Australian Journal of Basic and Applied Sciences, 7(4): 342-352, 2013
ISSN 1991-8178

Design of Thesis Topic Search Engine with Information Retrieval and Vector Space Model of TF-IDF Weighting

Nilo Legowo, Sofia, Rojali

Computer Science Departemetn, Binus University Jl. Kebon Jeruk Raya no.27 Jakarta 11530, Indonesia

Abstract: The development of internet makes improvement in relevant information needs. A way to get relevant information in internet is by using search engine application. Search engine application is a form of information retrieval system. Thesis searching is a problem that students face in their final study. A way to help them solve their problem is by using search engine, especially search engine that specifically looking for data from the theses that have been made. To make it more effective, the search should be not only by title but also by abstract of those theses. Methods that used is by literature study, then makes a search engine by applying vector space model of TF-IDF weighting. Every term is weighted to document where these term are and to other documents that available. Then their similarity is measured by using vector space model to the query that given by user. UML diagram is used to explain the system design. The search result by using information retrieval system with vector space model of TF-IDF weighting will giving documents that sorted by their similarity with query that searched by user.

Key words: *information retrieval system, abstract, vector space model, TF-IDF weighting*

INTRODUCTION

The development of technology especially internet is instrumental in everyday life. By using internet, information can be easily shared and accessed by many people. There is much information that makes the need of relevant information is increasing. A way to get relevant information is by using information retrieval system. An application of information retrieval system is search engine that usually used to access information from the internet. All areas of life become easier with the help of search engine. There are many important information that can be found by search engine so that the user can get relevant information in what they search.

In information retrieval system, there are many models that can be used to measure the similarity of the searches, including Boolean model, vector space model and probabilistic model. In this thesis, we will use vector space model that can display result that sorted by similarity between query and the information contained in the document. This is done so that users of the search engine that will obtain data sorted in similarity. It will make the searching process more efficient than by using search engine that does sorted by similarity between query and document.

Search thesis topic is one of the problems to be faced by every student at the end of his lecture. A way to help students looking for thesis topics is by using search engine, especially search engine that looking for data from theses which have been made before.

Therefore, in this journal the author wants to help student in searching thesis topic by using information retrieval system and vector space model of TF-IDF weighting. Hopefully by using this search engine, the search will be done more effectively and efficiently and can find appropriate document.

Theorem:

Information retrieval system according to Kowalski dan Maybury (2000, p2) is a system that can store, retrieve and maintain information. In this context, information may consist of text (including number and date), picture, audio, video and the others multimedia object.

According to E. Garcia in article at Mi Islita with topic Document Indexing Tutorial for Information Retrieval Students and Search Engine Marketers**,** there are 5 step that must do to build an inverted index:

(1)Deleting markup and format: At this step, all markup tags and special format are deleted from document.

(2)Tokenization: At this step, words in sentences are described one by one into a single word. Furthermore it also made the removal of punctuation and changes all characters in the word to lowercase.

(3)Filtration: At this step, we choose term that can be used to represent document and distinguish the document from the other document in the collection.

(4)Stemming: Stemming is conversion process from term into basic word.

Corresponding Author: Nilo Legowo, Computer Science Departemetn, Binus University Jl. Kebon Jeruk Raya no.27 Jakarta 11530, Indonesia
Telepon: +62-21-5345830, 5350660 Fax: 021-5300244; E-mail: nlegowo@binus.edu

(5)Weighting: Term weighting is weighted based on the model that chosen, it can be local, global or combination of both weighting. One of the commonly used weighting is the weighting that combines local and global weighting that called TF-IDF weighting.

Term Frequency (TF) according to Polettini (2004, p2) is formula that used to count how many times a term appear in a document. Frequency of term *i* in document *j* defined by Cios *et al.* (2007, p460) as:

$$tf_{ij} = \frac{f_{ij}}{\max_i(f_{ij})}$$

where f_{ij} is total appearance of term *i* in document *j*. This frequency is normalized by frequency from the most frequent term in the document.

Inverse Document Frequency (IDF) used to identify the difference by term *i*. In general, term that appears in many documents cannot used to measure a specific topic. Formula to measure inverse document frequency is:

$$idf_i = \log_2\left(\frac{n}{df_i}\right)$$

where df_i is document frequency of term i or can also interpreted as total document that contain term i. We use $\log_2$ to muffle relative effect to tf_{ij}.

