Rancang Bangun Fitur Pencarian Topik Penelitian dengan Metode TF-IDF (Kasus: *Website* Grup Riset I-Syis)

Adhistya Erna Permanasari, Hirzi Chandani, Silmi Fauziati
Departemen Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada
Jl. Grafika no. 2, 55281, Yogyakarta, Indonesia
adhistya@ugm.ac.id

Abstract— **Nowadays, information and communication technology has been widely applied in various institutions of higher education. Department of Electrical Engineering and Information Technology UGM has various research groups, including Intelligent Systems (I-Syis) research group. The I-Syis research group has a website that provide information relate to the group. However currently, it is not complemented with feature of lecturer based on research topic keywords. This research aims to develop such feature that able to generate lecturer research topic. The existing keywords are obtained from the extraction process of the research title of each lecturer. The process of keyword extraction is divided into two stages: preprocessing and term weighting. We used Term Frequency - Inverse Document Frequency (TF-IDF) method for term weighting stage. The keyword will be entered into the database for later use as the backend of the web page search feature. The output of this research is a search page which present the corresponding lecturers name of the related keywords.**

Keywords-term weighting; TF-IDF; searching feature

Intisari— **Pemanfaatan teknologi informasi dan komunikasi sudah banyak diaplikasikan di berbagai institusi pendidikan tinggi. Departemen Teknik Elektro dan Teknologi Informasi UGM memiliki berbagai grup riset, di antaranya grup riset Intelligent Systems (I-Syis). Grup riset I-Syis memiliki *website* yang berisi seputar informasi mengenai grup riset tersebut. Akan tetapi saat ini *website* tersebut belum memiliki fitur pencarian dosen berdasarkan kata kunci topik penelitian. Penelitian ini bertujuan untuk mengembangkan fitur pencarian dosen berdasarkan kata kunci topik penelitian untuk website grup riset Intelligent Systems DTETI UGM. Kata kunci yang ada diperoleh dari proses ekstraksi dari judul penelitian setiap dosen. Proses ekstraksi kata kunci dibagi menjadi dua tahapan yaitu tahap preprocessing dan tahap pembobotan kata. Untuk metode pembobotan kata, dalam penelitian ini digunakan metode Term Frequency – Inverse Document Frequency (TF-IDF). Hasil kata kunci yang diperoleh akan dimasukkan ke dalam basis data untuk nantinya digunakan sebagai backend halaman web fitur pencarian. Keluaran dari penelitian ini adalah sebuah halaman pencarian yang akan menampilkan nama dosen yang terkait dari kata kunci yang dimasukkan.**

Kata Kunci- pembobotan kata, TF-IDF, fitur pencarian

I. PENDAHULUAN

Penggunaan teknologi informasi dan komunikasi (TIK) di dunia pendidikan sudah menjadi kebutuhan mutlak dan banyak digunakan oleh institusi pendidikan tinggi jika ingin kualitas pendidikan meningkat. Pada institusi-institusi pendidikan tinggi, penggunaan teknologi informasi tidak hanya sebatas pada pendukung manajemen saja, tetapi juga digunakan untuk peningkatan pada proses pengambilan keputusan (*decision-making*) dibanyak tingkatan dalam manajemen. Penggunaan teknologi informasi dan komunikasi yang efektif di dalam dunia pendidikan akan terealisasi jika didukung oleh pengembangan manajemen sistem informasi yang efektif [1]. Saat ini pemanfaatan teknologi informasi dan komunikasi di dunia pendidikan sudah banyak diaplikasikan di berbagai institusi-institusi pendidikan tinggi di Indonesia. Oleh karena itu diperlukan pemasyarakatan dan implementasi yang tepat agar pelaksanaan dan pemanfaatannya optimal sesuai dengan kepentingan dan sasaran dunia pendidikan.

Departemen Teknik Elektro dan Teknologi Informasi (DTETI) UGM memiliki beberapa grup riset, salah satunya adalah grup riset *Intelligent Systems* (I-Sys). Grup I-Sys memfokuskan diri pada teori dan aplikasi sistem yang dapat mengetahui, merasakan, mempelajari, dan melakukan aksi dengan cerdas. Aplikasi yang dikembangkan oleh Grup I-Sys ini sebagian besar berupa topik *signal and image processing*, *pattern recognition*, dan *data mining*. Tujuan dari grup riset ini adalah untuk menciptakan sistem pintar di berbagai bidang seperti kelistrikan, sistem komputer dan elektronis, *biomedicine*, bisnis, *natural language processing* dan berbagai bidang lainnya [2]. Grup riset Intelligent Systems ini memiliki *website* yang berfungsi untuk menampilkan informasi terkait dengan grup riset tersebut. *Website* tersebut dapat diakses pada *http://ai.te.ugm.ac.id* dan *http://i-system.ft.ugm.ac.id*. Informasi-informasi yang ditampilkan pada website tersebut, termasuk cakupan penelitian, dosen yang tergabung. Data terkait topik penelitian dosen diperlukan mahasiswa untuk disesuaikan dengan arah riset mahasiswa baik tingkat Sarjana, Master, ataupun Doktor. Saat ini belum ada fitur pencarian pada situs yang ada mempersulit pembaca untuk mencari bidang keahlian dosen yang tergabung di Grup I-Sys.

Penelitian ini mengembangkan fungsi untuk melakukan pencarian dosen yang cocok dengan kata kunci topik penelitian yang dimasukkan. Data mengenai bidang-bidang keahlian dosen yang tergabung dalam grup riset *Intelligent Systems* diambil dari Google Scholar. Data tersebut akan diolah dan ditampilkan dalam bentuk *web*. Data Google Scholar tersebut akan dilakukan ekstraksi untuk mendapatkan kata dan frasa kunci dari keahlian dosen. Ekstraksi kata kunci menggunakan metode pembobotan *Term Frequency* –

Inverse Document Frequency (TF-IDF). Metode pembobotan TF-IDF dipilih sebagai metode ekstraksi kata kunci karena metode pembobotan TF-IDF merupakan salah satu metode pembobotan kata yang cukup dikenal dalam riset *text mining* [3]. Alasan selanjutnya yaitu metode pembobotan TF-IDF memiliki akurasi yang menjanjikan karena TF-IDF menentukan bobot tiap kata menggunakan dua pendekatan, frekuensi kemunculan kata dan seberapa banyak dokumen yang mengandung kata tersebut [3]. Keluaran dari penelitian ini adalah fitur sistem pencarian yang akan menampilkan daftar dosen-dosen berdasarkan kata kunci topik penelitian yang dimasukkan.

