IMPLEMENTASI TEXT MINING untuk KATEGORISASI DOKUMEN (STUDI KASUS DIGILIB.POLIBATAM.AC.ID)

TUGAS AKHIR

Oleh :

Rudy Susanto 3310911009

Disusun untuk memenuhi syarat kelulusan matakuliah Tugas Akhir

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

BATAM

2012

LEMBAR PENGESAHAN

Batam, 19 Juli 2012

Pembimbing,

Mir'atul Khusna Mufida, S.ST

NIK. 109057

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310911009

Nama : Rudy Susanto

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa proyek akhir dengan judul:

IMPLEMENTASI TEXT MINING untuk KATEGORISASI DOKUMEN

(STUDI KASUS DIGILIB.POLIBATAM.AC.ID)

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 19 Juli 2012

Rudy Susanto
3310911009

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, berkat rahmat dan karunianya, yang telah memberikan keyakinan dan semangat sehingga penulis dapat menyelesaikan penyusunan laporan Tugas Akhir ini dengan judul "IMPLEMENTASI TEXT MINING untuk KATEGORISASI DOKUMEN (STUDI KASUS DIGILIB.POLIBATAM.AC.ID)".

Laporan Tugas Akhir ini disusun guna untuk memenuhi salah satu persyaratan untuk menyelesaikan program studi DIPLOMA III Teknik Informatika Politeknik Negeri Batam.

Dalam penyelesaian laporan akhir ini, penulis dengan penuh kesungguhan dan kerja keras sehingga banyak mendapat bantuan dan bimbingan serta masukan dari berbagai pihak. Oleh karena itu, pada kesempatan ini dengan tulus dan rendah hati, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Uuf Brajawidagda, ST. MT, selaku Kaprodi Teknik Informatika.
2. Ibu Mir'atul Khusna Mufida, S.ST, selaku dosen pembimbing.
3. Orang Tua, keluarga dan orang-orang terdekat yang selalu memberikan dukungan baik moril maupun materil.

Dengan segala kemampuan dan keterbatasan yang ada, penulis menyadari sepenuhnya bahwa masih terdapat banyak kekurangan dalam laporan Tugas Akhir ini. Oleh karena itu, saran dan kritik sangat penulis harapkan demi meningkatkan kualitas laporan ini. Akhir kata penulis berharap semoga laporan Tugas Akhir ini dapat memberikan manfaat bagi semua pihak yang berkepentingan.

Batam, 19 Juli 2012

Penulis,

ABSTRAK

IMPLEMENTASI TEXT MINING untuk KATEGORISASI DOKUMEN

(STUDI KASUS DIGILIB.POLIBATAM.AC.ID)

Dengan berkembangnya teknologi media penyimpanan perangkat keras, data yang disimpan semakin banyak dan besar. Namun data tersebut jarang untuk dilihat kembali karena proses waktu yang digunakan relatif lama dan tidak efisien. Sehingga muncul ilmu penambangan data yang disebut dengan *Data Mining*. *Text Mining* merupakan ilmu turunan dari *Data Mining* yang penambangan data dilakukan pada data yang berupa teks.

Tugas akhir ini merancang suatu Sistem yang dapat melakukan penyaringan terhadap isi dokumen dengan ilmu penambangan data khususnya pada teks yang pada akhirnya dapat menemukan kata kunci terbanyak pada sebuah dokumen sehingga dapat melakukan kategorisasi terhadap dokumen tersebut. Sistem ini juga mampu melakukan pencarian terhadap kata kunci untuk menemukan dokumen yang diinginkan. Studi kasus yang dilakukan di dalam Tugas Akhir ini adalah terhadap *digital library* yang terdapat pada Politeknik Negeri Batam yaitu : digilib.polibatam.ac.id yang memuat isi-isi karya Tugas Akhir dari mahasiswa. Perancangan Sistem ini menggunakan PHP berbasis web dan menggunakan MySql sebagai RDBMSnya.

Dengan adanya implementasi *Text Mining* ini dapat membantu dalam pencarian terhadap isi dokumen Tugas Akhir Mahasiswa pada digilib.polibatam.ac.id dan melakukan kategorisasi dokumen dengan tepat.

Kata Kunci : Text Mining, Kategorisasi, Dokumen

ABSTRACT

IMPLEMENTASI TEXT MINING untuk KATEGORISASI DOKUMEN

(STUDI KASUS DIGILIB.POLIBATAM.AC.ID)

Due to the development of hardware storage media, the data that has been saved has become more and more extensive. However, the data is sparse to be reviewed caused of the long time usage and inefficieny that's why caused the emerging called as data mining. Text mining is the science derived from the Data Mining which performed on the data in the form of text.

This Final Project is to design a system that can filterize a document which will be processed with the implementation of text mining, and the process is to find the keywords in a document to help categorize the document. This System also can do the search by input the keyword to find the document that they need. This final project is study case towards digital library of Politeknik Negeri Batam "digilib.polibatam.ac.id" where the library contains mostly the students final project. This System design is using the PHP web based and using MySql as the database.

As the implementation of the Text Mining can help the search for the content of final project document on digilib.polibatam.ac.id and perform the categorization of the document appropriately.

Key words: Text Mining, Categorize, Document

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Bab I Pendahuluan

I.1 Latar Belakang

Pengumpulan data yang didukung oleh perangkat keras yang semakin besar kapasitasnya memungkinkan penyimpanan data dalam skala yang besar juga, namun dari data yang telah terkumpul biasanya jarang dilihat kembali karena data yang telah terkumpul isinya terlalu panjang, membosankan dan tidak menarik. Analisis terhadap data-data tersebut juga tidak memungkinkan dilakukan secara manual karena proses waktu yang relatif lama, sehingga gagasan untuk pemrosesan dari data yang banyak untuk menemukan data-data tersembunyi yang memungkinkan untuk menjadi sebuah informasi yang penting dan berguna bagi pemilik data tersebut dikenal dengan ilmu penambangan data (*Data Mining*).

Dalam ilmu penambangan data (*Data Mining*) terbagi atas kategori data-data yang telah dikumpulkan, seperti data teks, data *spatiotemporal*, data multimedia, data *streams*, dan sebagainya. Penambangan teks (*Text Mining*) muncul dan berkembang karena adanya sebuah kebutuhan untuk memproses data tak terstruktur (*Unstructured data*) dalam bentuk teks. Penambangan teks (*Text Mining*) merupakan turunan dari ilmu penambangan data (*Data Mining*) yang lebih spesifik terhadap teks dan memiliki tujuan dan menggunakan proses yang sama di dalam menerapkan penambangan teks.

Penambangan teks (*Text Mining*) adalah proses pemisahan pola berupa informasi dan pengetahuan yang berguna dari sejumlah sumber data teks, seperti dokumen Word, PDF, kutipan teks, dan lain-lain. Proses yang umum dilakukan oleh penambangan teks diantaranya adalah perangkuman otomatis, kategorisasi dokumen, penggugusan teks, dan lain-lain. Sehingga penambangan teks sangat dibutuhkan untuk meminimalisasi pencarian sebuah teks maupun kalimat yang sesuai untuk kebutuhan dokumentasi dan pengarsipan secara sistem otomatisasi dan pencarian dilakukan dengan menggunakan komputer.

Penerapan penambangan teks (*Text Mining*) sebenarnya telah umum dipergunakan pada instansi-instansi seperti dunia pendidikan, media informasi dan lain sebagainya yang membutuhkan kategorisasi pada pengarsipan dokumen dalam jumlah yang besar. Pengembangan sebuah perpustakaan digital (*Digital Library)* sudah dilaksanakan oleh Politeknik Negeri Batam dalam bentuk situs (*website)*, tetapi karena belum maksimal penerapannya, maka pencarian menggunakan kata kunci yang mampu memilah kategori dokumen tidak ada. Sehingga muncul ide untuk menerapkan fitur tambahan dalam bentuk kategorisasi dokumen pada situs *digilib.polibatam.ac.id* untuk memudahkan pencarian dokumen atau materi pendidikan oleh khalayak umum.

I.2 Rumusan Masalah

1. Bagaimana menyaring dokumen secara efektif.

2. Bagaimana menentukan kategorisasi data dengan akurat.

3. Implementasi Text Mining pada situs *digilib.polibatam.ac.id.*

I.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah tidak melayani kategorisasi type Audio, Video, dan gambar (*Image*).

I.4 Tujuan

Tujuan dari penelitian ini antara lain:

1. Membuat sistem yang dapat memfilter sebuah dokumen secara otomatis.

2. Membuat sistem yang mampu mengkategorisasikan data ataupun dokumen berdasarkan kata kunci (*keyword*).

3. Membuat sistem yang mampu membantu fitur-fitur pencarian terhadap kategorisasi dokumen pada situs *digilib.polibatam.ac.id*.

I.5 Sistematika Penulisan

Laporan ini terdiri dari 6 bab, yaitu Bab Pendahuluan, Tinjauan Pustaka, Analisis, Perancangan, Hasil dan Pembahasan, Kesimpulan dan Saran serta Lampiran yang berhubungan dengan aplikasi yang dibuat.

BAB I

Pendahuluan yang menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan.

BAB II

Tinjauan pustaka yang berisi mengenai pengertian terhadap penambangan teks (*Text Mining*), bahasa pemrograman java, aplikasi Netbeans IDE.

BAB III

Analisis dan Perancangan yang terdiri dari deskripsi umum sistem, fitur utama perangkat lunak, kebutuhan fungsional, kebutuhan non fungsional, *use case*, analisis kelas, diagram interaksi, diagram kelas, rancangan kelas rinci dan perancangan antarmuka.

BAB IV

Dalam bab ini membahas mengenai implementasi kelas dan implementasi antar muka.

BAB V

Kesimpulan dan saran yang berisi tentang kesimpulan dari hasil pembangunan aplikasi yang dibuat pada Tugas Akhir serta saran pengembangan aplikasi mengenai penyempurnaan dan ide yang dapat dilakukan terhadap aplikasi yang dibuat.

Bab II Tinjauan Pustaka

II.1 Penambangan Teks (*Text Mining*)

Salah satu bagian dari *Data Mining* yang cukup menarik adalah *Text Mining*. Metode ini digunakan untuk menggali informasi dari data-data dalam bentuk teks seperti buku, makalah, paper, dan lain sebagainya. Yang membedakan *data mining* dengan *text mining* adalah proses analisis terhadap suatu data. *Data Mining* atau KDD (*Knowledge Discovery in Databases*) adalah proses untuk menemukan pengetahuan dari sejumlah besar data yang disimpan baik di dalam *databases*, data *warehouses* atau tempat penyimpanan informasi lainnya. Sedangkan untuk *text mining* sering disebut dengan *Keyword-Based Association Analysis*. *Keyword-Bases Association Analysis* merupakan sebuah analisa yang mengumpulkan *keywords* atau *terms* (istilah) yang sering muncul secara bersamaan dan kemudian menemukan hubungan asosiasi dan korelasi di antara *keywords* atau *terms* itu.

Penambangan teks (*Text Mining*) adalah suatu proses semiotomatis yang bertujuan untuk menemukan informasi atau pengetahuan yang berguna yang sebelumnya masih belum terungkap, dengan memproses dan menganalisa data yang tak terstruktur (*unstructured data*) dalam jumlah yang besar. Dalam menganalisa sebagian atau keseluruhan teks yang tak terstruktur (*unstructured text)*, penambangan teks (*text mining*) mencoba mengasosiasikan suatu bagian teks dengan yang lainnya berdasarkan aturan dan parameter tertentu. Selain itu, penambangan teks (*text mining*) juga dapat diartikan sebagai kegiatan menambang data dari data yang berupa teks atau dokumen, dengan tujuan mencari kata-kata yang dapat mewakili apa yang ada dalam dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Beberapa area atau bidang penerapan penambangan teks yang umum antara lain:

1. Ekstraksi informasi (*information extraction*): Identifikasi frasa kunci dan keterkaitan di dalam teks dengan melihat urutan tertentu melalui pencocokan pola.

2. Pelacakan topik (*topic tracking*): Penentuan dokumen lain yang menarik seorang pengguna berdasarkan profil dan dokumen yang dilihat pengguna tersebut.

3. Perangkuman (*summarization*): Pembuatan rangkuman dokumen untuk mengefisienkan proses membaca.

4. Kategorisasi (*categorization*): Penentuan tema utama suatu teks dan pengelompokan teks berdasarkan tema tersebut ke dalam kategori yang telah ditentukan.

5. Penggugusan (*clustering*): Pengelompokan dokumen yang serupa tanpa penentuan kategori sebelumnya (berbeda dengan kategorisasi di atas).

6. Penautan konsep (*concept linking*): Penautan dokumen terkait dengan identifikasi konsep yang dimiliki bersama sehingga membantu pengguna menemukan informasi yang mungkin tidak akan ditemukan dengan hanya menggunakan metode pencarian tradisional.

7. Penjawaban pertanyaan (*question answering*): Pemberian jawaban terbaik terhadap suatu pertanyaan dengan pencocokan pola berdasarkan pengetahuan.

II.2 Pemrosesan bahasa alami (*Natural Language Processing*, NLP)

Penambangan teks juga memiliki ketergantungan erat dengan bidang pemrosesan bahasa alami (*natural language processing*) karena masukan yang diolahnya adalah teks dalam bentuk bahasa alami. Pemrosesan bahasa alami (NLP) adalah penerapan ilmu komputer, khususnya kecerdasan buatan (*artificial intelligence*), dan linguistik, khususnya linguistik komputasional (*computional linguistics*), yang digunakan untuk mengkaji interaksi hubungan antara komputer dengan bahasa (alami) manusia. Pemrosesan bahasa alami berupaya memecahkan masalah untuk memahami bahasa alami manusia, dengan segala aturan gramatika dan semantiknya, dan mengubah bahasa tersebut menjadi representasi formal yang dapat diproses oleh komputer.

Dalam penerapannya, tujuan dari pemrosesan bahasa alami untuk memahami bahasa manusia ini memiliki banyak tantangan atau hambatan, antara lain :

1. Penandaan kelas kata (*part-of-speech tagging*). Sulit untuk menandai kelas kata (kata benda, kata kerja, kata sifat dan sebagainya) suatu kata dalam teks karena pengkategorian kata sangat bergantung kepada konteks penggunaannya.