Weights w_{ij} calculated by using TF-IDF that already explained before and the formula is:

$$w_{ij} = tf_{ij} \times idf_i$$

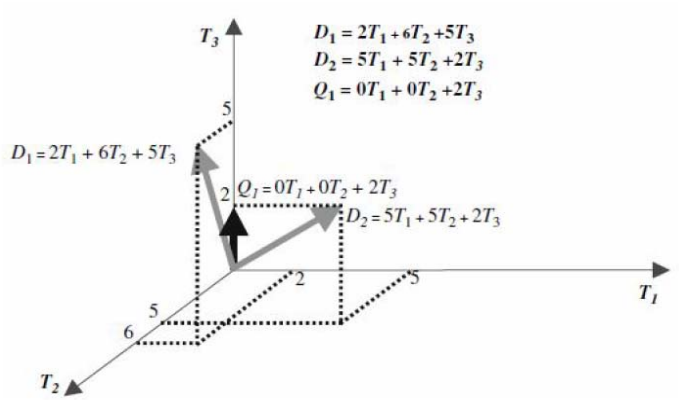

Fig. 1: Vector Space Model Source : Cios *et al.*(2007, p460)

According to Cios *et al.* (2007, p459), similarity between document determined by representation bag-of-words and by using vector space model, where every document in database and query from user represented by multidimensional vector. Dimension of vector depending on terms in database.

A way to measure the text similarity that most popular is by using cosine similarity. This measurement calculate the distance between two vector. The smaller angle between two vector, then the similarity between document and query are getting bigger. To take measurement, do the following calculation:

$$\cos\theta = similarity\left(\vec{d_j}, \vec{q}\right) = \frac{\vec{d_j} \cdot \vec{q}}{\left|\vec{d_j}\right|\left|\vec{q}\right|} = \frac{\sum_{i=1}^{t}\left(w_{ij} \cdot w_{iq}\right)}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$

So the process can be described as:

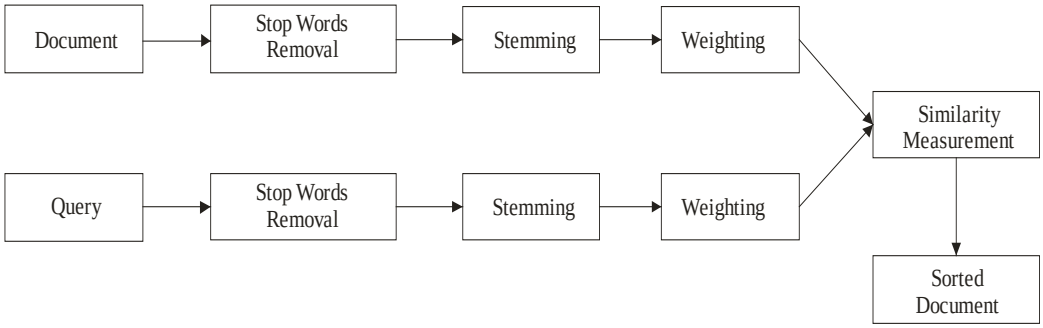

Fig. 2: Information Retrieval System Algorithm

Analysis of Problem:

The development of internet is rapidly increasing and makes any kind of information can be easily accessed and disseminated. The need for relevant information is an important thing in today's modern era of technology. The existence of internet makes many kind of information can be accessed easily, but unfortunately the information obtained is not always relevant to what is actually sought by internet users. A way to get relevant information is by using information retrieval system. Application of information retrieval system that most frequently used is search engine.

There are some ways to calculate similarity between document and query, in this journal similarity will calculate by using vector space model. This model will give sorted document by their similarity with query so that the search becomes more effective and efficient. For weighting, we will use Term Frequency – Inverse Document Frequency (TF-IDF) weighting, where frequency that calculated is not only frequency of a term in document, but also frequency a term in every documents.

In this journal, documents that will be used is title and abstract of thesis that has been made in department of Computer Science and Mathematics Binus University. For preprocessing, words that related with thesis from this department will be stored for ease of stemming.