II. TINJAUAN PUSTAKA

A. Ekstraksi Kata Kunci (Keywords)

Banyak jurnal atau artikel yang menampilkan daftar kata kunci (*keywords*). *Keywords* memiliki beberapa tujuan. Sebagai contoh antara lain, (1) Ketika *keywords* dicetak di halaman pertama sebuah artikel/jurnal, maka tujuan dari *keywords* tersebut adalah untuk peringkasan. Sehingga pembaca dapat dengan cepat mengetahui apakah jurnal/artikel tersebut sesuai dengan ketertarikan pembaca. (2) Disaat *keywords* dicetak pada indeks kumulatif sebuah jurnal, maka tujuan *keywords* tersebut adalah pengindeksan. Sehingga pembaca dapat dengan cepat mencari artikel yang relevan ketika pembaca memiliki kebutuhan spesifik terhadap apa yang ia cari. (3) Saat sebuah mesin pencari memiliki *field* yang diberi label *keywords*, maka tujuan dari *keywords* adalah untuk membuat pembaca dapat melakukan pencarian yang lebih presisi. *Keywords* dapat melayani berbagai tujuan seperti yang telah disebutkan, karena tujuan-tujuan tersebut memiliki maksud yang sama yakni kebutuhan untuk daftar frasa-frasa yang menangkap maksud utama dari sebuah dokumen. Jadi *keywords list* atau daftar kata kunci dapat didefinisikan sebagai daftar kata yang menangkap topik utama pada dokumen yang diberikan [4].

Pendekatan awal untuk ekstraksi kata kunci otomatis adalah fokus pada evaluasi statistik korpus tiap kata. Jones [5] dan Salton et al. [6] mendeskripsikan hasil positif dalam pemilihan indeks perbendaharaan kata diskriminatif secara statistik yang ada di dalam korpus. Kemudian riset ekstraksi kata kunci mengaplikasikan ukuran ini untuk memilih kata diskriminatif sebagai kata kunci untuk suatu dokumen. Sebagai contoh, Andrade dan Valencia [7] mendasari pendekatan mereka pada perbandingan frekuensi distribusi kata dalam sebuah teks dengan distribusi dalam korpus. Meskipun beberapa kata kunci harus dievaluasi karena secara statistik mendiskriminasi korpus, ada pula beberapa kata kunci dalam korpus yang tidak mendiskriminasi secara statistik. Metode *corpus-oriented* juga bekerja hanya pada satu kata. Batas pengukuran lebih jauh untuk kata diskriminatif secara statistik karena satu kata sering dipakai diberbagai konteks yang berbeda.

Turney [4] mendefinisikan lebih lanjut mengenai ekstraksi kata kunci otomatis sebagai pemilihan berbagai kata/frasa penting dan berkaitan dengan topik suatu dokumen. Turney pun membandingkan performa kata kunci yang dibuat oleh mesin dan kata kunci yang dibuat oleh manusia. Pada dokumen yang ia gunakan, rata-rata 75% kata kunci penulis terdapat diseluruh dokumen. Oleh karena itu, algoritme ekstraksi kata kunci yang ideal harus bisa membuat kata kunci yang cocok dengan 75% kata kunci penulis tadi.

B. Text Mining dengan Metode TF-IDF

Text mining adalah konsep untuk mengeksplorasi suatu informasi pada data yang berbentuk teks dari sekumpulan dokumen atau kalimat. *Text mining* merupakan cabang ilmu *data mining* yang menggunakan pola data dari bahasa alami (*natural language*). Proses dari *text mining* yaitu ekstraksi fitur dan mengolah fitur-fitur tersebut untuk membentuk suatu informasi, fakta dan hipotesis baru [8].

Aulia Hakim dkk. [3] telah melakukan *text mining* berupa klasifikasi dokumen otomatis untuk artikel berita berbahasa Indonesia dengan pendekatan TF-IDF. Dalam penelitian tersebut, tahapan pengerjaan dibagi menjadi menjadi dua yaitu tahap *preprocessing* dan tahap *processing*. Tahap *preprocessing* bertujuan untuk membuat kamus kata dan bobot kata. Tahap processing bertujuan untuk mengkategorikan artikel berdasarkan topik artikel tersebut. Dalam tahap *preprocessing*, Aulia Hakim dkk. menggunakan 8 tahap untuk membuat kamus kata dan bobot kata. Tahapan tersebut yaitu tokenisasi, *bi-gram creation*, penghapusan duplikasi, penghapusan *stop word*, filterisasi frekuensi kata, *supervised word removal*, implementasi TF-IDF dan normalisasi bobot TF-IDF.

III. METODE

A. Metode Pembobotan Term Frequency-Inverse Document Frequency (TF-IDF)

Semua sistem berbasis teks membutuhkan beberapa representasi dokumen, dan representasi yang baik bergantung pada tugas yang akan dikerjakan [9]. Selain itu, kemampuan untuk melakukan tugas klasifikasi secara akurat bergantung pada representasi dokumen yang akan diklasifikasikan [10]. Berbeda dengan *data mining* yang mengelola data terstruktur, *text mining* harus mengelola sekumpulan dokumen semi terstruktur bahkan tidak terstruktur. Hal ini menyebabkan salah satu tema utama yang mendukung *text mining* adalah transformasi teks ke vektor numeris, contoh: *text representation*.

Dalam penerimaan informasi, dokumen umumnya diidentifikasikan berdasarkan set kata atau kata kunci yang secara kolektif digunakan untuk mereprensentasikan konten mereka. *Vector Space Model* (VSM) adalah salah satu model yang paling banyak digunakan untuk representasi, karena konsepnya yang sederhana dan mendasari metafor penggunaan kedekatan spasial untuk kedekatan semantik. Pada umumnya ada dua jenis pekerjaan dalam representasi teks, yaitu pengindekan dan pembobotan kata [9]. Pengindekan adalah pekerjaan memasukkan indeks untuk dokumen. Pembobotan kata adalah pekerjaan mencari bobot tiap kata, yang

ukurannya menentukan tingkat kepentingannya dalam suatu dokumen.

Saat ini banyak metode yang dipakai untuk pembobotan kata, dimana banyak diturunkan dari berbagai asumsi karakteristik kata dalam teks. Sebagai contoh Inverse Document Frequency (IDF) mengasumsikan tingkat relatif kepentingan suatu kata terhadap dokumen berbanding terbalik dengan frekuensi kemunculan kata.