2. Segmentasi teks (*text segmentation*). Penentuan segmentasi sulit dilakukan pada bahasa tulis yang tidak memiliki pembatas kata spesifik (misalnya bahasa mandarin, Jepang, dan Thailand) serta pada bahasa lisan yang kadang membaurkan bunyi antarkata.

3. Disambiguasi makna kata (*word sense disambiguation*). Banyak kata memiliki lebih dari satu makna, baik dalam bentuk homonim (makna berbeda, namun terkait, misalnya ”ragu” dalam makna ”bimbang” dan ”sangsi”). Pembedaan makna hanya dapat dilakukan dengan melihat konteks penggunaan.

4. Ambiguitas sintaksis (*syntactic ambiguity*). Suatu bahasa memiliki berbagai kemungkinan struktur kalimat. Pemilihan struktur yang paling tepat biasanya membutuhkan gabungan informasi semantik dan kontekstual.

5. Masukan yang tak sempurna atau tak teratur (*imperfect or irregular input*). Aksen dalam bahasa lisan serta kesalahan ejaan dan gramatikal dalam bahasa tulis menyulitkan pemrosesan bahasa alami.

6. Pertuturan (*speech act*). Struktur kalimat saja kadang tidak dapat dengan tepat menggambarkan maksud penutur atau penulis. Kadang gaya bahasa dan konteks menentukan maksud yang diinginkan.

Selain dari tantangan atau hambatan yang tersebut di atas, pemrosesan bahasa alami telah berhasil diterapkan untuk berbagai tugas yang awalnya hanya dapat dilakukan oleh manusia. Beberapa bidang yang telah mampu menerapkan pemrosesan bahasa alami adalah sebagai berikut:

1. Pemerolehan informasi (*information retrieval*). Pencarian terhadap dokumen yang relevan, pencarian informasi spesifik di dalam dokumen, serta pembuatan metadata.

2. Penjawaban pertanyaan (*question answering*). Secara otomatis menjawab pertanyaan yang diajukan dengan bahasa alami dengan jawaban dalam bahasa alami pula.

3. Perangkuman otomatis (*automatic summarization*). Pembuatan versi singkat berisi butir-butir penting dari suatu dokumen dengan menggunakan program komputer.

4. Penerjemahan mesin (*machine translation*). Penerjemahan otomatis dari suatu bahasa alami ke bahasa lain.

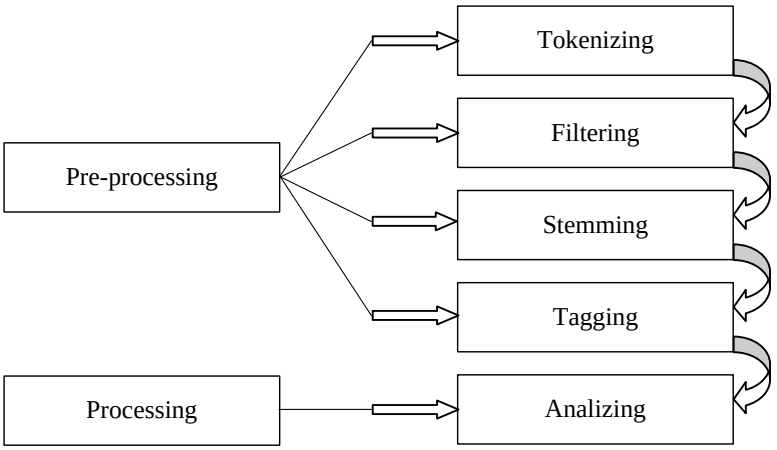
5. Pengenalan wicara (*speech recognition*). Pengubahan bahasa lisan menjadi masukan yang dikenali oleh mesin, misalnya pada pendiktean bahasa lisan kepada komputer untuk menghasilkan bahasa tulis atau pelaksanaan suatu perintah oleh komputer berdasarkan bahasa lisan dari manusia.

6. Sintesis wicara (*speech synthesis*). Pengubahan bahasa tulis menjadi bahasa lisan, merupakan kebalikan dari pengenalan wicara.

7. Pengenalan karakter optis (*optical character recognition*). Pengubahan tulisan tangan atau teks tercetak (biasanya melalui pemindai/*scanner*) menjadi dokumen yang dapat dikenali oleh mesin.

8. Analisis sentimen (*sentiment analysis*). Ekstraksi informasi dari sumber data teks untuk mendeteksi pandangan positif atau negatif terhadap suatu objek. Biasanya diterapkan untuk mengidentifikasi tren opini publik terhadap suatu produk atau perusahaan.

II.3 Proses implementasi pertambangan teks



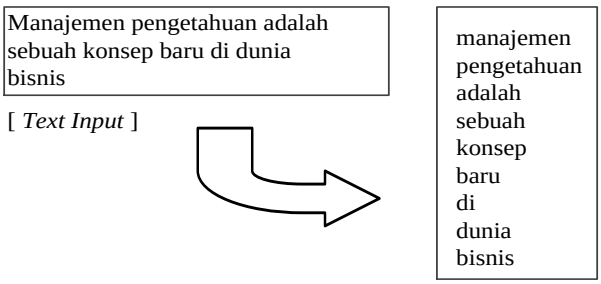
Gambar 1 Proses Implementasi Text Mining

Secara garis besar dalam melakukan proses implementasi *text mining* terdiri dari 2 tahap besar seperti pada Gambar 1, yaitu:

1. ***Pre-processing*** adalah tahap di mana aplikasi melakukan seleksi data yang akan diproses pada setiap dokumen. Setiap kata akan dipecah-pecah menjadi struktur bagian kecil yang nantinya akan mempunyai makna sempit. Tujuan dilakukan *pre-processing* adalah memilih setiap kata dari dokumen dan merubahnya menjadi kata dasar yang memiliki arti sempit. Ada beberapa hal yang perlu dilakukan pada tahap *pre-processing* ini, yaitu :

   a. Tokenizing

   Proses ini memotong setiap kata dalam teks, dan mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ’a’ sampai ’z’ yang diterima, sedangkan karakter selain huruf dihilangkan. Jadi hasil dari proses *tokenizing* adalah kata-kata yang merupakan penyusun kalimat/*string* yang dimasukkan seperti contoh pada Gambar 2.

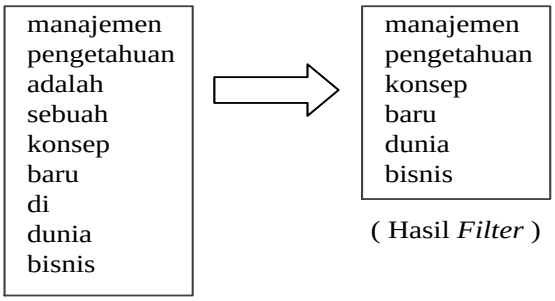


Gambar 2 Proses *Tokenizing*

   b. Filtering

   Pada proses ini dilakukan proses *filter* atau penyaringan kata hasil dari proses *tokenizing*, di mana kata yang tidak relevan dibuang. Proses ini menggunakan pendekatan *stoplist/stopword* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan

bag-of-words. Yang termasuk *stoplist* adalah ”yang”, ”di”, ”dari”, dan lain-lain. Contoh dari proses ini dapat dilihat pada Gambar 3.

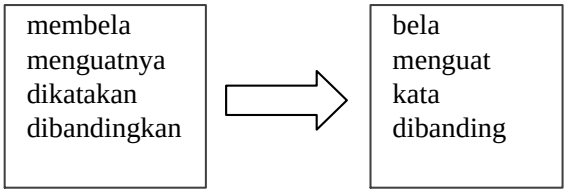


(Hasil *Token*) (Hasil *Filter*)

Gambar 3 Proses *Filtering*

c. Stemming

Stemming adalah proses untuk menggabungkan atau memecahkan setiap varian-varian suatu kata menjadi kata dasar. *Stem* (akar kata) adalah bagian dari akar yang tersisa setelah dihilangkan imbuhannya (awalan dan akhiran). Proses ini kebanyakan dipakai untuk teks berbahasa inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen. Contoh dari proses ini pada teks berbahasa inggris dapat dilihat pada Gambar 4.



(Hasil *Filter*) (Hasil *Stem*)

Gambar 4 Proses *Stemming*

d. Tagging

Tagging adalah suatu proses mencari bentuk asal dari kata bentuk lampau. Proses ini tidak digunakan pada teks berbahasa indonesia karena kata dalam bahasa indonesia tidak memiliki bentuk lampau.

2. Processing adalah tahap inti di mana setiap kata akan diolah dengan algoritma tertentu sehingga mempunyai bobot terhadap setiap dokumen yang akan diseleksi. Dalam tahap ini, proses yang digunakan adalah *Analizing*.

a. Analizing

Proses ini dilakukan dengan melakukan perhitungan bobot dokumen agar diketahui seberapa jauh tingkat similaritas atau persamaan antara *keyword* yang dimasukkan dengan dokumen.
Dalam tahap *processing*, dokumen akan dianalisa oleh aplikasi. Secara umum terdapat dua jenis metode yaitu metode yang tidak melakukan perhitungan bobot kalimat dan yang melakukan perhitungan bobot kalimat. Metode yang tidak menghitung bobot kalimat hanya mengambil beberapa kalimat awal dan akhir. Metode-metode yang menghitung bobot kalimat menggunakan bobot *term* (kata maupun pasangan kata) dari setiap *term* yang terdapat dalam kalimat tersebut.

Bobot *term* diperoleh dengan melakukan perhitungan sederhana terhadap *Term Frequency-Inverse Document Frequency* dari *term* tersebut yaitu *TF/IDF*. Dengan menggunakan *title factor*, bobot dari sebuah *term* dapat ditingkatkan jika *term* tersebut terdapat dalam judul. Penerapan terhadap dokumen akan dilakukan dengan menggunakan kombinasi dari Algoritma *TF/IDF* dan *Vector Space*.

II.4 Metode K-Means

Data *Clustering* merupakan salah satu metode *Data Mining* yang bersifat tanpa arahan (*unsupervised*). Ada dua jenis data *clustering* yang sering dipergunakan dalam proses pengelompokan data yaitu *hierarchical* (hirarki) data *clustering* dan *non-hierarchical* (non hirarki) data *clustering*. *K-Means* merupakan salah satu metode data *clustering* non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*/kelompok. Metode ini mempartisi data ke dalam *cluster*/kelompok sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu *cluster* yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam kelompok yang lain. Adapun tujuan dari data *clustering* ini adalah untuk meminimalisasikan *objective function* yang diset dalam proses *clustering*, yang pada umumnya berusaha meminimalisasikan variasi di dalam suatu *cluster* dan memaksimalisasikan variasi antar *cluster*.

Data *clustering* menggunakan metode *K-Means* ini secara umum dilakukan dengan algoritma dasar sebagai berikut:

a. Tentukan jumlah *cluster*

b. Alokasikan data ke dalam *cluster* secara *random*

c. Hitung *centroid*/rata-rata dari data yang ada di masing-masing *cluster*

d. Alokasikan masing-masing data ke *centroid*/rata-rata terdekat

e. Kembali ke Step 3, apabila masih ada data yang berpindah *cluster* atau apabila perubahan nilai *centroid*, ada yang di atas nilai *threshold* yang ditentukan atau apabila perubahan nilai pada *objective function* yang digunakan di atas nilai *threshold* yang ditentukan

II.5 Algoritma TF/IDF (Term Frequency-Inversed Document Frequency)

Pada algoritma TF/IDF digunakan rumus atau formula untuk menghitung bobot (W) masing-masing dokumen terhadap kata kunci, dapat dilihat pada rumus (1):

$$W_{d,t} = tf_{d,t} * IDF_t \quad (1)$$

dengan:

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 = *Inversed Document Frequency*;

IDF = log (D/df);

D = jumlah dokumen;

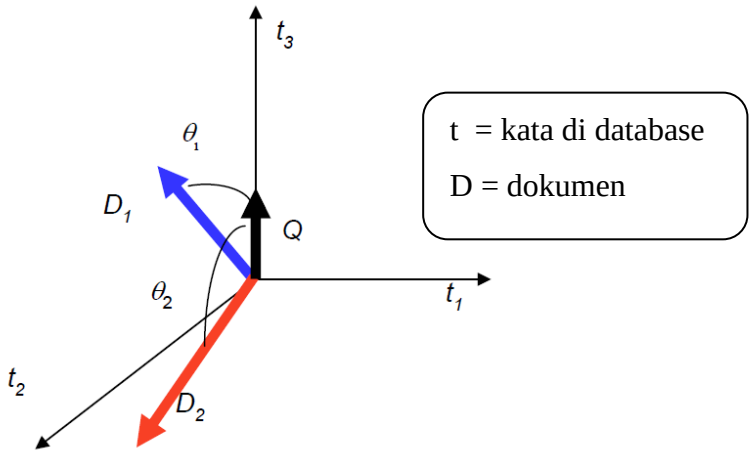
Df = banyak dokumen yang mengandung kata yang dicari.

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses *sorting* di mana semakin besar nilai W maka semakin besar tingkat similaritas dokumen terhadap kata yang dicari.

II.6 Algoritma Model Ruang Vektor (*Vector Space Algorithme*)

Model ruang vektor 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 ruang vektor adalah dengan cara melakukan perluasan vektor. Proses perluasan dapat dilakukan pada vektor *query*, vektor dokumen, atau pada kedua vektor tersebut. Pada algoritma model ruang vektor gunakan rumus untuk mencari nilai cosinus sudut antara dua vektor dari setiap bobot dokumen (WD) dan bobot dari kata kunci (WK). Rumus yang digunakan dapat dilihat pada rumus (2).



$$CosSim(d_j, q) \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t} (W_{ij} \cdot W_{iq})}{\sqrt{\sum_{i=1}^{t} W_{ij}^2 \cdot \sum_{i=1}^{t} W_{iq}^2}} \quad (2)$$

II.7 HTML (Hypertext Markup Language)

II.7.1 Pendahuluan HTML

HTML atau yang memiliki kepanjangan *Hypertext Markup Language* adalah *script* dimana kita bisa menampilkan informasi dan daya kreasi kita lewat *internet*. HTML sendiri adalah suatu dokumen teks biasa yang mudah dimengerti dibanding bahasa pemrograman lainnya, dan karena bentuknya itu maka HTML dapat dibaca oleh berbagai *platform* seperti: Windows, Linux, Macintosh. Kata "*Markup Language*" pada HTML menunjukkan fasilitas yang berupa tanda tertentu dalam skrip HTML dimana kita bisa mengatur judul, garis, tabel, gambar, dan lain-lain dengan perintah yang telah ditentukan pada elemen HTML.