Searching Procedure:

Abstract of a thesis usually be divided into opening, content and closing consists of approximately 30 sentences (300 words). Abstract contained words that became the keywords of a thesis. Suppose that the abstract of some thesis and its title are: The following are examples of data and abstracts of thesis title in Indonesian sentence (Abstract 1),

Abstract 1
STUDI DAN IMPLEMENTASI WATERMARKING CITRA DIJITAL DENGAN PENDEKATAN DISCRETE COSINE TRANSFORM.
Kemudahan dan kecepatan bertukar informasi di internet, menyebabkan penyebaran informasi semakin mudah dilakukan. Tapi terkadang informasi yang disebarkan ini digunakan sewenang-wenang oleh pihak yang tidak bertanggung jawab. Hal ini menjadi salah satu contoh pelanggaran hak cipta.
Salah satu cara untuk mencegah pelanggaran hak cipta tersebut adalah dengan digital watermarking. Metode yang digunakan untuk watermarking citra dijital adalah Discrete Cosine Transform (DCT).
Penggunaan DCT dalam watermarking cukup bisa dihandalkan. Hasil pengujian menunjukkan bahwa watermarking citra dijital dengan menggunakan DCT memiliki ketahanan yang cukup tinggi terhadap kompresi JPEG.

The following are examples of data and abstracts of thesis title in Indonesian sentence (Abstract 2),

Abstract 2

PERANCANGAN PROGRAM RETRIVAL CITRA BERBASIS KONTEN MENGGUNAKAN TRANSFORMASI WALSH-HADAMARD TERHADAP RATA-RATA BARIS DAN KOLOM WARNA CITRA

Bidang multimedia mengalami perkembangan yang sangat pesat. Berbagai citra dihasilkan setiap harinya, baik melalui pengambilan foto secara alami maupun melalui proses rekayasa. Dengan semakin banyaknya citra yang dihasilkan, pencarian citra juga semakin susah dilakukan.

Metode yang paling banyak digunakan untuk mencari citra pada saat ini adalah dengan melakukan pengindeksan melalui kata kunci yang berhubungan dengan citra. akan tetapi, terdapat bebedapa permasalahan dengan cara ini, antara lain citra memiliki banyak arti, pengindeksan citra sulit karena memiliki banyak kata kunci, dan adanya perbedaan dari segi bahasa untuk menyatakan suatu citra.

Skripsi ini mencoba menggunakan transformasi Walsh-Hadamard terhadap rata-rata baris dan kolom warna citra sebagai vektor fitur untuk dapat melakukan retrival terhadap citra.

The following are examples of data and abstracts of thesis title in Indonesian sentence (Abstract 3),

Abstract 3

PERANCANGAN PROGRAM APLIKASI STEGANOGRAPHY PADA DIGITAL VIDEO BERBASIS METODE SINGULAR VALUE DECOMPOSITION DAN DISCRETE WAVELET TRANSFORM

Teknologi informasi dan komunikasi pada dunia digital masa kini mengalami perkembangan yang sangat pesat dengan kehadiran jaringan internet. Pertukaran informasi antara seseorang dengan orang lain dapat dilakukan dengan mudah dan cepat dalam berbagai bentuk tanpa batas ruang dan waktu. Muncul kebutuhan pengiriman sebuah informasi yang mengandung rahasia dan privasi tanpa diketahui orang yang tidak dituju, hanya antara pengirim pesan dan penerima pesan saja. Proses pengiriman informasi rahasia yang aman, cepat dan akurat menjadi prioritas utama. Dibutuhkan suatu aplikasi yang mampu menyembunyikan informasi ke dalam suatu media yang dapat diakses oleh semua orang, namun mereka tidak menyadari bahwa media tersebut telah disisipkan informasi rahasia.

Untuk menyikapi masalah tersebut, gabungan antara steganography dan cryptography pada media digital video dapat memastikan keamanan pengiriman pesan. Steganography akan menyisipkan pesan ke dalam suatu media sehingga tidak diketahui keberadaannya, sedangkan cryptography akan mengacak pesan (enkripsi) sehingga tidak dapat terbaca. Aplikasi ini berbasiskan metode Singular Value Decomposition dan Discrete Wavelet Transform pada steganography. Sedangkan pada cryptography dengan enkripsi Data Encryption Standard. Media yang digunakan adalah digital video dengan format uncompressed AVI.