TF-IDF dikenalkan pertama kali oleh Jones [11] dengan pemikiran bahwa *term query* yang muncul di banyak dokumen bukan merupakan diskriminator yang baik dan harus diberi bobot yang lebih kecil dibandingkan yang muncul lebih sedikit dalam dokumen [12]. Formula TF-IDF yang digunakan untuk pembobotan dapat dirumuskan dalam Persamaan 1 berikut:

$$w_{ij} = tf_{ij} \times log\left(\frac{N}{df_i}\right) \quad (1)$$

dimana wi,j adalah bobot dari kata atau *term* i dalam dokumen j. N adalah jumlah dokumen total, tfi,j adalah *term frequency* atau frekuensi kemunculan kata i dalam dokumen j, dan dfi adalah banyaknya dokumen atau *document frequency* yang terdapat *term* i [16].

Ide dasar dari TF-IDF dari teori permodelan bahasa yaitu *term* dalam dokumen dapat dibagi menjadi dua kategori yakni kata elit dan kata tidak elit [13], contohnya yaitu relevan tidaknya sebuah *term* terhadap dokumen. Lebih jauhnya, tingkat keelitan sebuah *term* dapat dievaluasi dari TF dan IDF serta dalam TF*TDF. Hal tersebut digunakan untuk mengukur tingkat kepentingan sebuah term dalam sekumpulan dokumen. TF-IDF telah dibuktikan dapat mengklasifikasikan artikel berita dalam Bahasa Indonesia dengan akurasi tinggi sebesar 98,3% [3].

Meskipun begitu, ada beberapa kritik terhadap penggunaan TF*IDF dalam representasi teks. Pertama yaitu TF*IDF terlalu ‘ad hoc’ karena tidak secara langsung diturunkan dari sebuah model matematika, walaupun biasanya dijelaskan oleh teori informasi Shanon [14]. Kritik kedua yakni dimensi (ukuran set fitur) dalam TF*IDF untuk data tekstual besarnya sama seperti ukuran *vocabulary* di seluruh *data set*, akibatnya adalah komputasi yang besar dalam proses pembobotan [15].

B. Pengumpulan Dataset Judul Publikasi

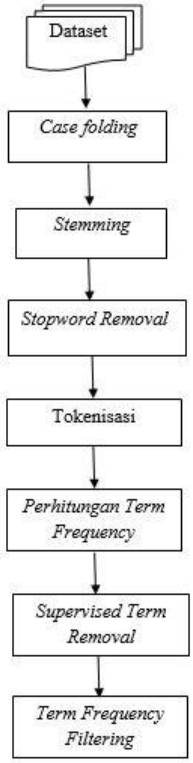
Dataset yang akan digunakan merupakan judul-judul publikasi dosen grup riset Intelligent Systems DTETI yang diambil dari *website* Google Scholar. Pengumpulan dataset melalui Google Scholar dilakukan dengan memasukkan kata kunci nama dosen dan dilakukan proses *screen capture* untuk hasil yang keluar. Total keseluruhan ada 13 dosen grup riset Intelligent Systems yang diambil data judul publikasinya. Hasil pengumpulan dataset dimasukkan ke dalam file CSV. Format file CSV dipilih karena format tersebut merupakan salah satu format yang dapat diproses oleh Python. Hasil *input* ke dalam *file* CSV ditunjukkan dalam Gbr 1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
AEP	Automatic short answer scoring using words overlapping methods													
AEP	Pengembangan data warehouse untuk mendukung Report Pengadaan di instansi pemerintahan													
AEP	Combat aircraft effectiveness assesment using hybrid multi -criteria decision making methodology													
AEP	Toward Development of automated plasmodium detection for Malaria diagnosis in thin blood smear image : An overview													
AEP	A review of missing values handling methods on time - series data													
AEP	The benefit of the web 2.0 technologies in higher education: student's perspectives													
AEP	Comprative study on data mining classification methods for cervical cancer prediction using pap smear results													
AEP	Pattern of accesbility level of health facilities in Yogyakarta													
AEP	Management informantion systems development for veterinary hospital patient registration using first in first out algorithm													
AEP	ARIMA implementation to predict the ammount of antiseptic medicine usage in vetenary hospital													
AEP	I Forex trend prediction technique using multiple indicators and multiple pairs correlations DSS : A software design													
AEP	Rancang bangun cloud data center dalam upaya meningkatkan kualitas sekolah di SMKN 2 sewon													
AEP	A review of an information extraction technique approach for automatic short answer grading													
AEP	Improvement in learning coordinate system using metacognitive strategy path for learning in classroom and intelegent tutoring system													
AEP	Design adaptive learning systems using metacognitive strategy path for learning in classroom and intelligent tutoring systems													
AEP	Cosine similarity to determine similarity measure : study case in online essay assesment													
AEP	Student's Metacognitive Modelling Using Radial Basis Function Network to Support Adaptive Learning													

Gambar 1. Hasil Input File CSV

C. Preprocessing

Preprocessing adalah tahap persiapan data teks sebelum diklasifikasikan. Tujuan dari *preprocessing* adalah untuk meningkatkan kualitas dari teks dengan menghilangkan bagian-bagian yang tidak diperlukan. Tahap *preprocessing* dalam penelitian ini ditunjukkan dalam Gbr 2.



Gambar 2. Tahap *Preprocessing* Teks

Tahap pertama yaitu *case folding* adalah tahap normalisasi teks dengan mengubah semua huruf kapital menjadi huruf kecil. Tujuannya adalah untuk menyamakan fitur. Tahap selanjutnya yaitu *stemming* yang merupakan tahap mengubah suatu kata menjadi kata dasarnya. *Stemming* dilakukan berdasarkan dari bahasa yang digunakan. Dalam penelitian ini, bahasa yang digunakan yaitu Bahasa Indonesia. Pustaka yang digunakan untuk *stemming* Bahasa Indonesia yaitu pustaka Sastrawi.

Tahap selanjutnya yaitu *stopword removal* dimana kata-kata yang sifatnya umum dan tidak mempunyai makna dalam suatu dokumen akan dihapus. Penghapusan *stopword* dilakukan berdasarkan kamus *stopword* Bahasa Indonesia. Kamus *stopword* yang digunakan adalah

stopword Tala [3]. Setelah dilakukan penghapusan *stopword*, tahap selanjutnya dilakukan tokenisasi. Tokenisasi adalah tahap membagi dokumen ke dalam token-token tertentu. Tujuan dari tahap ini adalah untuk menyederhanakan proses kalkulasi dan komputasi di tahap selanjutnya.