II.7.2 Bagian-bagian HTML

HTML terdiri dari beberapa bagian yang fungsinya sebagai penanda suatu kelompok perintah tertentu, misalnya kelompok perintah *form* yang ditandai dengan kode <form>, judul dengan <title> dan sebagainya. Untuk lebih lanjut mengenai bagian-bagian HTML perhatikan skema dibawah ini :

<html>

<head>

<title>…</title>

</head>

<body>

… isi dari halaman web …

</body>

</html>

Keterangan:

1. Dokumen HTML selalu diawali dengan tanda tag pembuka <html> dan diakhiri dengan tag penutup </html>.

2. Pada elemen *head* <head>, dapat kita sisipkan kode-kode untuk menuliskan keterangan tentang dokumen HTML, atau dapat juga kita sisipkan *scripts* pemograman *web* seperti *JavaScript, VBScripts*, atau *CSS* untuk menambah daya tarik pada situs yang kita buat agar lebih menarik dan dinamis.

3. Elemen <body> </body> berisi *tag-tag* untuk isi atau *layout* tampilan pada situs, seperti : , <table>, </table>, <form>, </form>. *Tag* adalah kode-kode yang digunakan untuk mengatur dokumen HTML. Secara garis besar bentuk umum tag adalah sebagai berikut :

<tag-awal>TEKS<tag-akhir>

Namun ada juga tag yang tidak perlu ada tag penutup seperti
, <hr>, <img>, dan lain-lain sebagainya.

II.8 Pengenalan PHP (*Personal Home Page*)

PHP singkatan dari PHP *HyperText Preprocessor* yang digunakan sebagai bahasa S*cript Server-Side* dalam pengembangan *Web* yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga *maintenance web* tersebut menjadi lebih mudah dan efisien.

PHP merupakan S*oftware Open Source* yang disebarkan dan dilisensikan secara gratis serta dapat di*download* secara bebas dari situs resminya http://www.php.net. Pengguna dapat mengubah *Soure Code* dan mendistribusikannya secara bebas serta diedarkan secara gratis.

II.8.1 Sejarah PHP

PHP diciptakan pertama kali oleh Ramus Lerdorf pada tahun 1994. Awalnya, PHP digunakan untuk mencatat jumlah serta untuk mengetahui siapa saja pengunjung pada *homepag*e-nya. Rasmus Lerdorf adalah salah seorang pendukung *Open Source*. Oleh karena itu, Rasmus mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis, kemudian menambah kemampuan PHP 1.0 dan meluncurkan PHP 2.0.

Pada tahun 1996, PHP telah banyak digunakan dalam *Website* di dunia. Sebuah kelompok pengembang *software* yang terdiri dari Rasmus, Zeew Surasaki, Andi Gutman, Stig Bakken, Shane Caraveo dan Jim Winstead berkerja sama untuk menyempurnakan PHP 2.0. Akhirnya, pada tahun 1998 PHP 3.0 dikeluarkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan PHP 4.0.

Tahun 2004 bulan juli dirilis PHP 5 dengan inti *Zend Engine* 2.0. PHP 5 adalah versi PHP terbaru yang mendukung penuh *object-oriented programming* (OOP), integrasi XML, mendukung semua eksistensi terbaru MySQL, pengembangan *web service* dengan SOAP dan REST, serta ratusan peningkatan lainnya dibandingkan dengan versi sebelumnya PHP 4.0.

Sejak PHP 5 keluar eksistensi SQLite sudah langsung tersedia dalam PHP. SQLite adalah *Embeddable* mesin *database* SQL yang tidak hanya mengharuskkan *client* terkoneksi ke sebuah *database server* misalnya MySQL.

II.8.2 Kelebihan dan Kelemahan PHP

PHP memiliki kelebihan yang tidak dimiliki oleh bahasa *script* sejenis. PHP difokuskan pada pembuatan S*cript Server-Side*, yang bisa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data dari *form*, menghasilkan isi halaman *web* dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih dari pada kemampuan CGI.

PHP dapat digunakan pada semua sistem operasi, antara lain *Linux, Unix, Microsoft Windows, Mac OS X, RISC OS*. PHP juga mendukung banyak *Web Server*, seperti *Apache*, *Microsoft Internet Information Server* (MIIS), *Personal Web Server* (PWS), *Netscape* and *iPlanet servers* dan masih banyak lainnya.

PHP tidak hanya terbatas pada hasil keluaran HTML (*HyperText Markup Languages*). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, *filePDF*, dan *movies Flash*. PHP juga dapat menghasilkan teks seperti XHTML dan XML lainnya.

Fitur-fitur yang banyak dapat diandalkan oleh PHP adalah dukungannya terhadap banyak *database*. Berikut *database* yang dapat didukung oleh PHP:

1. *Adabas D*

2. *dBase*
3. *Direct MS-SQL*
4. *Empress*
5. *FilePro* (*read only*)
6. *FrontBase*
7. *Hyperwave*
8. *IBM DB2*
9. *Informix*
10. *MSQL*
11. *MySQL*
12. *PostgrSQL*
13. *Unix DBM*
14. *Solid*
15. *Sybase*
16. *Velocis*

Adapun kelemahan PHP adalah :

1. Tidak ideal untuk pengembangan skala besar.
2. Tidak bisa memisahkan antara tampilan dengan logik dengan baik (walau penggunaan *template* dapat memperbaikinya).

3. PHP memiliki kelemahan *security* tertentu apabila *programmer* tidak jeli dalam pemrograman dan kurang memperhatikan isu dan konfigurasi PHP.

II.8.3 Penggabungan *Script* PHP dan HTML

Bahasa pemrograman PHP dapat digabungkan dengan HTML dengan terlebih dahulu memberikan tanda *tag* buka dilanjutkan dengan tanda tanya (<?) kemudian ditutup dengan tanda tanya dilanjutkan tanda *tag* tutup (?>). Ada dua tipe pengabungan PHP dam HTML yaitu:

1. Embbedded Script

Embedded script adalah *script* PHP yang disisipkan di antara *tag-tag* dokumen HTML. *Embedded script* menempatkan PHP sebagai bagian dari HTML.

Contoh penulisan *Embedded Script* dapat dilihat di bawah ini:

```
<html>

<head>

<title>Embedded Script</title>

</head>

<body>

<?php

echo “Hallo, Selamat menggunakan PHP”;

?>

</body>
```

</html>

2. ***Non-Embedded Script***

Non-Embedded Script adalah *script* atau program PHP murni. Termasuk tag HTML yang disisipkan dalam *script* PHP. *Non-Embedded Script* menempatkan bagian HTML sebagai bagian dari *script* PHP.

Contoh penulisan *Non-Embedded Script* dapat dilihat dibawah ini:

```
<?php
echo “<html>”;
echo “<head>”;
echo “<title> Non-Embedded Script</title>”;
echo “</head>”;
echo “<body>”;
echo “<p>Selamat Menggunakan PHP</p>”;
echo “</body>”;
echo “</html>”;
?>
```

II.8.4 Fungsi dan PHP dan MySQL

Adapun fungsi PHP untuk mengakses MySQL yang biasa digunakan diantaranya adalah:

a. ***mysql_connect()***

Fungsi *mysql_connect* adalah untuk menghubungkan PHP dengan *database* MySQL. Format fungsinya adalah:

mysql_connect (string hostname, sting username, string password);

b. ***mysql_select_db()***

Setelah terhubung ke *database* MySQL dengan menggunakan *mysql_connect*, langkah selanjutnya adalah memilih *database* yang akan digunakan. Fungsi *mysql_select_db* digunakan untuk memilih *database*. Format fungsinya adalah:

mysql_select_db (string database, koneksi);

c. ***mysql_query()***

Dalam *database* MySQL, perintah untuk melakukan transaksi ialah perintah SQL. Sebutan untuk mengirim perintah SQL dinamakan *query*. *Query* memberi perintah kepada *database* untuk melakukan apa yang dikehendaki. Format fungsinya adalah:

int mysql_query(string query, int [link_identifier]);

d. mysql_num_rows()

Kegunaan dari fungsi ini adalah untuk menghitung jumlah baris yang dikenai oleh proses SQL. Format fungsinya adalah:

int mysql_num_rows(int result);

e. ***mysql_fetch_array()***

Fungsi ini berkaitan dengan menampilkan data. Untuk menampilkan data, digunakan fungsi *mysql_fetch_array*. Dengan fungsi ini, hasil *query* ditampung dalam bentuk *array*. Format fungsinya adalah:

Array mysql_fetch_array(int result, int [result_type]);

II.9 MySQL

MySQL merupakan *software* yang tergolong kedalam DBMS yang bersifat *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source* (kode yang dipakai untuk membuat MySQL), selain itu tentu saja bentuk *executable*nya atau kode yang dijalankan secara langsung dalam sistem operasi dan bisa diperoleh dengan cara mendownload di *internet* secara gratis.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Sehingga istilah seperti tabel, baris, dan kolom tetap digunakan dalam MySQL. Pada MySQL sebuah *database* mengandung satu beberapa tabel, tabel terdiri dari sejumlah baris dan kolom. Dalam konteks bahasa SQL, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logika merupakan struktur dua dimensi yang terdiri atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom. Baris pada tabel sering disebut sebagai *instance* dari data sedangkan kolom sering disebut sebagai *attributes* atau *field*.

II.9.1 Perintah SQL

Perintah-perintah SQL yang sering digunakan untuk kebutuhan *web database* diantaranya:

a. INSERT

Digunakan untuk mengisi data atau menambahkan *record* pada suatu tabel.

INSERT INTO nama_tabel (kolom1,kolom2..) VALUES (nilai1, nilai2..);

b. SELECT

Digunakan untuk melihat data dari suatu atau beberapa tabel.

SELECT kolom-kolom;

FROM nama_tabel;

Untuk melihat seluruh isi kolom dari suatu tabel digunakan *query* SELECT *.

SELECT * FROM nama_tabel;

c. WHERE

Digunakan untuk menyaring hasil *query* sehingga *record* yang dikeluarkan hanyalah *record* yang sesuai dengan yang diinginkan.

SELECT kolom1,kolom2

FROM kolom1

WHERE kolom2<kriteria;

d. DISTINCT

Dapat digunakan untuk menghilangkan *record-record* yang sama.

SELECT DISTINCT kolom2 FROM kolom1;

e. BETWEEN

Digunakan untuk membatasi suatu kolom berada pada suatu batas nilai tertentu.

SELECT kolom1,kolom2,kolom3

FROM kolom1

WHERE kolom2 BETWEEN .. AND ..;

f. LIKE

Digunakan untuk pencarian data yang memiliki pola tertentu.

SELECT kolom1,kolom2

FROM kolom1

WHERE kolom1 LIKE ‘A%’;

g. ORDER BY

Digunakan untuk mensortir data hasil *query* sesuai dengan kebutuhan.

SELECT kolom1,kolom2

FROM kolom1

ORDER BY kolom1;

Untuk mensortir dengan urutan terbalik, digunakan *keyword* tambahan DESC. Sedangkan untuk urutan yang teratur digunakan *keyword* ASC.

SELECT kolom1,kolom2

FROM kolom1

ORDER BY kolom1 DESC;

h. DELETE

Digunakan untuk menghapus suatu *record* dengan kriteria tertentu.

DELETE FROM nama_tabel WHERE criteria;

Untuk menghapus *record* pada suatu tabel, digunakan perintah DELETE tanpa menentukan kriteria.

DELETE FROM nama_tabel;

i. UPDATE

Digunakan untuk memodifikasi nilai kolom dari suatu *record*.

UPDATE nama_tabel

SET nama_kolom1=nilai_baru1,nilai_kolom2=nilai_baru2,..

WHERE criteria;

II.10 Perpustakaan Digital (*Digital Library*)

Perpustakaan Digital (*Digital Library*) adalah perpustakaan yang mengelola atau memproses informasi dan menyimpan informasi secara komputerisasi sebagai alternatif atau pelengkap terhadap cetakan konvensional dan disajikan menggunakan protokol melalui jaringan komputer baik itu internet maupun intranet sehingga dapat diakses dari satu tempat ke tempat lain dalam waktu yang singkat sehingga hambatan fisik yang sering terjadi pada perpustakaan bisa diatasi.

Manfaat terbesar dari perpustakaan digital ini adalah akses tak terbatas terhadap sebuah artikel ilmiah atau sebuah buku elektronik (*electronic book*). Artikel atau buku yang berada dalam format elektronik tidak pernah *out of print*, sedangkan artikel yang terbit dalam versi cetak, seringkali terbatas jumlah terbitannya. Perpustakaan digital yang dapat diakses *online* membuat akses terhadap artikel ilmiah atau buku pun menjadi lebih mudah.

II.11 Mesin Pencari Dokumen Dengan Pengklasteran Secara Otomatis

Web mining untuk pencarian berdasarkan kata kunci dengan pengklasteran otomatis adalah suatu metode pencarian dokumen dengan cara mengelompokkan atau mengklaster dokumen dari dokumen-dokumen berdasarkan kata kuncinya. Selanjutnya dilakukan pengklasteran dengan metode *centroid linkage hierarchical*

method (CLHM) terhadap jumlah kata kunci yang diperoleh dari masing-masing dokumen. Dalam pengklasteran, umumnya harus dilakukan inisialisasi jumlah klaster yang ingin dibentuk terlebih dahulu, padahal pada beberapa kasus pengklasteran, user bahkan tidak tahu berapa banyak klaster yang bisa dibangun. Untuk itu, pada makalah ini diaplikasikan metode *Valley Tracing* sebagai *constraint* yang akan melakukan identifikasi terhadap pergerakan varian dari tiap tahap pembentukan klaster dan menganalisa polanya untuk membentuk suatu klaster secara otomatis (*automatic clustering*). Data yang digunakan adalah data hasil dari proses *text mining* pada dokumen. Dari percobaan yang dilakukan dengan 424 dokumen hasilnya memberikan simpulan bahwa pada umumnya pencarian dokumen menggunakan teknik pengklasteran dengan algoritma CLHM dapat digunakan untuk mengelompokkan dokumen dengan jumlah yang tepat secara otomatis.