After going through stop words elimination and stemming process will be obtained a collection of words as follows:

Abstract 1
studi implemen watermark citra digital dekat diskrit cosinus transform mudah cepat tukar info internet sebab sebar info makin mudah laku tapi kadang info sebar guna wenang wenang pihak tidak tanggung jawab hal jadi salah satu contoh langgar hak cipta
salah satu cara cegah langgar hak cipta digital watermark metode guna watermark citra digital diskrit cosinus transform dct
guna dct watermark cukup bisa handal hasil uji tunjuk bahwa watermark citra digital guna dct milik tahan cukup tinggi hadap kompresi jpeg

Abstract 2
rancang program retrival citra basis konten guna transform walsh hadamard hadap rata rata baris kolom warna citra bidang multimedia alam kembang sangat pesat bagai citra hasil setiap hari baik lalu ambil foto cara alami maupun lalu proses rekayasa makin banyak citra hasil cari citra makin susah laku
metode paling banyak guna cari citra saat laku indeks lalu kata kunci hubung citra akan tetapi dapat berapa masalah cara antara lain citra milik banyak arti indeks citra sulit milik banyak kata kunci ada beda segi bahasa nyata suatu citra skripsi coba guna transform walsh hadamard hadap rata rata baris kolom warna citra sebagai vektor fitur laku retrival hadap citra

Abstract 3
rancang program aplikasi steganografi digital video basis metode singular nilai dekomposisi diskrit wavelet transform teknologi info komunikasi dunia digital masa kini alam kembang sangat pesat hadir jaringan internet tukar info antara orang orang lain laku mudah cepat bagai bentuk tanpa batas ruang waktu muncul butuh kirim sebuah info kandung rahasia privasi tanpa tahu orang tidak tuju hanya antara kirim pesan terima pesan saja proses kirim info rahasia aman cepat akurat jadi prioritas utama butuh suatu aplikasi mampu sembunyi info suatu media akses semua orang namun mereka tidak sadar bahwa media telah sisip info rahasia
untuk sikap masalah gabung antara steganografi kriptografi media digital video pasti aman kirim pesan steganografi akan sisip pesan suatu media tidak tahu berada kriptografi akan acak pesan enkripsi tidak baca aplikasi basis metode singular nilai dekomposisi diskrit wavelet transform steganografi kriptografi enkripsi data enkripsi standard media guna digital video format uncompressed avi

From that example, then we calculate the weight using TF-IDF formula in **Table 1, Table 2, Table 3**.

Then when user enters a query, it will make the following calculation process: (**Table 4.**)

Query : Pengolahan citra digital
Preprocessing (stop words elimination and stemming) : olah citra digital

Similarity Calculation:
Abstract 1:

$$similarity\left(\overrightarrow{D_1},\vec{Q}\right)=\frac{\left(0\cdot 0+0.350977\cdot 0.584962+0.46797\cdot 0.584962\right)}{3.125536\cdot 0.82726}$$

$$=\frac{0.479053}{2.585631}$$

$$=0.185275$$

Abstract 2:

$$similarity\left(\overrightarrow{D_2},\vec{Q}\right)=\frac{\left(0\cdot 0+0.584962\cdot 0.584962+0\cdot 0.584962\right)}{1.539891\cdot 0.82726}$$

$$=\frac{0.342181}{1.27389}$$

$$=0.268611$$

Abstract 3:

$$similarity\left(\overrightarrow{D_3},\vec{Q}\right)=\frac{\left(0\cdot 0+0\cdot 0.584962+0.389975\cdot 0.584962\right)}{4.066151\cdot 0.82726}$$

$$=\frac{0.22812}{3.363764}$$

$$=0.067817$$

From the result, we can get conclusion that abstract rank by their similarity with query starting from the most similar are 2, 1 and 3.

Design:

Class Diagram:

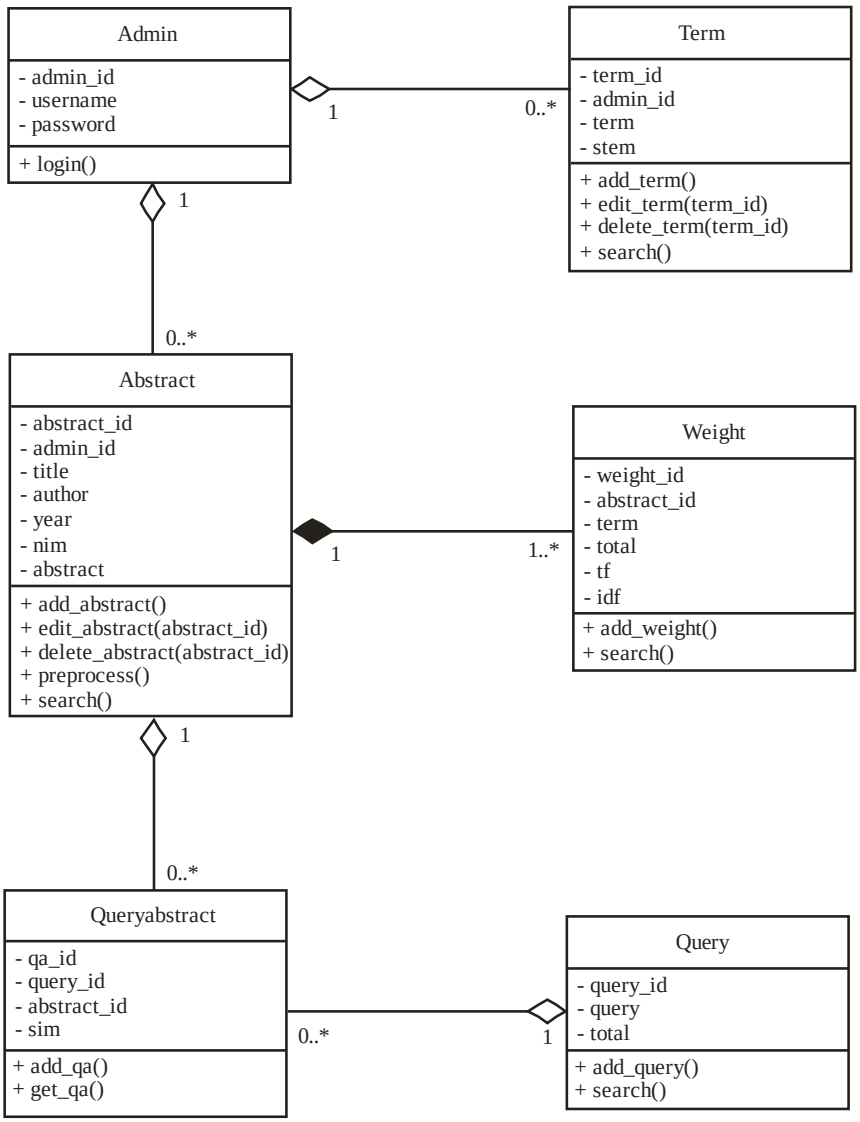

Fig. 3: Class Diagram

Use-Case Diagram:

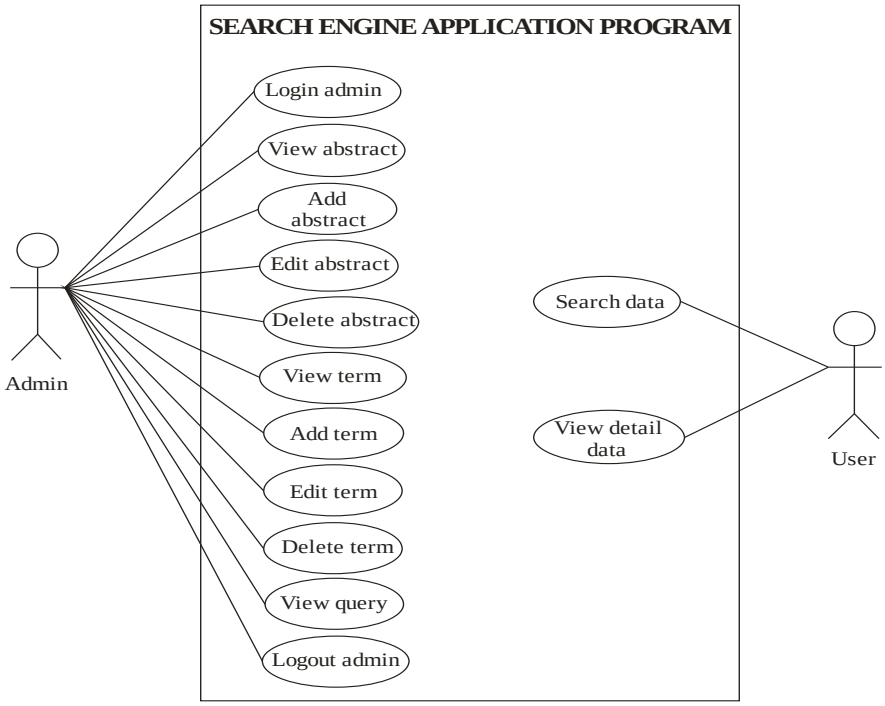

Fig. 4: Use-Case Diagram

Table 1: TF-IDF Calculation of Abstract 1

Term	Total	TF	IDF	TF-IDF
studi	1	0.2	1.58496	0.316992
implemen	1	0.2	1.58496	0.316992
watermark	5	1	1.58496	1.58496
citra	3	0.6	0.584962	0.350977
digital	4	0.8	0.584962	0.46797
dekat	1	0.2	1.58496	0.316992
diskrit	2	0.4	0.584962	0.233985
cosinus	2	0.4	1.58496	0.633984
transform	2	0.4	0	0
mudah	2	0.4	0.584962	0.233985
cepat	1	0.2	0.584962	0.116992
tukar	1	0.2	0.584962	0.116992
info	3	0.6	0.584962	0.350977
internet	1	0.2	0.584962	0.116992
sebab	1	0.2	1.58496	0.316992
sebar	2	0.4	1.58496	0.633984
makin	1	0.2	0.584962	0.116992
laku	1	0.2	0	0
tapi	1	0.2	1.58496	0.316992
kadang	1	0.2	1.58496	0.316992
guna	4	0.8	0	0
wenang	2	0.4	1.58496	0.633984
pihak	1	0.2	1.58496	0.316992
tidak	1	0.2	0.584962	0.116992
tanggung	1	0.2	1.58496	0.316992
jawab	1	0.2	1.58496	0.316992
hal	1	0.2	1.58496	0.316992
jadi	1	0.2	0.584962	0.116992
salah	2	0.4	1.58496	0.633984
satu	2	0.4	1.58496	0.633984
contoh	1	0.2	1.58496	0.316992
langgar	2	0.4	1.58496	0.633984
hak	2	0.4	1.58496	0.633984
cipta	2	0.4	1.58496	0.633984
cara	1	0.2	0.584962	0.116992
cegah	1	0.2	1.58496	0.316992
metode	1	0.2	0	0
dct	3	0.6	1.58496	0.950976

Term	Total	TF	IDF	TF-IDF
cukup	2	0.4	1.58496	0.633984
bisa	1	0.2	1.58496	0.316992
handal	1	0.2	1.58496	0.316992
hasil	1	0.2	0.584962	0.116992
uji	1	0.2	1.58496	0.316992
tunjuk	1	0.2	1.58496	0.316992
bahwa	1	0.2	0.584962	0.116992
milik	1	0.2	0.584962	0.116992
tahan	1	0.2	1.58496	0.316992
tinggi	1	0.2	1.58496	0.316992
hadap	1	0.2	0.584962	0.116992
kompresi	1	0.2	1.58496	0.316992
jpeg	1	0.2	1.58496	0.316992

Vector length = $\sqrt{\sum (TF - IDF)^2}$ = 3.125536

Table 2: TF-IDF Calculation of Abstract 2

Term	Total	TF	IDF	TF-IDF
rancang	1	0.0833	0.5849	0.0487
program	1	0.0833	0.5849	0.0487
retrival	2	0.1666	1.584	0.2641
citra	12	1	0.5849	0.5849
basis	1	0.0833	0.5849	0.0487
konten	1	0.0833	1.5849	0.1320
guna	3	0.25	0	0
transform	2	0.1666	0	0
walsh	2	0.1666	1.5849	0.2641
hadamard	2	0.1666	1.5849	0.2641
hadap	3	0.25	0.5849	0.1462
rata	4	0.3333	1.5849	0.5283
baris	2	0.1666	1.5849	0.2641
kolom	2	0.1666	1.5849	0.2641
warna	2	0.1666	1.5849	0.2641
bidang	1	0.0833	1.5849	0.1320
multimedia	1	0.0833	1.5849	0.1320
alam	1	0.0833	0.5849	0.0487
kembang	1	0.0833	0.5849	0.0487
sangat	1	0.0833	0.5849	0.0487
pesat	1	0.0833	0.5849	0.0487
bagai	1	0.0833	0.5849	0.0487
hasil	2	0.1666	0.5849	0.0974
setiap	1	0.0833	1.5849	0.1320
hari	1	0.0833	1.5849	0.1320
baik	1	0.0833	1.5849	0.1320
lalu	3	0.25	1.5849	0.3962
ambil	1	0.0833	1.5849	0.1320
foto	1	0.0833	1.5849	0.1320
cara	2	0.1666	0.5849	0.0974
alami	1	0.0833	1.5849	0.1320
maupun	1	0.0833	1.5849	0.1320
proses	1	0.0833	0.5849	0.0487
rekayasa	1	0.0833	1.5849	0.1320
makin	2	0.1666	0.5849	0.0974
banyak	4	0.3333	1.5849	0.5283
cari	2	0.1666	1.5849	0.2641
susah	1	0.0833	1.5849	0.1320
laku	3	0.25	0	0
metode	1	0.0833	0	0
paling	1	0.0833	1.5849	0.1320
saat	1	0.0833	1.5849	0.1320
indeks	2	0.1666	1.5849	0.2641
kata	2	0.1666	1.5849	0.2641
kunci	2	0.1666	1.5849	0.2641
hubung	1	0.0833	1.5849	0.1320
akan	1	0.0833	0.5849	0.0487
tetapi	1	0.0833	1.5849	0.1320
dapat	1	0.0833	1.5849	0.1320
berapa	1	0.0833	1.5849	0.1320
masalah	1	0.0833	0.5849	0.0487