Setelah dilakukan tokenisasi, tahap selanjutnya yaitu perhitungan *term frequency*. Tiap-tiap *term* akan dihitung frekuensi kemunculannya dan diurutkan dari yang paling besar. Perhitungan *term frequency* dilakukan menggunakan Python. Hasil dari tahap ini aka dibawa ke tahap *Supervised Term Removal*. Tahap *Supervised Term Removal* adalah tahap pengecekan secara manual untuk menghapus *term* yang *invalid* atau *term* yang tidak tepat dari hasil tahapan sebelumnya. Proses *Supervised Term Removal* dilakukan menggunakan *text editor* Atom. Tahap terakhir dari *preprocessing* yaitu *Term Frequency Filtering*. Tahap ini merupakan tahap mereduksi hasil tahap *Supervised Term Removal*. Dimensi dataset yang dihasilkan dari tahapan sebelumnya cukup besar, oleh karena itu direduksi dan diambil 75 *term* dengan frekuensi tertinggi untuk dibawa ke pembobotan *term*.

D. Pembobotan Kata

Metode pembobotan kata merupakan proses memperhitungkan hubungan antar kata, pengaruh kata terhadap dokumen serta penting tidaknya kata pada dokumen. Penelitian ini menerapkan metode pembobotan kata untuk mendapatkan kata kunci atau *keyword* yang tepat dari data hasil *preprocessing* sebelumnya. Metode pembobotan kata yang digunakan dalam penelitian ini adalah *Term Frequency Inverse Document Frequency* (TF-IDF). Metode TF-IDF dipilih karena metode ini merupakan salah satu metode pembobotan kata yang paling dikenal dalam riset *text mining*. Alasan selanjutnya adalah karena metode TF-IDF memiliki akurasi yang menjanjikan, karena metode TF-IDF menggunakan dua pendekatan untuk menentukan bobot tiap kata, yaitu frekuensi kemunculan kata dan berapa banyak dokumen yang mengandung kata tersebut. Pengerjaan pembobotan kata dalam penelitian ini dilakukan secara manual menggunakan perangkat lunak Microsoft Excel.

E. Pembuatan Basis Data dan Halaman Web

Pembuatan basis data dalam penelitian ini dilakukan menggunakan basis data MySQL versi 5.5.39 yang sudah tercakup dalam *package* perangkat lunak XAMPP versi 3.2.1. Pembuatan basis data dilakukan di *localhost* menggunakan antarmuka GUI PhpMyAdmin versi 4.2.7.1. Data hasil pengolahan pembobotan kata akan dimasukkan ke dalam basis data tersebut.

Terakhir adalah tahap pengembangan *web*. Pembuatan halaman *web* pencarian terbagi menjadi dua proses yaitu pengembangan *frontend* dan pengembangan *backend*. Pengembangan *frontend* bertujuan untuk membuat sebuah antarmuka agar *user* dapat memasukkan kata kunci untuk diproses dan menampilkan hasil dari input kata kunci tersebut. Pengembangan *frontend* menggunakan HTML (*Hypertext Markup Language*) dan CSS (*Cascading Style Sheet).* Karena tidak menjadi fokus utama, desain antarmuka halaman *web* pencarian pada penelitian ini tergolong sederhana dan dapat menampilkan hasil dengan baik. Pengembangan *frontend* dilakukan menggunakan perangkat lunak *text editor* Atom dan XAMPP. Text editor Atom digunakan untuk menulis kode HTML dan CSS. XAMPP digunakan untuk mengaktifkan Apache Web Server pada *localhost* sehingga pengembangan *frontend* dapat dilakukan pada komputer lokal. Setelah proses pengembangan *frontend*, selanjutnya adalah pengembangan *backend*. Pengembangan *backend* untuk halaman *web* pencarian pada penelitian ini bertujuan untuk memproses kata kunci yang dimasukkan pengguna melalui *frontend* halaman *web*. Untuk itu perlu mengaitkan halaman *web* dengan basis data yang telah dibuat sebelumnya. Setelah basis data dikaitkan, proses selanjutnya yaitu menulis kode untuk melakukan *query* ke basis data tersebut. Dari proses *query* inilah, MySQL dapat mengembalikan hasil untuk ditampilkan ke *frontend* halaman *web*. Pengembangan *backend* dilakukan dengan bahasa pemrograman PHP. Perangkat lunak yang digunakan adalah *text editor* Atom dan XAMPP. *Text editor* Atom digunakan untuk menulis kode PHP dan XAMPP digunakan untuk mengaktifkan Apache Web Server pada *localhost*. Pengembangan *backend* dilakukan pada komputer lokal.

F. Penyematan dan Uji Fungsionalitas Web

Proses pengembangan halaman *web* pencarian dilakukan pada komputer lokal. Oleh karena itu perlu melakukan *deploy* halaman web tersebut ke internet untuk menyematkannya pada *web* grup riset *Intelligent System*. *Website* grup riset *Intelligent System* dibangun pada *platform* Wordpress. Karena itu tidak dapat dilakukan *deploy* langsung pada layanan *hosting* bawaan *website* tersebut. Jadi untuk menyematkan halaman *web* pencarian ke *website Intelligent System* perlu dilakukan *deploy* ke layanan *hosting* eksternal. Kemudian akan dilakukan pemanggilan dari *website* grup riset ke halaman *web* pencarian tersebut agar tampil pada website grup riset *Intelligent System*. Layanan *hosting* yang digunakan adalah layanan hosting gratis 000webhost yang dapat diakses pada *https://000webhost.com*. Layanan *hosting* gratis digunakan karena nantinya web yang di-*deploy* hanya akan dipanggil untuk ditampilkan pada *web* grup riset *Intelligent System*. Jadi tidak perlu menggunakan layanan *hosting* berbayar. Kemudian pada *website* grup riset *Intelligent System* akan diberikan *link* pada bagian *About Us*. Pada *link* tersebut akan dilakukan pemanggilan pada halaman *web* pencarian yang sudah di-*deploy*.

Kemudian untuk pengujian *web* dilakukan menggunakan metode *black box testing*. Metode *black box testing* dipilih dengan mempertimbangkan bahwa *web* masih berupa pengembangan tahap awal dimana struktur dan logika *source code* belum terlalu dijadikan fokus. Fokus utamanya adalah fungsi pencarian *web* dapat berjalan dengan baik. Uji *black box* dilakukan dengan cara menguji kerja fungsionalitas *web* untuk pencarian berdasarkan kata kunci yang dimasukkan apakah sesuai atau tidak.