II.12 Alat Penambangan Teks

Berikut ini beberapa perangkat lunak komersial dan bebas yang dapat digunakan sebagai alat untuk melakukan penambangan teks.

a. Komersial

Berikut ini daftar beberapa perangkat lunak komersial untuk penambangan teks.

- *ClearForest*[12]
- *IBM Intelligent Miner Data Mining Suite* (bagian dari IBM Info *Sphere Warehouse*)[13]
- *Megaputer TextAnalyst*[14]
- *SAS Text Analytics*[15]
- *SPSS Text Mining for Clementine*[16]

- *Statistica Text Miner*[17]
- *VantagePoint*[18]
- *WordStat*[19]

b. Bebas

Berikut ini daftar beberapa perangkat lunak bebas untuk penambangan teks. Beberapa di antaranya juga merupakan perangkat lunak sumber terbuka.

- *GATE (General Architecture for Text Engineering)*[20]
- *LingPipe*[21]
- *LPU* (tadinya *S_EM*)[22]
- *RapidMiner*[23]
- *UIMA*[24]

c. Daring

Berikut beberapa alat daring yang dapat digunakan untuk penerapan spesifik penambangan teks.

- *Ranks.nl*[25]
- *Wordle*[26]

II.13 Bidang Penerapan Penambangan Teks

Penambangan data telah diaplikasikan dalam beberapa bidang seperti dijabarkan berikut ini.

a. **Pemasaran**. Penambangan teks terhadap transkripsi percakapan pusat panggilan (*call center*), tulisan blog, ulasan produk oleh situs independen, dan

diskusi pada forum diskusi daring telah digunakan untuk menganalisis persepsi dan sentimen konsumen terhadap produk atu produsen. Informasi ini dapat dipakai untuk meningkatkan kepuasan dan nilai produk bagi pelanggan.

b. **Keamanan**. Penambangan teks telah digunakan antara lain sebagai sumber intelijen dalam Perang Dingin (*Echelon* oleh Amerika Serikat, Australia, Inggris, Kanada, dan Selandia Baru), pelacakan kejahatan terorganisasi lintas negara (OASIS oleh Europol, Uni Eropa), serta pemantauan keamanan gabungan oleh FBI, CIA, dan Departemen Keamanan AS. Selain itu, penambangan teks telah dipakai untuk mendeteksi kebohongan terhadap pernyataan tertulis, sebagai alternatif dari metode *poligraf* yang hanya dapat diterapkan untuk pernyataan lisan.

c. **Biomedis**. Penambangan teks berpotensi untuk memproses literatur dalam bidang ini secara otomatis karena (1) jumlah publikasi meningkat pesat, (2) literatur bidang medis lebih terstandardisasi dan teratur, dan (3) terminologi yang digunakan relatif konstan dengan ontologi yang cukup baku.

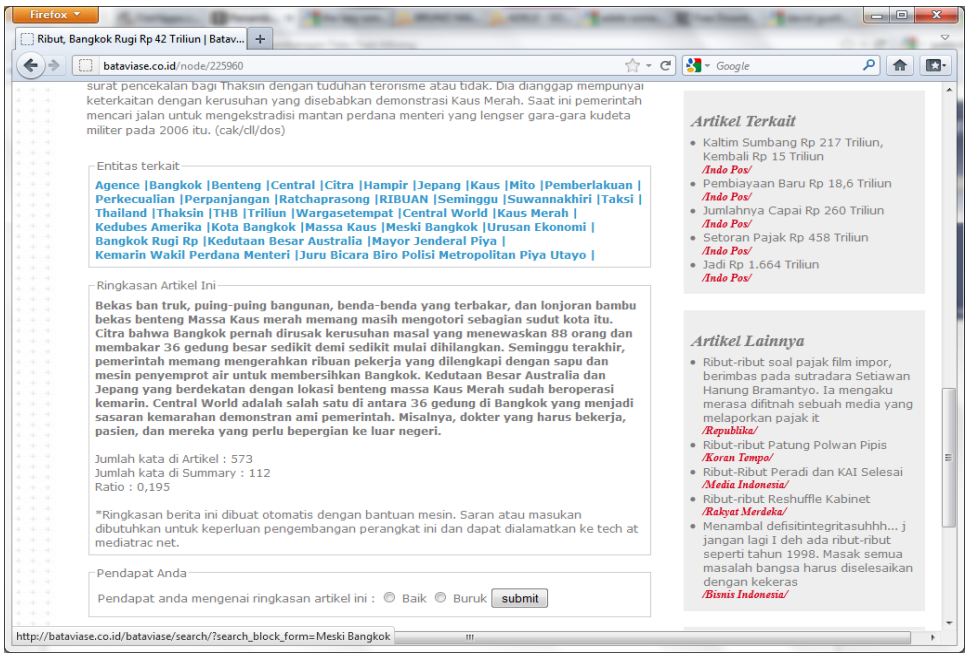
d. **Akademis**. Penambangan teks telah dimanfaatkan oleh berbagai penerbit jurnal akademis dan lembaga pendidikan untuk memproses basis data artikel besar yang memerlukan pengindeksan untuk membantu para pencari informasi. Prakarsa yang telah dilakukan pada bidang ini antara lain adalah *Open Text Mining Interface (Nature), Journal Publishing Document Type Definition (National Institute of Health), National Centre for Text Mining (University of Manchester and Liverpool)*, serta *BioText (University of California, Barkeley)*.

II.14 Penerapan penambangan teks bahasa Indonesia

Berikut adalah beberapa contoh penerapan penambangan data untuk bahasa Indonesia yang dapat ditemukan di Internet.

a. *Bataviase*: Perangkum berita otomatis

Bataviase <http://bataviase.co.id/> adalah situs yang membuat ringkasan atau rangkuman berita secara otomatis. *Bataviase* menerapkan perangkuman otomatis dari penambangan teks untuk membuat ringkasan berita dari berbagai surat kabar di Indonesia. Selain itu, *Bataviase* juga menerapkan kategorisasi berdasarkan 19 kategori yang telah ditentukan serta pelacakan topik dalam bentuk artikel terkait.



Gambar 5 Ringkasan artikel otomatis pada *Bataviase*

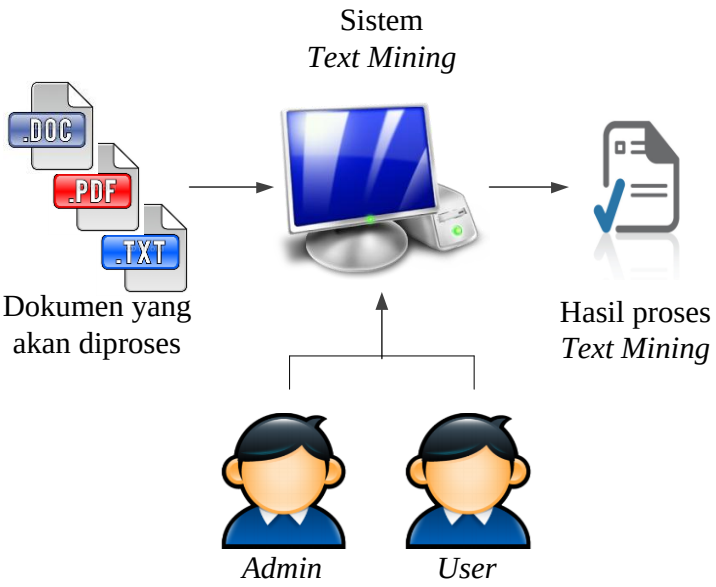
b. SITTI: *Platform* iklan kontekstual

SITTI <http://www.sitti.co.id/> adalah layanan *platform* iklan kontekstual yang menampilkan iklan sesuai dengan target yang diinginkan pemasang iklan. SITTI memanfaatkan ekstraksi informasi untuk mencari kata kunci yang terkait dengan suatu laman *web* dan pelacakan topik untuk menampilkan iklan yang sesuai dengan pengunjung laman tersebut. *Platform* lain yang memanfaatkan teknologi yang mirip dengan SITTI adalah *Google AdWords*.

Gambar 6 Beranda SITTI

Bab III Analisis dan Perancangan

III.1 Deskripsi Umum Sistem



Gambar 7 Deskripsi Umum Sistem *Text Mining*

Sistem *Text Mining* adalah sebuah sistem yang mampu mengolah file-file dokumen seperti file-file berbentuk *Word, PDF*, dan *Text* yang kemudian diproses melalui sistem dengan melakukan beberapa proses penyaringan dan dilanjutkan dengan proses analisa yang mampu memberikan hasil proses *Text Mining* berupa kata kunci (*keyword*) dari sebuah dokumen yang telah diproses.

III.2 Fitur Utama Perangkat Lunak

Sistem *Text Mining* dapat memproses sebuah dokumen yang berisi informasi-informasi yang panjang sehingga menghasilkan kata kunci, dan pengelompokkan yang telah didefinisikan terlebih dahulu.

III.3 Kebutuhan Fungsional

Adapun kebutuhan fungsional dari sistem pemanggil antrian, antara lain :

F-001 Admin dapat mengunggah file dengan format pdf, doc, dan txt

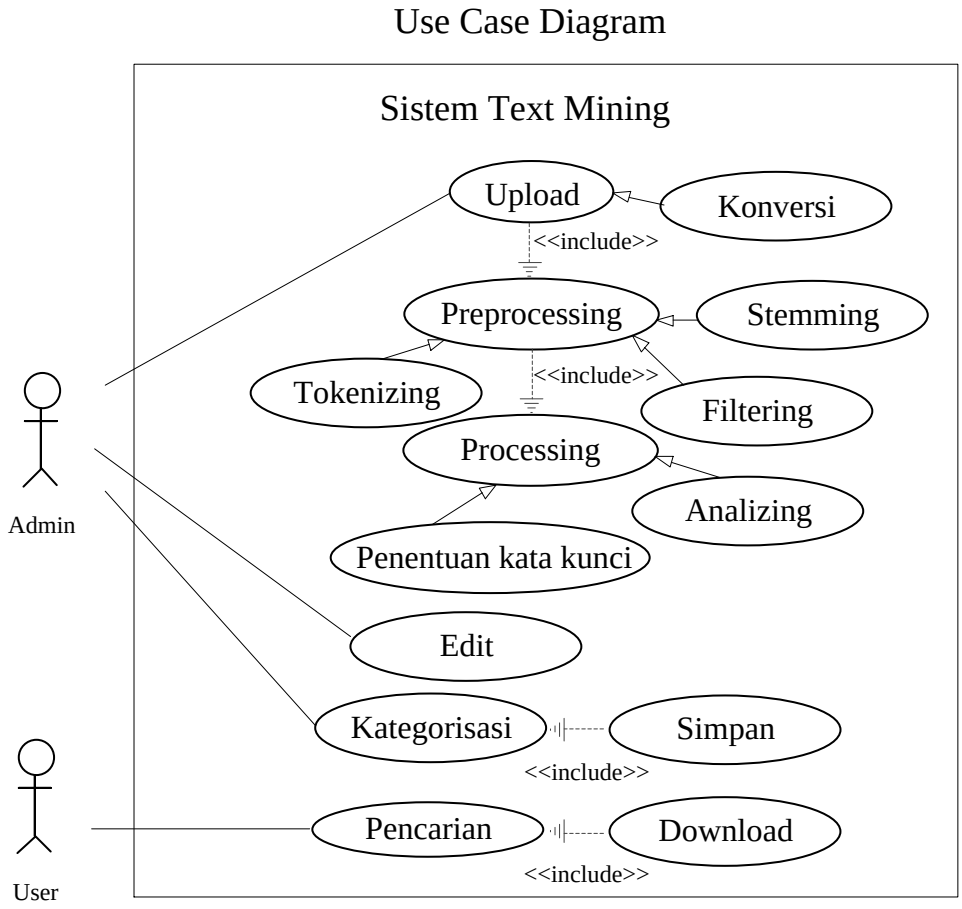
F-002 Sistem dapat melakukan proses *preprocessing* terhadap dokumen

F-003 Sistem dapat melakukan proses *processing* terhadap dokumen

F-004 Admin dapat menentukan kategorisasi terhadap dokumen tersebut

F-005 User dapat melakukan pencarian baik dengan kata kunci atau kategorisasi

III.4 Diagram Use Case



Gambar 8 Diagram *Use Case* Sistem *Text Mining*

III.5 Skenario Use Case

III.5.1 Use Case <Upload>

Kondisi Awal : Belum terdapat *file* dokumen yang akan diupload.

Kondisi Akhir : *File* dokumen telah selesai diupload dan siap dilakukan proses *Text Mining*.

Skenario : Pada awalnya sebuah *file* dokumen yang akan dilakukan proses *Text Mining* diseleksi melalui *browser window* oleh *user*

dengan menekan sebuah tombol menu *browse*, setelah *browse* telah selesai dan dilanjutkan oleh *user* dengan menekan tombol *upload* maka *file* yang diseleksi akan disimpan kedalam *webserver* dan *file* pun telah selesai diupload dengan format *file* antara lain : format dokumen .txt, .doc, .pdf, .html, .xml untuk dilakukan proses *text mining* dengan membaca seluruh isi dokumen dan mengkonversi dokumen menjadi dengan format *file* .txt.

III.5.2 Use Case <Preprocessing>

Kondisi Awal : *File* dokumen belum diproses.

Kondisi Akhir : *File* dokumen telah dipotong menjadi kata-kata berupa data.

Skenario : Proses *tokenizing* mengubah huruf besar pada dokumen menjadi huruf kecil (huruf yang akan diterima hanya dari ‘a’ – ‘z’), menghilangkan tanda-tanda pembantu. Kemudian memotong setiap kalimat dalam teks dokumen menjadi per kata. Setelah sebuah *file* dokumen di *tokenizing* maka proses selanjutnya adalah *filtering*, Proses *filtering* adalah proses untuk menyaring atau menghilangkan kata-kata yang tidak bermakna dan kata-kata yang bersifat sebagai kata pembantu dalam sebuah kalimat dengan menggunakan sebuah daftar yang disebut dengan daftar kata pembantu atau “*stoplist*”. Setelah proses *filtering* dilanjutkan dengan *stemming* yaitu proses menghilangkan kata-kata imbuhan awalan dan akhiran pada sebuah kata sehingga menjadi kata-kata dasar yang menghasilkan kata-kata kunci dari dokumen tersebut.

III.5.3 Use Case <Processing>

Kondisi Awal : Kata-kata dasar telah siap dilakukan proses analisa.

