan nonbuku terdiri dari film, 4,8 juta judul peta, lembar musik, 2,7 juta judul rekaman suara, lebih dari 13,7 juta lembar foto (termasuk gambar arsitektur), serta biola Betts Stradivarius dan Cassavetti Stradivarius.

2.12 *Library of Congress Classification*

Klasifikasi kongres perpustakaan (LCC) adalah sebuah sistem klasifikasi perpustakaan yang dikembangkan oleh Perpustakaan Kongres Amerika Serikat . Hal ini digunakan oleh sebagian besar penelitian dan perpustakaan akademis di Amerika Serikat dan beberapa negara lain, termasuk Taiwan, ROC. Klasifikasi dibedakan dari nomor panggilan untuk salinan buku tertentu dalam koleksi, seperti "PZ7.J684 Wj 1982 FT MEADE Copy 1" di mana klasifikasi adalah "PZ7.J684 Wj 1982".

Klasifikasi ini diciptakan oleh Herbert Putnam pada tahun 1897, tepat sebelum ia diangkat sebagai kepustakawanan Kongres. Dengan saran dari Charles Ammi Cutter, yaitu dipengaruhi oleh Klasifikasi *Cutter Expansive* (dikembangkan di tahun 1880-an) dan oleh DDC, Dewey (dari 1876). Ini dirancang khusus untuk tujuan dan koleksi Perpustakaan Kongres untuk menggantikan sistem lokasi tetap yang dikembangkan oleh Thomas Jefferson.

Pengklasifikasian dari *Library of Congress Classification* terdiri dari :

Class A – General Works

Class B – Philosophy, Psychology, Religion

Class C – Auxiliary Sciences History

Class D – World History and History of Europe, Asia, Africa, Australia, New Zealand, etc.

Class E & F – History of The Americas

Class G – Geography, Anthropology, Recreation

Class H – Social Sciences

Class J – Political Science

Class K – Law

Class L – Education

Class M – Music and Books On Music

Class N – Fine Arts

Class P – Language and Literature

Class Q – Science

Class R – Medicine

Class S – Agriculture

Class T – Technology

Class U – Military Science

Class V – Naval Science

Class Z – Bibliography. Library Science. Information Resources (General)