IV. HASIL DAN PEMBAHASAN

A. *Preprocessing*

Preprocessing merupakan tahap persiapan dataset agar memiliki kualitas yang lebih baik dan menghilangkan elemen-elemen tidak berguna dalam proses pembobotan kata. Masukan untuk tahap *preprocessing* adalah dataset yang telah dikumpulkan pada *file* CSV. Dataset yang ada di dalam *file* CSV tersebut telah diatur sedemikian rupa agar dapat diproses dalam Python. Dalam *preprocessing* terdapat beberapa tahap yang perlu dilakukan agar dataset siap untuk diolah ke proses selanjutnya. Tahap pertama yang dilakukan yaitu *case folding*. Contoh hasil dari *case folding* ditunjukkan dalam Tabel 1.

TABEL 1. HASIL *CASE FOLDING*

Judul Awal	Hasil *Case Folding*
Automatic short answer scoring using words overlapping methods	automatic short answer scoring overlapping methods
Analisis performa klasifikasi untuk diagnosis penyakit parkinson	analisis performa klasifikasi untuk diagnosis penyakit parkinson
Evaluation 3D gaze tracking in virtual space : A computer graphics appoach	evaluation 3d gaze tracking in virtual space : a computer graphics approach
Sistem pendukung keputusan uang kuliah tunggal dengan metode activity based costing	sistem pendukung keputusan uang kuliah tunggal dengan metode activity based costing
Desain sistem informasi akreditasi program studi berbasis website di Indonesia	desain sistem informasi akreditasi program studi berbasis website di indonesia
A context aware based flood detection and monitoring system using K-median method	a context aware based flood detection and monitoring system using k-median method
Automatic short answer scoring using words overlapping methods	automatic short answer scoring overlapping methods
Analisis performa klasifikasi untuk diagnosis penyakit parkinson	analisis performa klasifikasi untuk diagnosis penyakit parkinson
Evaluation 3D gaze tracking in virtual space : A computer graphics appoach	evaluation 3d gaze tracking in virtual space : a computer graphics approach
Sistem pendukung keputusan uang kuliah tunggal dengan metode activity based costing	sistem pendukung keputusan uang kuliah tunggal dengan metode activity based costing

Tahap selanjutnya yaitu *stemming*. Contoh hasil dari tahap ini ditunjukkan dalam Tabel 2.

TABEL 2. HASIL *STEMMING*

Judul Awal	Hasil *Stemming*
aplikasi metode fuzzy mamdani untuk rekomendasi **pemilihan** minat grup riset mahasiswa	aplikasi metode fuzzy mamdani untuk rekomendasi **pilih** minat grup riset mahasiswa
pemanfaatan konten **pembelajaran** bagi siswa sekolah menengah **kejuruan**	**manfaat** konten **ajar** bagi siswa sekolah menengah **juru**

Tahap selanjutnya yaitu tokenisasi. Contoh hasil tokenisasi ditampilkan pada Gbr 2.

```
ubuntu1604@ubuntu:~/Documents/nyoba2/cobatf$ python3 tftest.py
Enter File:
['"automatic', 'short', 'answer', 'scoring', 'overlapping', 'methods"']
['"kembang', 'data', 'warehouse', 'dukung', 'report', 'instansi', 'perintah"']
['"combat', 'aircraft', 'effectiveness', 'assesment', 'hybrid', 'multi', 'criter
ia', 'decision', 'making', 'methodology"']
['"development', 'automated', 'plasmodium', 'detection', 'malaria', 'diagnosis',
 'blood', 'smear', 'image', 'overview"']
['"review', 'missing', 'values', 'handling', 'methods', 'time', 'series', 'data"
']
['"benefit', 'web', 'technologies', 'higher', 'education', 'student', 'perspecti
ves"']
['"comprative', 'study', 'data', 'mining', 'classification', 'methods', 'cervica
l', 'cancer', 'prediction', 'pap', 'smear"']
['"pattern', 'accesbility', 'level', 'health', 'facilities', 'yogyakarta"']
['"management', 'informantion', 'systems', 'development', 'veterinary', 'hospita
l', 'patient', 'regristration', 'algorithm"']
['"arima', 'implementation', 'predict', 'ammount', 'antiseptic', 'medicine', 'us
age', 'vetenary', 'hospital"']
['"forex', 'trend', 'prediction', 'technique', 'multiple', 'indicators', 'multip
le', 'pairs', 'correlations', 'dss', 'software', 'design"']
['"rancang', 'bangun', 'cloud', 'data', 'center', 'upaya', 'tingkat', 'kualitas'
, 'sekolah', 'smkn', 'won"']
['"review', 'extraction', 'technique', 'approach', 'automatic', 'short', 'answer
', 'grading"']
['"improvement', 'learning', 'coordinate', 'metacognitive', 'strategy', 'path',
'learning', 'classroom', 'intelegent', 'tutoring"']
['"design', 'adaptive', 'learning', 'systems', 'metacognitive', 'strategy', 'pat
h', 'learning', 'classroom', 'intelligent', 'tutoring', 'systems"']
['"cosine', 'similarity', 'determine', 'similarity', 'measure', 'study', 'case',
 'online', 'essay', 'assesment"']
['"student', 'metacognitive', 'modelling', 'radial', 'basis', 'function', 'netwo
rk', 'support', 'adaptive', 'learning"']
```

Gambar 2. Hasil Tokenisasi

Setelah dilakukan tokenisasi, tahap selanjutnya yaitu perhitungan *term frequency*. Contoh hasil perhitungan *term frequency* ditampilkan pada Gbr 3.

```
ubuntu1604@ubuntu:~/Documents/nyoba2/preprocessing/AEP$ python3 countterm.py
(14, 'metode')
(10, 'sistem')
(10, 'learning')
(8, 'support')
(8, 'dukung')
(7, 'data')
(7, '"analisis')
(6, 'studi')
(5, 'putus')
(5, 'bas')
(5, 'adaptive')
(4, 'yogyakarta"')
(4, 'web')
(4, 'sukses')
(4, 'siswa')
(4, 'sakit')
(4, 'pilih')
(4, 'network')
(4, 'negeri')
(4, 'mahasiswa')
(4, 'informasi')
(4, 'fuzzy')
(4, 'framework')
(4, 'diagnosis')
(4, 'decision')
(4, 'bangun')
(4, 'ajar')
(4, '"rancang')
(4, '"evaluasi')
(4, '"decision')
(3, 'usage')
(3, 'ugm')
```