Kondisi Akhir : Telah muncul hasil analisa perhitungan dari dokumen yang

telah di *processing*.

Skenario : Setelah proses *Preprocessing* selesai, maka dilanjutkan dengan proses *Processing*. Dari hasil proses *Preprocessing* akan dilakukan proses analisa terhadap bobot terhadap tingkat kemunculan kata-kata kunci pada sebuah dokumen dan dilanjutkan dengan melakukan pengetesan terhadap seluruh dokumen untuk mencari tingkat nilai masing-masing dokumen yang ada di *database*.

III.5.4 Use Case <Edit>

Kondisi Awal : Data dokumen telah berada di *database*.

Kondisi Akhir : Data dokumen telah selesai dirubah.

Skenario : Data dokumen yang telah ditambah dapat dilakukan perubahan apabila perlu penambahan maupun perubahan terhadap data yang telah ditambah sehingga data yang telah dirubah disimpan kembali ke dalam *database*.

III.5.5 Use Case <Kategorisasi>

Kondisi Awal : Dokumen belum dikategorisasikan.

Kondisi Akhir : Dokumen telah dikategorisasikan.

Skenario : Setelah seluruh perhitungan bobot selesai dilakukan, maka *admin* akan melakukan kategorisasi dokumen dan melakukan penyimpanan terhadap file tersebut.

III.5.6 Use Case <Pencarian>

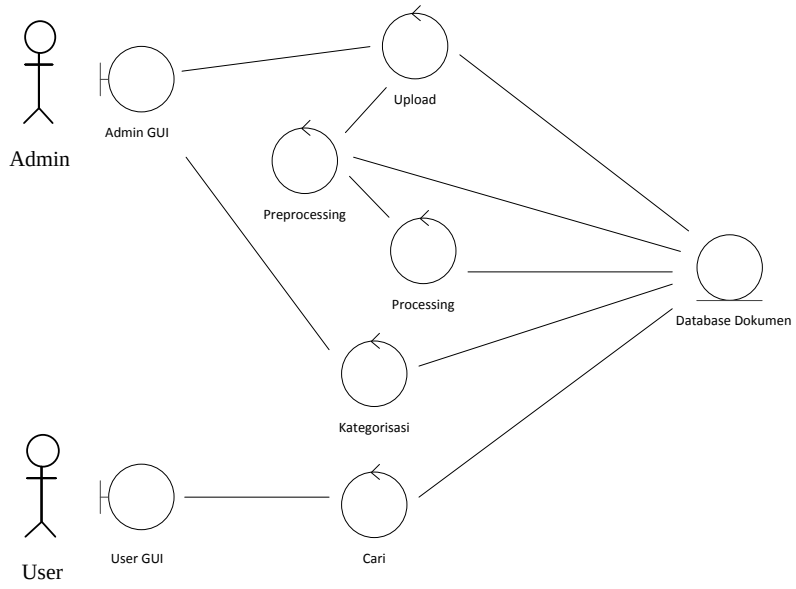
Kondisi Awal : Belum dilakukan pencarian.

Kondisi Akhir : Dokumen sesuai pencarian telah ditemukan.

Skenario : *User* akan melakukan pencarian pada tampilan pencarian dengan mengetikkan kata kunci yang ingin dicari, sehingga hasil pencarian adalah berdasarkan kata-kata kunci yang sering

muncul serta pengurutan dilakukan berdasarkan pada nilai dokumen tersebut atau dengan pilihan menampilkan daftar dokumen terbaru yang ada di dalam kategorisasi dan *user* dapat melakukan proses *download* terhadap dokumen yang dicari.

III.6 Analisis Kelas



Gambar 9 Analisis Kelas Sistem *Text Mining*

Kelas Controller

Sebuah objek *controller* adalah yang merespon semua isyarat yang dilakukan oleh pengguna sistem baik *admin* maupun *user*. Isyarat yang dilakukan pengguna itu melalui tampilan antarmuka. Dalam analisis kelas yang menjadi objek *controller* ini adalah *upload*, *preprocessing*, *processing*, kategorisasi dan cari.

Kelas Boundary

Sebuah *boundary object* mendeklarasikan komponen individu sistem itu sendiri yang terdiri dari antarmuka *admin* dan antarmuka *user*.

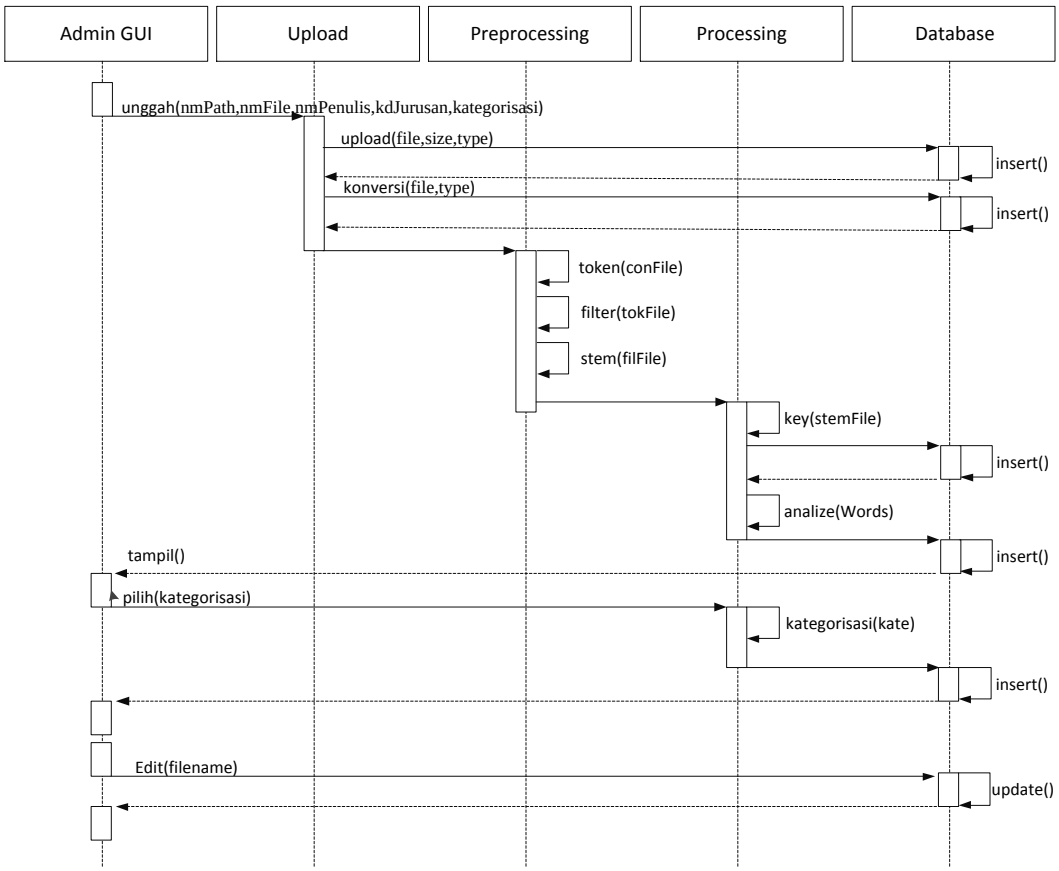
Kelas Entitas

Sebuah objek entitas adalah yang menyatakan tempat penyimpanan data dokumen, hasil proses *preprocessing* dan *processing* yang hasilnya disimpan ke dalam *database*.

III.7 Sequence Diagram

Sequence Diagram adalah penggambaran proses interaksi antar kelas dalam satuan waktu. Berikut adalah penggambaran dari interaksi antar kelas pada sistem *text mining*.

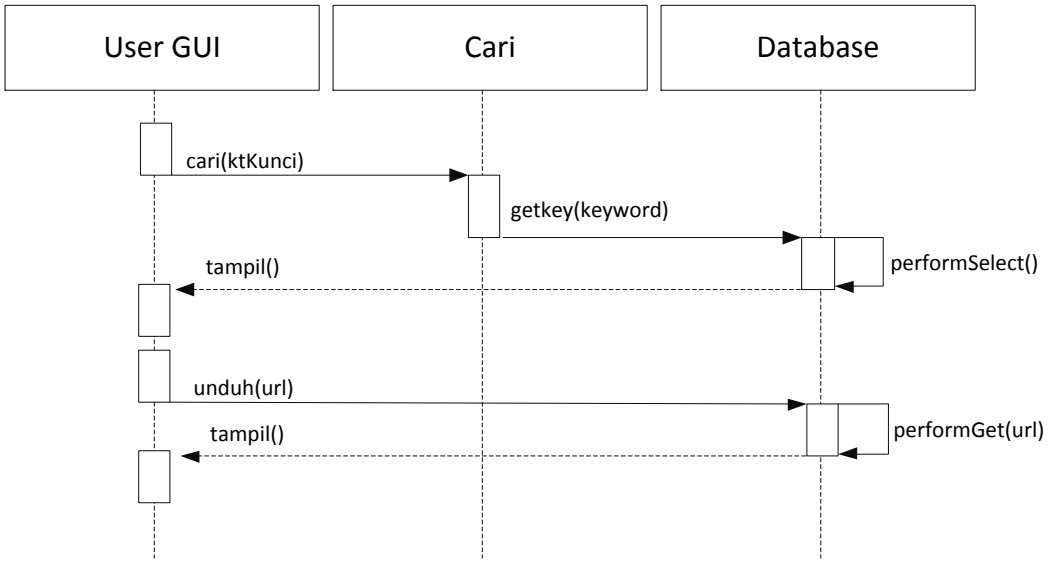
III.7.1 Sequence Diagram Use Case <Upload, Preprocessing, Processing, Edit dan Kategorisasi>



Gambar 10 Diagram *Sequence Upload, Preprocessing, Processing, Edit* dan Kategorisasi

Gambar 10 memperlihatkan *sequence* interaksi antar kelas *admin GUI* menerima inputan *upload*, dan melaksanakan proses *text mining*, menyimpan hasil ke *database* serta menampilkan hasil *text mining* ke *admin GUI*.

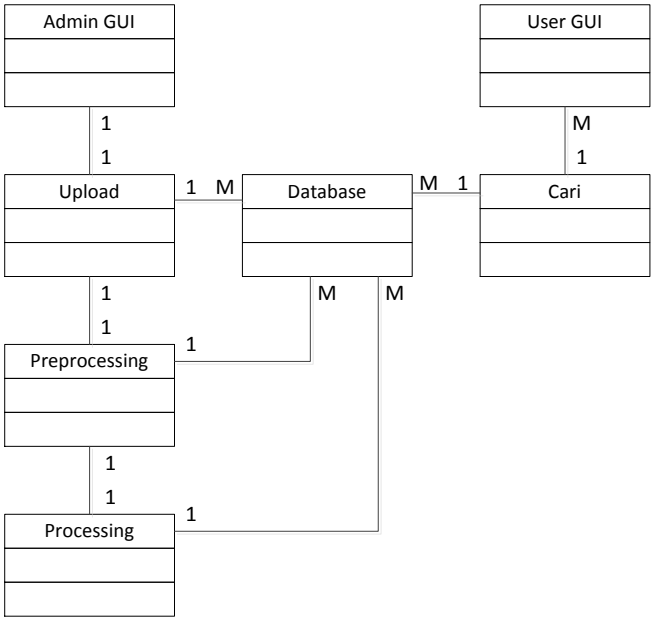
III.7.2 Sequence Diagram Use Case <Cari>



Gambar 11 Diagram *Sequence* Cari dan *Download*

Gambar 11 memperlihatkan *sequence* interaksi antar kelas *user GUI* menerima inputan kata kunci yang akan dicari di dalam database dan menampilkan hasil pencarian ke *user GUI* dan dapat melakukan *download* terhadap dokumen tersebut.

III.8 Diagram Kelas



Gambar 12 Diagram Kelas Sistem *Text Mining*

III.9 Rancangan Kelas Rinci

Kelas <Admin GUI>

Admin
+nmPath : String +nmFile : String +nmPenulis : String +kdJurusan : String +kategorisasi : String
unggah(nmPath,nmFile,nmPenulis,kdJurusan,kategorisasi) : String pilih(kategorisasi) : String edit(nmFile) : String del(nmFile) : String tampil() : void

Kelas <Upload>

Upload
+file :String +size : Integer +type : String
upload(file,size,type) : Void konversi(file,type) : Void

Kelas <Preprocessing>

Preprocessing
+conFile : String +tokFile : String +filFile : String
token(conFile) : String filter(tokFile) : String stem(filFile) : String

Kelas <Processing>

Processing
+stemFile : String +words : Integer +kate : String
key(stemFile) : String analize(words) : Integer kategorisasi(kate) : String

Kelas <User GUI>

User
+ktKunci : String
cari(ktKunci) : String

download(nmFile) : String tampil() : void

Kelas <Cari>

Cari
+keyword: String
getKey(keyword) : String getPath(nmPath) : String

III.10 Algoritma

Nama Kelas : Upload

Nama Operasi : Mengunggah file atau dokumen (upload)

Algoritma : (Algo-001)

```
//upload file atau dokumen
$files ← (file atau dokumen)
If($files > 0) {echo ”error files!!”};
Elseif($files != ’pdf’,’txt’,’doc’){echo ”error file type”;
Elseif($files > 10000000){echo ”maximum file size 10MB”;
Else{if(move_uploaded_file($files[’file’][’tmp_name’], $files[’file’][’name’]
        {echo ”uploaded file success”;}
        Else{ echo ”uploaded file success”;}
```

Nama Kelas : Preprocessing

Nama Operasi : Memilah isi teks dokumen (token)

Algoritma : (Algo-002)

```
//memilah isi teks dokumen
$teks; ← (isi teks dokumen)
$kata; ← (pecah-pecah $teks berdasarkan kata-kata dengan menghilangkan tanda
        ~!@#$%^&*()_+`-={}|[]\:";'<>?,./01234567890)
While ($kata)  // perulangan sebanyak
```

	// kata yang telah ditoken
{$kata_kecil	*← (ubah $kata menjadi huruf kecil*
$kata	*←(pecah-pecah $kata berdasarkan kata-kata dengan menghilangkan tanda* ~!@#$%^&*()_+`-={}|[]\:";'<>?,./ 01234567890

Nama Kelas : Preprocessing

Nama Operasi : Memfilterisasi kata-kata yang tidak bermakna (filter)

Algoritma : (Algo-003)

```
//memfilterisasi kata-kata yang tidak bermakna
Inisialisasi flag = 0
Inisialisai content_filter =””,string_input=hasil token
While (!feof(string_input)),
kata_string=fgets(string_input)
while(!feof(stoplist))
check if string input
```

Nama Kelas : Preprocessing

Nama Operasi : Menghilangkan imbuhan kata menjadi kata dasar (stem)

Algoritma : (Algo-004)

```
Masukkan hasil filter kedalam arrayWord.
Inisialisasi word=arrayWord[i].
Cek awalan dengan memasukkan ke dalam fungsi
cekAwalan(word). Kemudian cek resultWord.
Apabila resultWord is empty maka word =
ArrayWord[i], jika tidak maka word = resultWord().
Cek akhiran dengan memasukkan ke dalam fungsi
cek Akhiran(word). Kemudan cek resultWord.
Apabila resultWord is empty maka word =
ArrayWord[i], jika tidak maka word = resultWord().
```

Cek KPTS dengan memasukkan ke dalam fungsi cek
KPTS(word). Kemudan cek resultWord.
Apabila resultWord is empty maka word =
ArrayWord[i], jika tidak maka word = resultWord().