Gambar 3. Hasil Perhitungan *Term Frequency*

Setelah dihitung frekuensi kemunculan kata, tahap akhir dari *preprocessing* yaitu *supervised term frequency* dan *term frequency filtering*. Contoh hasil dari tahap tersebut ditampilkan pada Gbr 4.

```
hasiltffiltering.txt
1  (14, 'metode')
2  (10, 'sistem')
3  (10, 'learning')
4  (8, 'support')
5  (8, 'dukung')
6  (7, 'data')
7  (7, '"analisis')
8  (6, 'studi')
9  (5, 'adaptive')
10 (4, 'yogyakarta"')
11 (4, 'web')
12 (4, 'siswa')
13 (4, 'sakit')
14 (4, 'network')
15 (4, 'negeri')
16 (4, 'mahasiswa')
17 (4, 'informasi')
18 (4, 'fuzzy')
19 (4, 'framework')
20 (4, 'diagnosis')
21 (4, 'decision')
22 (4, 'bangun')
23 (4, '"rancang')
24 (4, '"evaluasi')
25 (4, '"decision')
26 (3, 'usage')
27 (3, 'ugm')
28 (3, 'tenap')
29 (3, 'study')
30 (3, 'sma')
```

Gambar 4. Hasil Tahap Akhir *Preprocessing*

B. Pembobotan Kata

Setelah melalui tahap *preprocessing*, kemudian dilakukan pembobotan kata untuk penentuan kata kunci dari tiap-tiap nama dosen. Pembobotan kata dalam penelitian ini dilakukan menggunakan metode *Term Frequency Inverse Document Frequency* (TF-IDF). Pengerjaan pembobotan kata dilakukan secara manual menggunakan perangkat lunak Microsoft Excel. Hasil akhir dari pembobotan kata ini ditampilkan berupa kualifikasi kata. Kata yang nilainya di atas rata-rata akan terkualifikasi. Kata yang terkualifikasi akan dimasukkan ke dalam basis data. Kata yang tidak terkualifikasi merupakan kata yang nilai TF-IDF nya dibawah rata-rata. Kata yang tidak terkualifikasi merupakan kata yang frekuensi kemunculannya kecil atau yang tersebar di banyak judul publikasi. Contoh hasil dari kualifikasi kata ditampilkan pada Gbr 5.

EU	EV	EW	EX	EY	EZ	FA
						AVERAGE
						4.7723
d71	d72	d73	d74	SUM	Q	Qualification
0	0	0	0	10.123	metode	Qualified
0	0	0	0.828	9.1062	sistem	Qualified
0	0	0	0	9.8187	learning	Qualified
0.915	0.915	0.915	0	8.2349	support	Qualified
0	0	0	0.966	7.7291	dukung	Qualified
0	0	0	0	7.7291	data	Qualified
0	0	0	0	7.7291	analisis	Qualified
0	0	0	0	6.5465	studi	Qualified
0	0	0	0	5.8513	adaptive	Qualified
0	0	0	0	5.0687	yogyakarta	Qualified
0	0	0	0	5.8513	web	Qualified
0	0	0	0	5.0687	siswa	Qualified
0	0	0	1.267	5.0687	sakit	Qualified
0	0	0	0	5.8513	network	Qualified
0	0	0	0	5.0687	negeri	Qualified
0	0	0	0	5.8513	mahasiswa	Qualified
0	0	0	0	5.0687	informasi	Qualified
0	0	0	0	5.8513	fuzzy	Qualified
1.267	1.267	1.267	0	5.0687	framework	Qualified
0	0	0	0	5.0687	diagnosis	Qualified
0.966	0.966	0.966	0	7.7291	decision	Qualified
0	0	0	0	5.0687	bangun	Qualified
0	0	0	0	5.0687	rancang	Qualified
0	0	0	0	5.0687	evaluasi	Qualified
0	0	0	0	4.1763	usage	Not Qualified
0	0	0	0	5.0687	ugm	Qualified
0	0	0	0	5.8513	terap	Qualified
1.392	0	0	0	4.1763	study	Not Qualified
0	0	0	0	4.1763	sma	Not Qualified
0	0	0	0	4.1763	sentimen	Not Qualified
0	0	0	0	4.1763	sekolah	Not Qualified
0	0	1.392	0	4.1763	seasonal	Not Qualified
1.267	0	1.267	0	5.0687	prediction	Qualified
0	0	0	0	4.1763	path	Not Qualified
0	0	0	0	4.1763	normal	Not Qualified

Gambar 5. Hasil Kualifikasi Kata

C. Pembuatan Basis Data

Setelah melalui proses ekstraksi kata kunci, kemudian masuk ke tahap pembuatan basis data. Basis data dibuat menggunakan MySQL dengan bantuan *interface* PhpMyAdmin yang dijalankan melalui *browser*. Pembuatan basis data dilakukan di *localhost*. Untuk rancangan kebutuhan kolom basis data akan ditampilkan pada Gbr 6.

keyword

Column	Type	Null	Default	Comments	MIME
id	int(255)	No			
kata_kunci	varchar(255)	No			
dosen_id	varchar(255)	No			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	164	A	No	

Gambar 6. Kebutuhan Kolom Basis Data

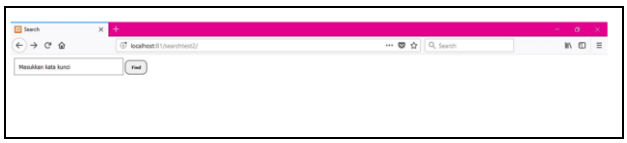
Dari Gbr 6 dapat dilihat bahwa kolom yang diperlukan sebanyak 3 kolom, yaitu kolom *id*, kolom *kata_kunci*, dan kolom *dosen_id*. Kolom *id* berisi identitas atau ID unik tiap-tiap kata kunci. Kolom *kata_kunci* berisi kata kunci. Selain kata kunci hasil dari proses pembobotan kata, ditambah pula kata kunci yang sudah dicantumkan dosen pada halaman profil Google Scholar dan *profile web Intelligent System* masing-masing dosen. Dan kolom *dosen_id* berisi nama dosen yang terkait dengan kata kunci pada kolom *kata_kunci*. Basis data nantinya akan dikaitkan dengan proses *backend* halaman *web* pencarian. Contoh hasil *input* data ke dalam basis data akan ditunjukkan dalam Gbr 7.

Edit Copy Delete	154	biomedical signal	Hanung Adi Nugroho
Edit Copy Delete	155	sinyal biomedis	Hanung Adi Nugroho
Edit Copy Delete	156	image processing	Hanung Adi Nugroho
Edit Copy Delete	157	pemrosesan gambar	Hanung Adi Nugroho
Edit Copy Delete	158	computer vision	Hanung Adi Nugroho
Edit Copy Delete	159	medical instrumentation	Hanung Adi Nugroho
Edit Copy Delete	160	medical imaging	Hanung Adi Nugroho
Edit Copy Delete	161	instrumentasi medis	Hanung Adi Nugroho
Edit Copy Delete	162	pencitraan medis	Hanung Adi Nugroho
Edit Copy Delete	163	statistical pattern	Hanung Adi Nugroho
Edit Copy Delete	164	pola statistik	Hanung Adi Nugroho
Edit Copy Delete	165	fundus	Hanung Adi Nugroho
Edit Copy Delete	166	retinal	Hanung Adi Nugroho
Edit Copy Delete	167	ultrasound	Hanung Adi Nugroho
Edit Copy Delete	168	classification	Hanung Adi Nugroho
Edit Copy Delete	169	klasifikasi	Hanung Adi Nugroho
Edit Copy Delete	170	feature extraction	Hanung Adi Nugroho
Edit Copy Delete	171	ekstraksi fitur	Hanung Adi Nugroho
Edit Copy Delete	172	studi	Hanung Adi Nugroho
Edit Copy Delete	173	sistem	Hanung Adi Nugroho

Gambar 7. Hasil *Input* Data ke Basis Data

D. Pembuatan Halaman Web Pencarian

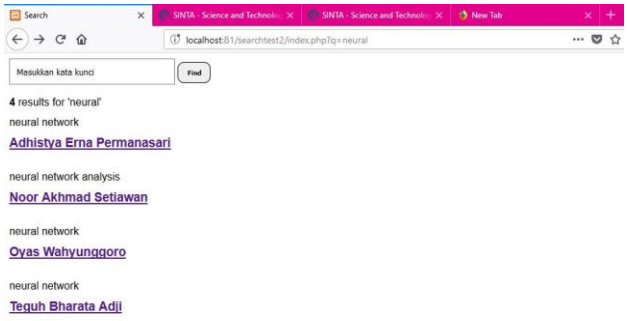
Pengembangan halaman *web* pencarian diawali dengan pengembangan *frontend*. Pengembangan *frontend* merupakan pengerjaan antarmuka halaman *web* atau aplikasi yang akan dikembangkan. Antarmuka halaman *web* pencarian pada penelitian ini tergolong sederhana karena fungsi pencarian yang dilakukan pun tidak kompleks. Jadi hanya perlu *form box* untuk tempat memasukkan kunci dan *button* untuk melakukan pencarian. Antarmuka halaman web pada penelitian ini tidak dijadikan fokus pada penelitian. Pengembangan *frontend* dilakukan dengan menggunakan bahasa HTML dan CSS. Penulisan kode dilakukan dengan menggunakan *text editor* Atom. Proses pengembangan *frontend* dilakukan di *localhost*. Hasil *frontend* halaman *web* pencarian ditunjukkan pada Gbr 8.



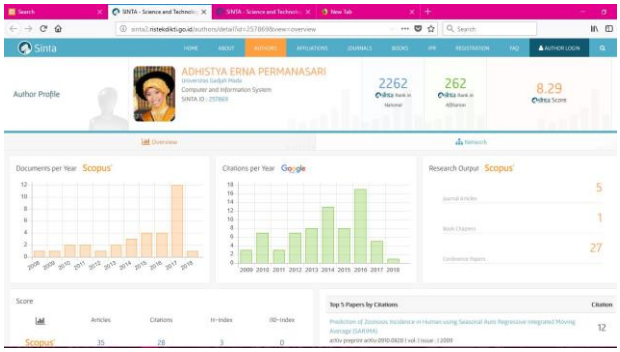
Gambar 8. *Frontend* Halaman *Web* Pencarian

Setelah *frontend* dikembangkan, tahap selanjutnya yaitu pengembangan *backend*. Pengembangan *backend* merupakan proses pengerjaan dari apa yang terjadi

dibelakang suatu halaman *web* atau aplikasi yang sedang dikembangkan. Pada penelitian ini, pengembangan *backend* berupa pengerjaan proses *query* ke basis data berdasarkan kata kunci yang dimasukkan pengguna melalui *form box* pada *frontend* halaman *web* pencarian. Kemudian hasil *query* ditampilkan ke *frontend* yang berupa nama dosen. Lalu pengguna dapat melakukan klik pada nama dosen tersebut untuk mendapatkan informasi lebih lanjut mengenai dosen tersebut. Pengguna akan diarahkan ke halaman profil dosen yang berada di Google Scholar setelah melakukan klik pada nama dosen hasil pencarian. Pengembangan *backend* dilakukan pada *localhost* dan menggunakan bahasa PHP dengan bantuan perangkat lunak XAMPP. Penulisan kode dilakukan menggunakan *text editor* Atom. Contoh hasil pencarian akan ditunjukkan pada Gbr 9 dan contoh hasil klik nama dosen akan ditunjukkan pada Gbr 10.



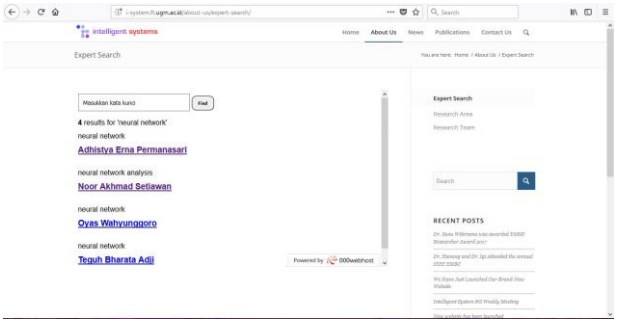
Gambar 9. Contoh Hasil Pencarian



Gambar 10. Contoh Hasil Klik Nama Dosen

E. *Penyematan dan Pengujian Fungsionalitas Web*

Untuk men-*deploy* pada layanan *hosting* gratis 000webhost, perlu dilakukan pembuatan akun terlebih dahulu. Setelah akun dibuat, layanan sudah dapat digunakan untuk *deploy* halaman web pencarian. Karena menggunakan layanan *hosting* gratis, tentunya domain yang diberikan sudah diatur oleh penyedia layanan. Setelah *web* pencarian di-*deploy*, dibuat *link* untuk menempatkan fitur pencarian pada *website* grup riset *Intelligent System*. Fitur pencarian telah berhasil disematkan dan dapat diakses pada *http://i-system.ft.ugm.ac.id/about-us/expert-search/*. Hasil penyematan web pencarian ditunjukkan pada Gbr 11.