Nama Kelas : Processing
Nama Operasi : Menghitung tingkat kemunculan sebuah kata (hitung)
Algoritma : (Algo-005)

Insialisasi word=keyWord[i]
Inisialisasi jumlah;
Cek setiap word
Jika word sama yang muncul
Maka keyWord[i] = jumlah + 1

Nama Kelas : Cari
Nama Operasi : mencari sebuah dokumen berdasarkan kata kunci (cari)
Algoritma : (Algo-006)

Inisialisasi keyWords;
Input keyWords yang akan dicari
Temukan dokumen sesuai dengan keyWords yang diinput
Jika ada, maka tampilkan list dokumen sesuai keywords
Jika tidak, maka tampilkan "Tidak ada dokumen yang sesuai dengan keywords"
end

Nama Kelas : Cari
Nama Operasi : mendownload file yang telah dicari (cari)
Algoritma : (Algo-007)

Inisialisasi keyWords;
Input keyWords yang akan dicari
Temukan dokumen sesuai dengan keyWords yang diinput

Jika ada, maka tampilkan list dokumen sesuai keywords

Jika tidak, maka tampilkan "Tidak ada dokumen yang sesuai dengan keywords"

//Download file dengan

Select url from dokumen where namaFile = "nama dokumen";

end

Nama Kelas : Admin GUI

Nama Operasi : mengedit data file yang telah diupload (Edit)

Algoritma : (Algo-008)

Inisialisasi namaFile;

Saat tombol edit ditekan

Select namaFile from dokumen;

Data dokumen ditampilkan sesuai namaFile

Data dokumen diedit

Update dokumen set (judul, penulis1, penulis2, penulis3, jurusan) values ("judul","penulis1","penulis2","penulis3","jurusan")

end

Nama Kelas : Admin GUI

Nama Operasi : menghapus data file yang telah diupload (Hapus)

Algoritma : (Algo-009)

Inisialisasi namaFile;

saat tombol delete ditekan

Menu ok dan cancel muncul

If (tombol = 'ok')

Delete from dokumen where namaFile="$nama"*;*

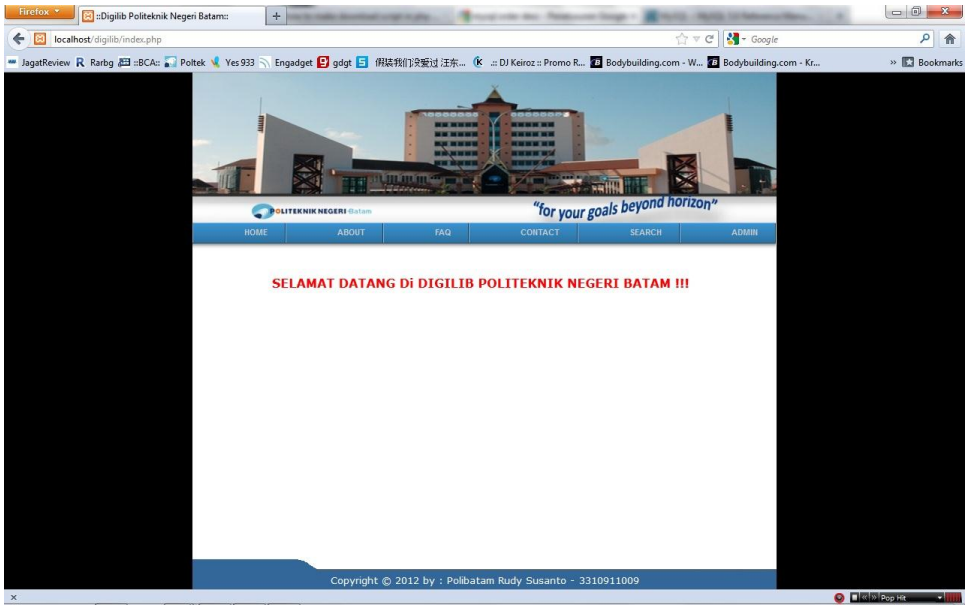
Else

Kembali ke tampilan sebelumnya;

end

III.11 Perancangan Antarmuka

III.11.1 Antarmuka Home Situs digilib.polibatam.ac.id



Gambar 13 Tampilan Antarmuka Home situs digilib.polibatam.ac.id

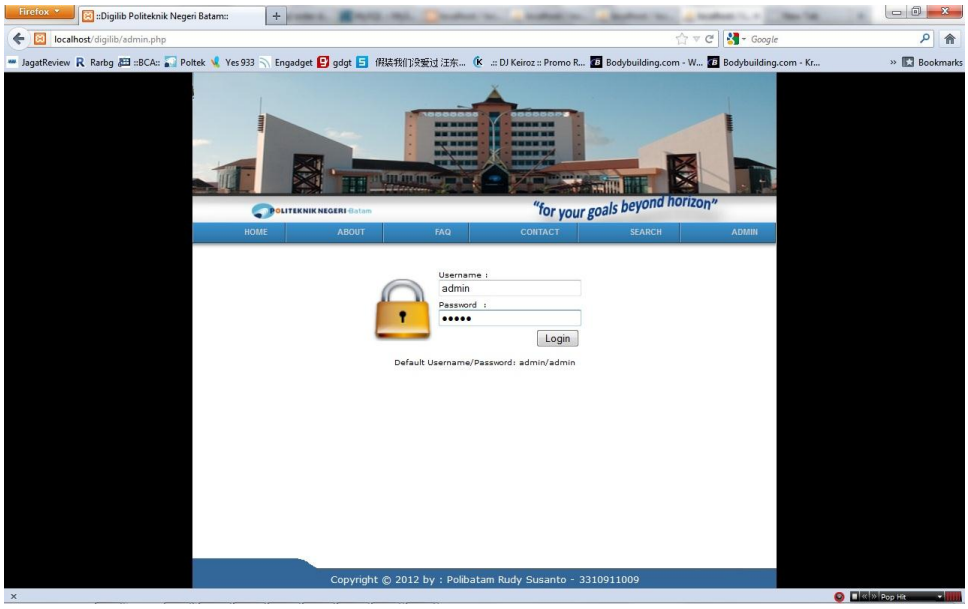
Pada Gambar 13 menunjukkan tampilan utama / tampilan *home* pada saat mengakses situs digilib.polibatam.ac.id. Terdapat beberapa menu yang dapat dipilih yaitu : menu *Home*, menu *About*, menu *FAQ (Frequently Asked Question)*, menu *Contact*, menu *Search*, dan menu *Admin*. Penjelasan perancangan antarmuka akan difokuskan pada menu ***Admin*** dan menu ***Search***.

Tabel 1 Deskripsi Tampilan Antarmuka Home situs digilib.polibatam.ac.id

Id_Objek	Jenis	Keterangan
home.php	*Button*	*Situs home digilib.polibatam.ac.id*
about.php	*Button*	*Tentang informasi situs*
faq.php	*Button*	*Tentang faq situs*
contact.php	*Button*	*Tentang contact situs*
search.php	*Button*	*Cari kata kunci dokumen*

Id_Objek	Jenis	Keterangan
admin.php	*Button*	*Login admin*

III.11.2 Antarmuka Login Administrator



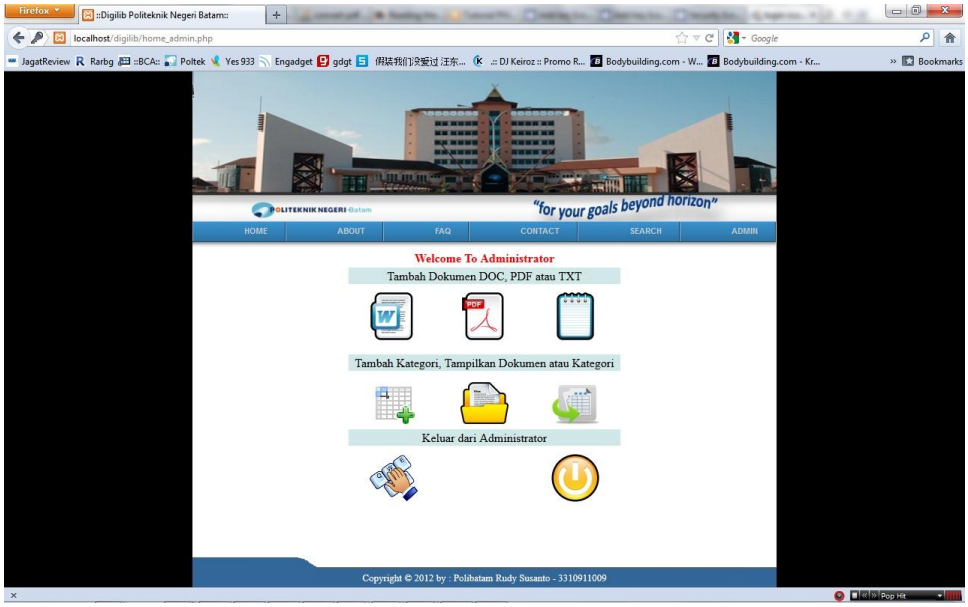
Gambar 14 Tampilan Antarmuka *Login* Admin

Pada Gambar 14 menunjukkan tampilan antarmuka *Login* untuk *admin*. Proses *Login* dilakukan untuk menghindari sistem diakses secara umum oleh pihak yang tidak bertanggungjawab.

Tabel 2 Deskripsi Tampilan Antarmuka *Login* Admin

Id_Objek	Jenis	Keterangan
Username	*Input Text*	*Username*
Password	*Input Text*	*Password*
Login	*Button*	*Tombol Login Admin*

III.11.3 Antarmuka Home Utama Admin



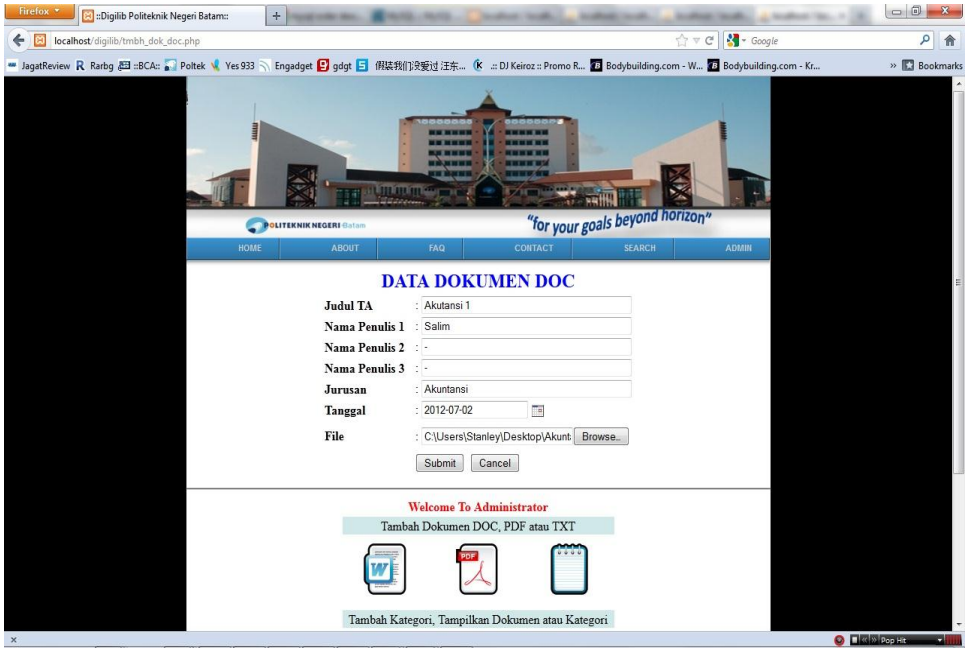
Gambar 15 Tampilan Antarmuka Home Utama Admin

Pada Gambar 15 menunjukkan tampilan utama pada admin setelah proses *login*, didalam tampilan terdapat tombol tambah dokumen (format doc, pdf, txt), tombol tambah kategori, tombol tampil dokumen, tombol tampil kategori, tombol tambah kata kunci dan tombol logout.

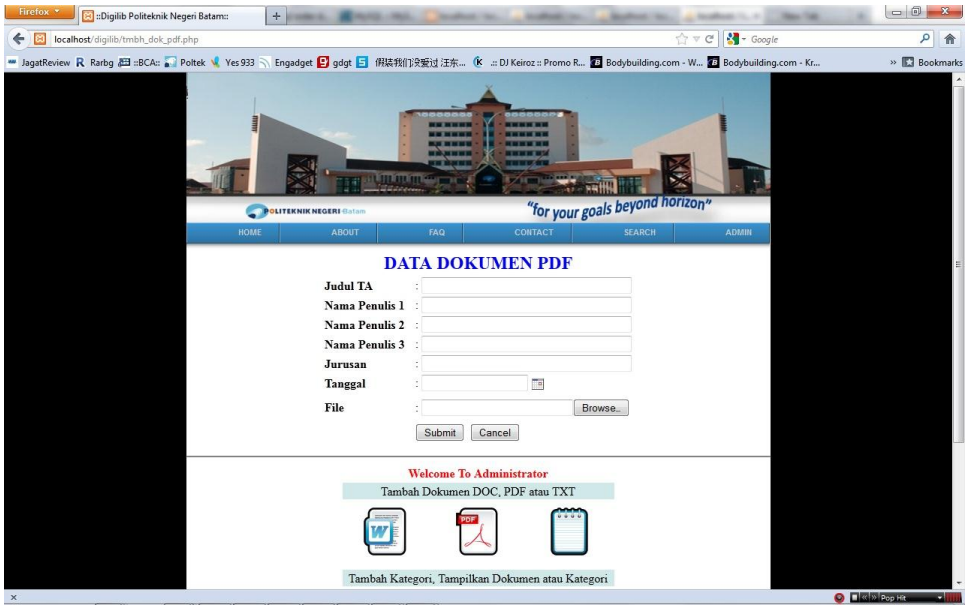
Tabel 3 Deskripsi Tampilan Antarmuka Home Utama Admin

Id_Objek	Jenis	Keterangan
tmbh_dok	*Button*	*Tambah dokumen dok*
tmbh_pdf	*Button*	*Tambah dokumen pdf*
tmbh_txt	*Button*	*Tambah dokumen text*
tmbh_kategori	*Button*	*Tambah kategori*
tmpl_dok	*Button*	*Tampil daftar dokumen*
tmpl_kategori	*Button*	*Tampil daftar kategori*
tmbh_ktkunci	*Button*	*Tambah kata kunci*
logout	*Button*	*Keluar dari Administrator*

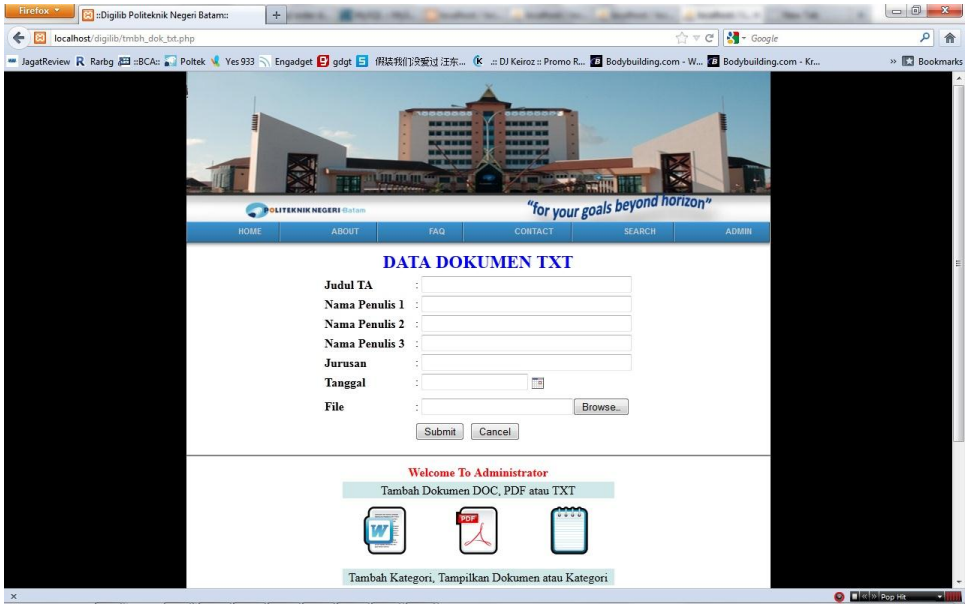
III.11.4 Antarmuka Tambah Dokumen DOC, PDF dan TXT



Gambar 16 Tampilan Antarmuka Tambah Dokumen DOC



Gambar 17 Tampilan Antarmuka Tambah Dokumen PDF



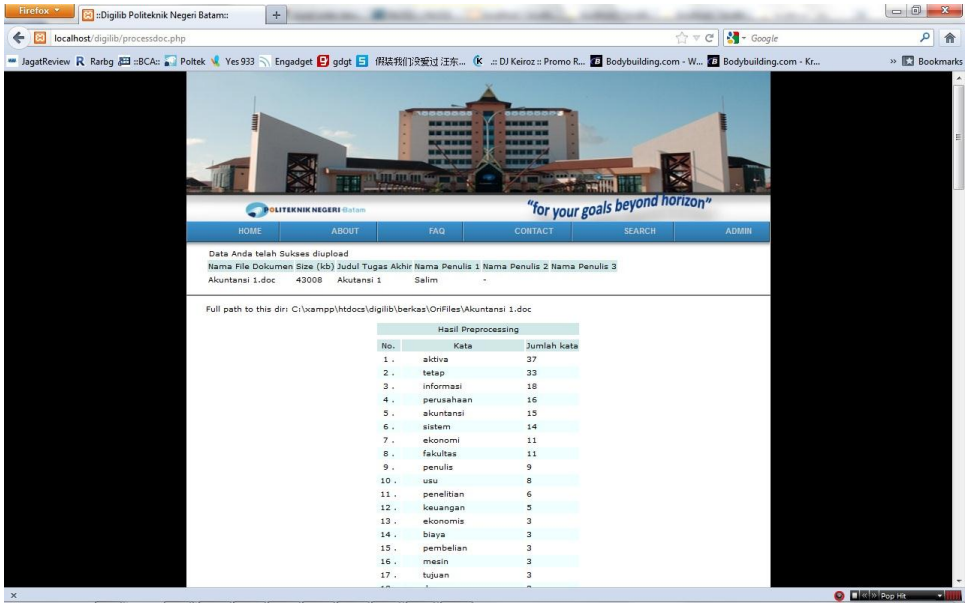
Gambar 18 Tampilan Antarmuka Tambah Dokumen TXT

Pada Gambar 16, 17, dan 18 menunjukkan tampilan masing-masing untuk penambahan dokumen baik format doc, pdf maupun txt. Masing-masing tampilan berisi inputan untuk dokumen dengan judul TA, nama penulis 1, nama penulis 2, nama penulis 3, jurusan, lokasi file, tombol *submit* dan *cancel*.

Tabel 4 Deskripsi Tampilan Antarmuka Tambah Dokumen DOC, PDF dan TXT

Id_Objek	Jenis	Keterangan
namaFile	*Input text*	*Judul Tugas Akhir*
Penulis1	*Input text*	*Nama penulis1*
Penulis2	*Input Text*	*Nama penulis2*
Penulis3	*Input Text*	*Nama penulis3*
jurusan	*Input text*	*Jurusan*
tanggal	*Combo Box*	*Tanggal Dokumen*
browse	*Button*	*Tombol explore file dokumen*
submit	*Button*	*Tombol Submit Tambah Dokumen*
cancel	*Button*	*Tombol Batal Tambah Dokumen*

III.11.5 Antarmuka Informasi dan Tabel Hasil *Preprocessing*



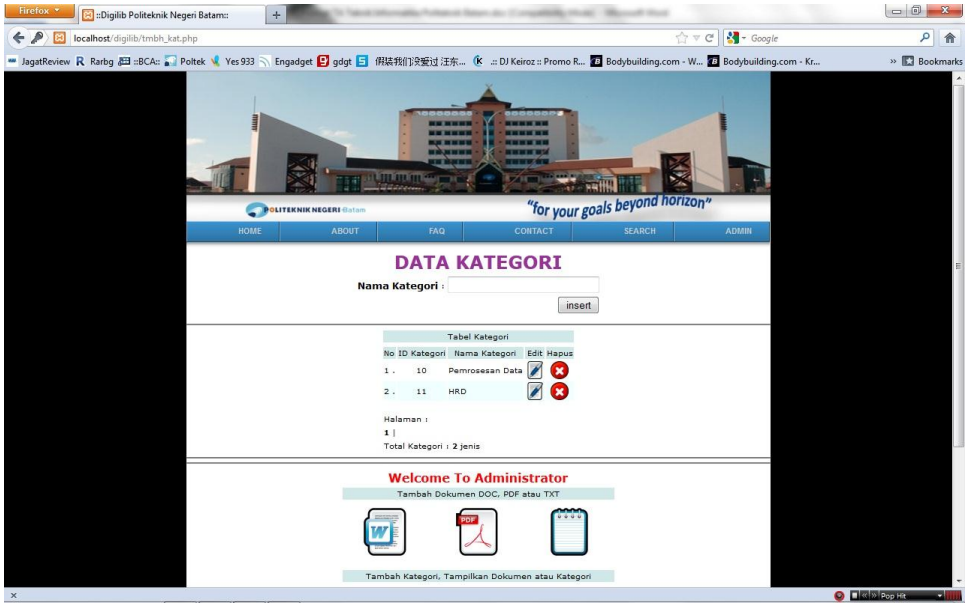
Gambar 19 Tampilan Antarmuka Informasi dan Tabel Hasil *Preprocessing*

Pada Gambar 19 menunjukkan tampilan informasi dokumen yang telah selesai diupload dan menampilkan hasil tabel dari *preprocessing.*

Tabel 5 Deskripsi Tampilan Antarmuka Informasi dan Tabel Hasil *Preprocessing*

Id_Objek	Jenis	Keterangan
tabel_process	*Tabel*	*Tabel Hasil Preprocesing*

III.11.6 Antarmuka Tambah Data Kategori



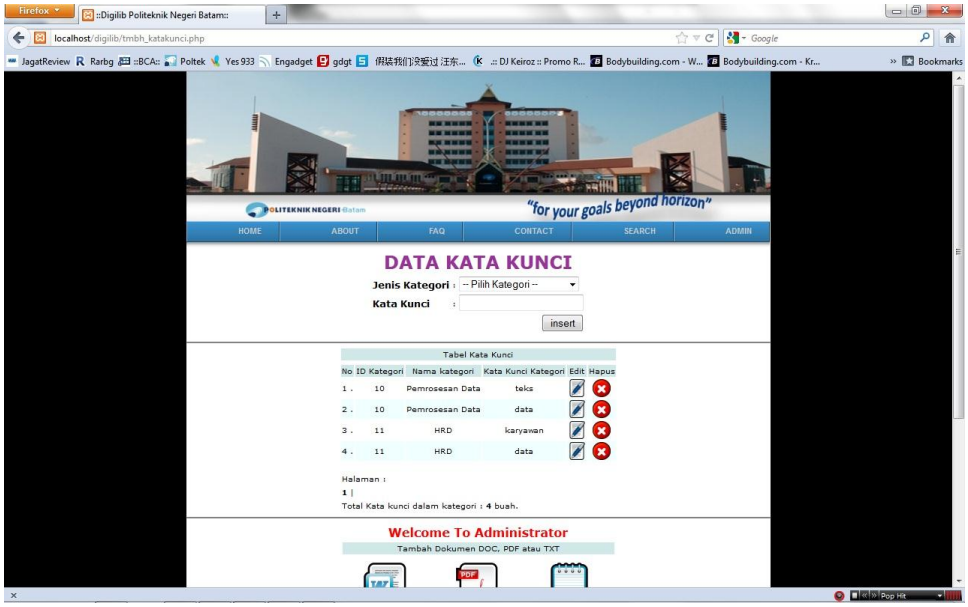
Gambar 20 Tampilan Antarmuka Tambah Data Kategori

Pada Gambar 20 menunjukkan tampilan untuk menambah data kategori pada sistem Text Mining ini dan menampilkan hasil tabel kategori setelah data kategori ditambah.

Tabel 6 Deskripsi Antarmuka Tambah Data Kategori

Id_Objek	Jenis	Keterangan
namaKate	*Input text*	*Nama Kategori*
Insert	*Button*	*Tombol Tambah Data Kategori*
tabel_kate	*Tabel*	*Tabel Data Kategori*
Edit	*Button*	*Edit Data Kategori*
Delete	*Button*	*Hapus Data Kategori*

III.11.7 Antarmuka Tambah Data Kata Kunci



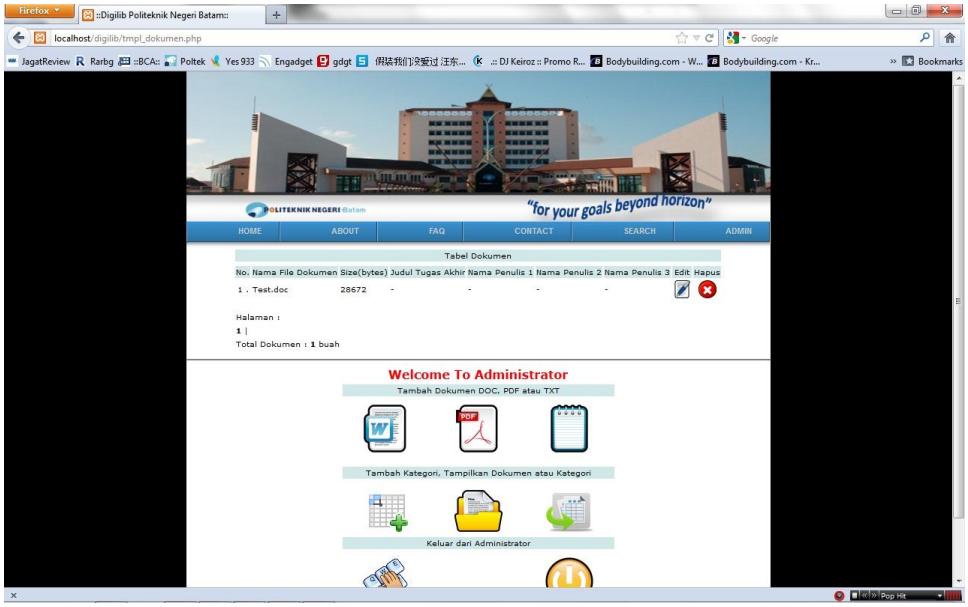
Gambar 21 Tampilan Antarmuka Tambah Data Kata Kunci

Pada Gambar 21 menunjukkan tampilan untuk menambah data kata kunci pada kategori tertentu di sistem Text Mining ini dan menampilkan hasil tabel kategori dan kata kunci setelah data kata kunci ditambah.