Gambar 11. Hasil Penyematan *Web* Pencarian

Pengujian *black box* dilakukan untuk menguji fungsionalitas *web* apakah berjalan sesuai ekspektasi tidak tanpa memperhatikan struktur dan logika kode di dalamnya. Pengujian dilakukan dengan mencoba melakukan pencarian dengan memasukkan kata kunci yang sudah dihasilkan dari proses ekstraksi ditambah kata kunci bidang keahlian tiap dosen yang sudah disertakan pada halaman profil dosen Google Scholar dan *web* grup riset *Intelligent System*. Hasil pengujian dapat dilihat pada Tabel 3.

TABEL 3. HASIL UJI *BLACK BOX*

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
Memasukkan kata kunci untuk masing-masing dosen. Kata kunci ini adalah hasil dari ekstraksi ditambah kata kunci bidang yang tertera pada profil Google Scholar.	Muncul nama dosen yang terhubung dengan kata kunci yang dimaksud.	Berhasil
Melakukan klik pada hasil pencarian tiap nama dosen yang keluar.	Terbuka ***tab*** baru *browser* yang berisi halaman profil dosen di Google Scholar.	Berhasil
Memasukkan kata kunci yang tidak relevan.	Tidak akan menampilkan hasil (0 *results*)	Berhasil
Memasukkan kata kunci nama dosen grup riset I-Syis	Muncul hasil nama dosen sesuai kata kunci yang dimasukkan.	Berhasil

F. KELEMAHAN SISTEM

Sistem pencarian yang telah dikembangkan masih memiliki beberapa kekurangan, yaitu:

1. Tampilan antarmuka sistem pencarian yang tergolong sederhana. Perlu adanya peningkatan dari segi *frontend* sistem agar lebih memperhatikan elemen *User Experience*.
2. Keamanan sistem yang belum dijadikan fokus. Perlu adanya peningkatan keamanan sistem.
3. Sistem pencarian mengarahkan hasil pencarian ke profil Google Scholar tiap dosen dengan membuka tab *browser* baru (*new tab browser*). Pada *browser* keluaran terbaru akan ada notifikasi *allow to open new tab* karena *browser* keluaran terbaru sangat memperhatikan keamanan. Oleh karena itu pengguna perlu memberikan akses untuk membolehkan sistem membuka tab baru pada *browser*.

V. KESIMPULAN

Penelitian yang dilakukan ini telah berhasil mengembangkan halaman web pencarian berdasarkan kata kunci untuk fitur pada website grup riset *Intelligent System* menggunakan Metode TF-IDF. Metode ini menggunakan dua faktor untuk menentukan bobot suatu kata, yaitu dengan frekuensi kemunculan kata dan berapa banyak dokumen. Metode TF-IDF sering digunakan dalam riset *text mining* karena memiliki akurasi yang baik serta mudah untuk diterapkan. Namun metode TF-IDF masih memiliki kekurangan yaitu jumlah data yang diproses terbatas di kategori kecil dan menengah. Apabila jumlah data yang diproses sangat besar, maka perlu metode yang lebih modern seperti metode *machine learning.* Untuk penelitian selanjutnya perlu dilakukan peningkatan pada tahap *stopword removal* dengan menggunakan *stopword list* kata untuk Bahasa Inggris karena beberapa judul penelitian menggunakan Bahasa Inggris. Selain itu perlu dilakukan peningkatan pada desain *front end* agar lebih menarik.

REFERENSI

[1] E. Indrayani, "Management of Academic Information System (AIS) at Higher Education in The City Of Bandung" in 13 th International Educational Technology Conference, 2013 © ScienceDirect. doi : 10.1016/j.sbspro.2013.10.381

[2] Intelligent Systems Research Group [Online]. Available : http://ai.te.ugm.ac.id/ [Diakses : 24 Agustus 2017]

[3] A. A. Hakim, A. Erwin, K.I. Eng, M. Galinium, W. Muliady, "Automated Document Classification for News Article in Bahasa Indonesia based on Term Frequency Inverse Document Frequency (TF-IDF) Approach", 6th International Conference on Information Technology and Electrical Engineering (ICITEE) Yogyakarta Indonesia, Faculty of Engineering and of Information Technology, Swiss German University, BSD, Tangerang, Indonesia, 2014.

[4] P. Turney, "Learning Algorithms for Keyphrases Extraction", Institute for Information Technology, National Research Council of Canada, 1999.

[5] K. Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval", Journal of Documentation, 1972, pp. 11-21.

[6] G. Salton, A. Wong, C. Yang, "A Vector Space Model for Automatic Indexing", Communications of the ACM, 1975, pp. 613-620.

[7] M. Andrade, A. Valencia, "Automatic Extraction of Keywords from Scientific Text: Application to the Knowledge Domain of Protein Families", Bioinformatics, 1998, pp. 600-607.

[8] M. Hearst, "What is Text Mining ?", 2003. [Online]. Available: http://people.ischool.berkeley.edu/~hearst/text-mining.html.

[9] D. D. Lewis, "Text Representation for Intelligent Text Retrieval: A Classification-oriented view", In S. J. Paul (Ed.), Text-based intelligent systems: current research and practices in information extraction and retrieval, Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates, Inc, 1992, pp. 179-197.

[10] R. J. Quinlan, "Learning Efficient Classification Procedures and Their Applications to Chess and Games", In Ryszard S. Michalski, Jaime F. Carbonell, and Tom M. Mitchell (Eds.), Machine learning: An artificial intelligence approach, Los Altos, CA: Morgan Kaufmann, 1983, pp. 463-482.

[11] K. Sparck Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval", Journal of Documentation, 1972, pp. 11-21.

[12] K. Sparck Jones, "IDF Term Weighting and IR Research Lessons", Journal of Documentation, 2004, pp. 521-523.

[13] S. Robeston, "Understanding Inverse Document Frequency: On Theoretical Argument for IDF", Journal of Documentation, 2004, pp. 503-520.

[14] F. M. Caropreso, S. Matwin, F. Sebastiani, "A Learner-Independent evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization", in Amita G. Chin (Ed.), Text databases and document management: theory and practice, Hershey, U.S: Idea Group Publishing, 2001, pp. 78-102.

[15] M. D. Christopher, S. Hinrich, "Foundations of Statistical Natural Language Processing", Cambridge, Massachusetts: MIT Press, 2001, pp. 529-574.

[16] W. Zhang, T. Yoshida, X. Tang, "A Comparative Study of TF*IDF, LSI and Multi-Words for Text Classification" in Expert Systems with Application, 2010 © Elsevier Ltd. Available at ScienceDirect. doi: 10.1016/j.eswa.2010.08.066