Tabel 7 Deskripsi Tampilan Antarmuka Tambah Data Kata Kunci

Id_Objek	Jenis	Keterangan
nmKate	*Input text*	*Nama Kategori*
ktKunci	*Input text*	*Kata Kunci*
Insert	*Button*	*Tombol Tambah Data Kata Kunci*
tabel_ktKunci	*Tabel*	*Tabel Data Kata Kunci*
Edit	*Button*	*Edit Data Kata Kunci*
Delete	*Button*	*Hapus Data Kata Kunci*

III.11.8 Antarmuka Tampil Tabel Dokumen



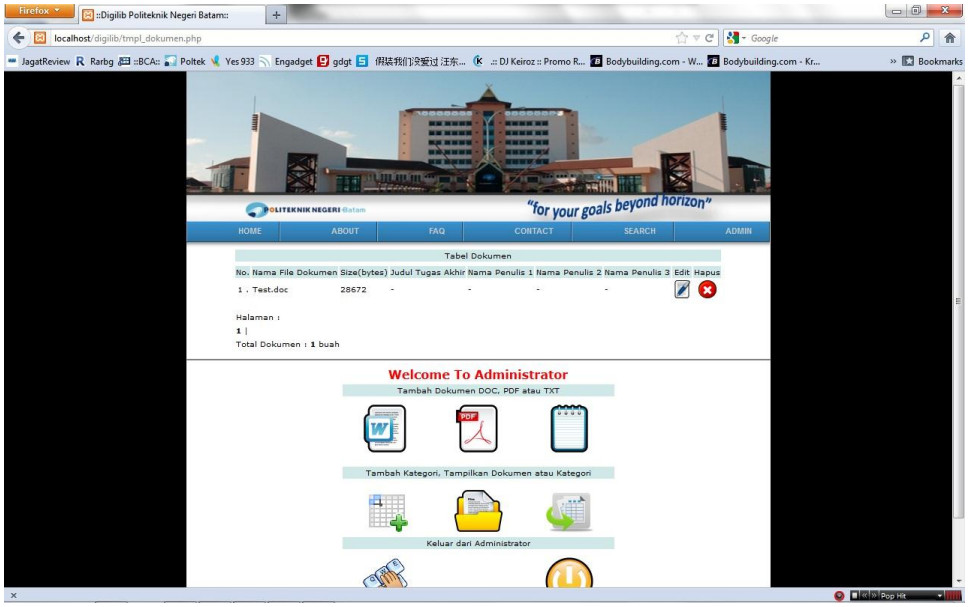
Gambar 22 Tampilan Antarmuka Tampil Tabel Dokumen

Pada Gambar 22 menunjukkan tampilan untuk tampil tabel data dokumen yang telah diupload pada sistem Text Mining ini dan dapat mengedit data dokumen dan menghapus dokumen yang telah diupload.

Tabel 8 Deskripsi Tampilan Antarmuka Tampil Tabel Dokumen

Id_Objek	Jenis	Keterangan
tabel_dokumen	*Tabel*	*Tabel Data Dokumen*
Edit	*Button*	*Edit Data Dokumen*
Delete	*Button*	*Hapus Dokumen*

III.11.9 Antarmuka Tampil Tabel Kategori



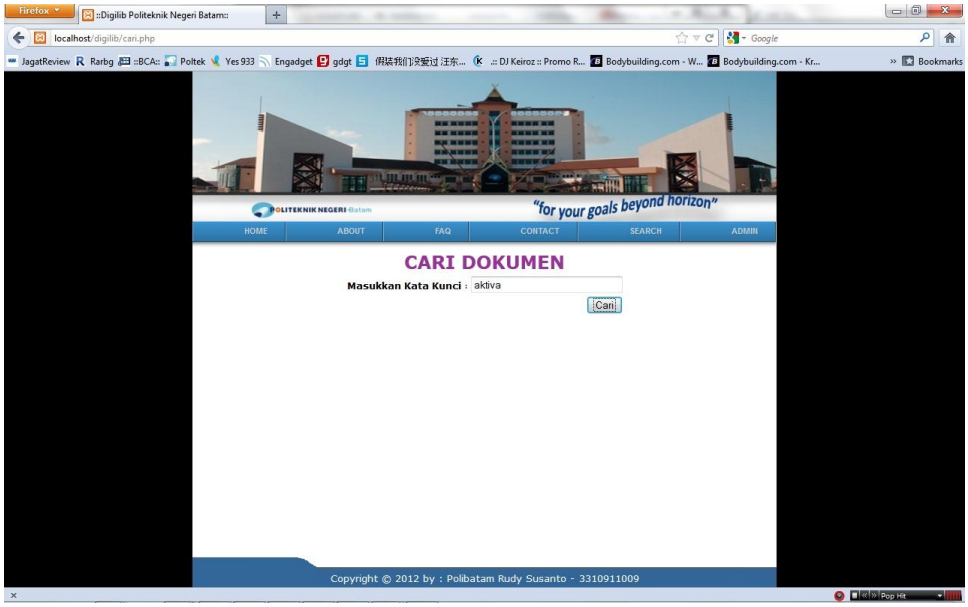
Gambar 23 Tampilan Antarmuka Tampil Tabel Kategori

Pada Gambar 23 menunjukkan tampilan untuk tampil tabel data kategori yang telah ditambah ke dalam sistem Text Mining.

Tabel 9 Deskripsi Tampilan Antarmuka Tambah Data Kata Kunci

Id_Objek	Jenis	Keterangan
tabel_kategori	*Tabel*	*Tabel Data Kategori*
Edit	*Button*	*Edit Data Kategori*
Delete	*Button*	*Hapus Kategori*

III.11.10 Antarmuka *User* Mencari Dokumen



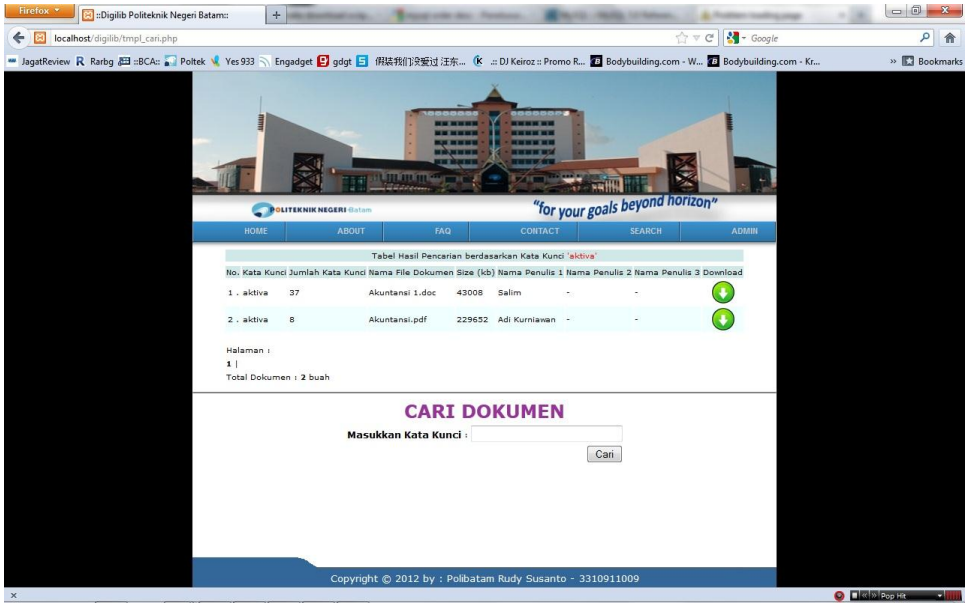
Gambar 24 Tampilan Antarmuka *User* Mencari Dokumen

Pada Gambar 24 menunjukkan tampilan bagi *user* mencari dokumen dengan memasukkan kata kunci ke dalam tempat yang telah disediakan kemudian dilanjutkan dengan menekan tombol cari.

Tabel 10 Deskripsi Tampilan Antarmuka *User* Mencari Dokumen

Id_Objek	Jenis	Keterangan
ktKunci	*Input text*	*Input Kata Kunci*
Cari	*Button*	*Cari Dokumen*

III.11.11 Antarmuka *User* Hasil Pencarian Dokumen



Gambar 25 Tampilan Antarmuka *User* Hasil Pencarian Dokumen

Pada Gambar 25 menunjukkan tampilan bagi *user* berupa tabel hasil pencarian dokumen berdasarkan kata kunci yang dicari dan tombol *download* untuk meng*download* dokumen yang diinginkan.

Tabel 11 Deskripsi Tampilan Antarmuka *User* Hasil Pencarian Dokumen

Id_Objek	Jenis	Keterangan
tabel_hasilcari	*Table*	*Tabel hasil pencarian Dokumen berdasarkan kata kunci*
download	*Button*	*Download Dokumen dari tabel*

Bab IV Hasil dan Pembahasan

IV.1 Implementasi Kelas

Tabel 12 Implementasi Kelas

No	***Nama Kelas***	***Nama File***
1	*Upload*	*tmbh_dok_doc.php* *tmbh_dok_pdf.php* *tmbh_dok_txt.php*
2	*Preprocessing*	*processdoc.php*
3	*Processing*	*processpdf.php*
4	*Kategorisasi*	*processtxt.php* *tmpl_dokumen.php* *tmpl_kate.php* *tampil_dokumen.php* *tampil_kategori.php*
5	*Cari*	*cari.php* *cari_dok.php* *isi_cari.php* *tmpl_cari.php*

IV.2 Hasil Pengujian

Tabel 13 Hasil Pengujian

Penguji : Mir'atul Khusna Mufida

No	*Nama Proses*	*Data*	*Hasil yang diharapkan*	*Status*
1	*Upload*	*Input data File, upload file dengan nama Akuntansi 1.doc*	*File Tersimpan ke folder server, data file tersimpan ke database dan menampilkan hasil tabel data yang telah tersimpan*	
2	*Tambah Dokumen, Tambah Kategori dan Tambah Kata Kunci Kategori*	*Tambah data Dokumen Akuntansi 1.doc, Tambah Jenis Kategori "Akuntansi", Tambah Kata Kunci "Aktiva" ke Kategori "Akuntansi"*	*Data dokumen, kategori, dan kata kunci dapat tersimpan ke database dan ditampilkan dalam bentuk tabel dokumen dan kategori di Adm GUI*	
2	*Token*	*simbol-simbol telah dihilangkan*	*Data dilakukan proses token, filter, stem dan melakukan perhitungan term frequency kemudian disimpan, ditampilkan pada Adm GUI.*	
3	*Filter*	*Kata-kata tidak penting telah dihilangkan*		
4	*Stem*	*Kata-kata berimbuhan telah menjadi*		

		kata dasar		
5	*Edit*	*Data Dokumen dan Data Kategori*	*Data dokumen, data kategori dapat diedit apabila ada update informasi tambahan.*	
6	*Hapus*	*Daftar Dokumen dan Daftar Kategori*	*Data yang berada pada daftar dokumen dan kategori telah terhapus.*	
5	*Cari*	*Masukkan kata kunci yang ingin dicari, misal "aktiva"*	*Dokumen dengan kata kunci "aktiva" dicari dan daftar dokumen ditampilkan pada User GUI*	

BAB V Kesimpulan dan Saran

V.1 Kesimpulan

Dari proses implementasi *text mining* pada digilib.polibatam.ac.id dapat diambil kesimpulan, yaitu:

1. Sistem dapat memfilter sebuah dokumen secara otomatis.

2. Sistem dapat melakukan kategorisasi data dengan parameter kata kunci

3. Sistem dapat melakukan pencarian terhadap dokumen sesuai dengan kata kunci pada situs *digilib.polibatam.ac.id*.

V.2 Saran

Pemrosesan *text mining* dengan Bahasa Indonesia dapat dikembangkan dengan pemrosesan dokumen berganda / *multiple file* dan metode *Vector Space Model*.

DAFTAR PUSTAKA

[1] http://www.scribd.com/doc/57180813/Penambangan-Teks-Text-Mining
Diakses pada tanggal 12 Oktober 2011.

[2] http://www.eepis-its.edu/uploadta/downloadmk.php?id=997
Diakses pada tanggal 12 Oktober 2011.

[3] http://lecturer.eepis-its.edu/~iwanarif/kuliah/dm/6Text%20Mining.pdf
Diakses pada tanggal 15 Oktober 2011.

[4] http://journal.uii.ac.id/index.php/Snarti/article/viewFile/1547/1323
Diakses pada tanggal 15 Oktober 2011.

[5] http://telkomnika.ee.uad.ac.id/n9/files/Vol.8No.1Apr10/8.1.4.10.06.pdf
Diakses pada tanggal 15 Oktober 2011.

[6] http://lecturer.ukdw.ac.id/budsus/pdf/textwebmining/TextMining_Kuliah.pdf
Diakses pada tanggal 18 Oktober 2011.

[7] http://repository.upi.edu/operator/upload/s_d545_060696_chapter2.pdf
Diakses pada tanggal 21 Oktober 2011.

[8] http://papers.gunadarma.ac.id/index.php/industry/article/viewFile/834/794
Diakses pada tanggal 21 Oktober 2011.

[9] http://journal.uii.ac.id/index.php/Snarti/article/viewFile/1268/1076
Diakses pada tanggal 21 Oktober 2011.

[10] http://www.bataviase.co.id
Diakses pada tanggal 24 November 2011.

[11] http://www.sitti.co.id/tentang-sitti.html
Diakses pada tanggal 26 November 2011.

[12] http://clearforest.com/solutions.html
Diakses pada tanggal 28 November 2011.

[13] http://www.ibm.com/infosphere/warehouse/
Diakses pada tanggal 28 November 2011.

[14] http://www.megaputer.com/textanalyst.php
Diakses pada tanggal 28 November 2011.

[15] http://www.sas.com/text-analytics/
Diakses pada tanggal 28 November 2011.
[16] http://www.spss.com/text_mining_for_clementine/
Diakses pada tanggal 28 November 2011.
[17] http://www.statsoft.com/products/statistica-text-miner/
Diakses pada tanggal 28 November 2011.
[18] http://www.thevintagepoint.com/
Diakses pada tanggal 28 November 2011.
[19] http://www.provalisresearch.com/wordstat/wordstat.html
Diakses pada tanggal 28 November 2011.
[20] http://gate.ac.uk/
Diakses pada tanggal 28 November 2011.
[21] http://alias-i.com/lingpipe/
Diakses pada tanggal 28 November 2011.
[22] http://www.cs.uic.edu/~liub/LPU/LPU-download.html
Diakses pada tanggal 28 November 2011.
[23] http://www.rapidminer.com/
Diakses pada tanggal 28 November 2011.
[24] http://uima.apache.org/
Diakses pada tanggal 28 November 2011.
[25] http://www.ranks.nl/
Diakses pada tanggal 28 November 2011.
[26] http://www.wordle.net/
Diakses pada tanggal 28 November 2011.