Term	Total	TF	IDF	TF-IDF
antara	1	0.0833	0.5849	0.0487
lain	1	0.0833	0.5849	0.0487
milik	2	0.1666	0.5849	0.0974
arti	1	0.0833	1.5849	0.1320
sulit	1	0.0833	1.5849	0.1320
ada	1	0.0833	1.5849	0.1320
beda	1	0.0833	1.5849	0.1320
segi	1	0.0833	1.5849	0.1320
bahasa	1	0.0833	1.5849	0.1320
nyata	1	0.0833	1.5849	0.1320
suatu	1	0.0833	0.5849	0.0487
skripsi	1	0.0833	1.5849	0.1320
coba	1	0.0833	1.5849	0.1320
sebagai	1	0.0833	1.5849	0.1320
vektor	1	0.0833	1.5849	0.1320
fitur	1	0.0833	1.5849	0.1320

Vector length = $\sqrt{\sum (TF - IDF)^2}$ = 1.539891

Table 3: TF-IDF Calculation of Abstract 3

Term	Total	TF	IDF	TF-IDF
rancang	1	0.1666	0.5849	0.0974
program	1	0.1666	0.5849	0.0974
aplikasi	3	0.5	1.5849	0.7924
steganografi	4	0.6666	1.5849	1.0566
digital	4	0.6666	0.5849	0.3899
video	3	0.5	1.5849	0.7924
basis	2	0.3333	0.5849	0.1949
metode	2	0.3333	0	0
singular	2	0.3333	1.5849	0.5283
nilai	2	0.3333	1.5849	0.5283
dekomposisi	2	0.3333	1.5849	0.5283
diskrit	2	0.3333	0.5849	0.1949
wavelet	2	0.3333	1.5849	0.5283
transform	2	0.3333	0	0
teknologi	1	0.1666	1.5849	0.2641
info	6	1	0.5849	0.5849
komunikasi	1	0.1666	1.5849	0.2641
dunia	1	0.1666	1.5849	0.2641
masa	1	0.1666	1.5849	0.2641
kini	1	0.1666	1.5849	0.2641
alam	1	0.1666	0.5849	0.0974
kembang	1	0.1666	0.5849	0.0974
sangat	1	0.1666	0.5849	0.0974
pesat	1	0.1666	0.5849	0.0974
hadir	1	0.1666	1.5849	0.2641
jaringan	1	0.1666	1.5849	0.2641
internet	1	0.1666	0.5849	0.0974
tukar	1	0.1666	0.5849	0.0974
antara	3	0.5	0.5849	0.2924
orang	4	0.6666	1.5849	1.0566
lain	1	0.1666	0.5849	0.0974
laku	1	0.1666	0	0
mudah	1	0.1666	0.5849	0.0974
cepat	2	0.3333	0.5849	0.1949
bagai	1	0.1666	0.5849	0.0974
bentuk	1	0.1666	1.5849	0.2641
tanpa	2	0.3333	1.5849	0.5283
batas	1	0.1666	1.5849	0.2641
ruang	1	0.1666	1.5849	0.2641
waktu	1	0.1666	1.5849	0.2641
muncul	1	0.1666	1.5849	0.2641
butuh	2	0.3333	1.5849	0.5283
kirim	4	0.6666	1.5849	1.0566
sebuah	1	0.1666	1.5849	0.2641
kandung	1	0.1666	1.5849	0.2641
rahasia	3	0.5	1.5849	0.7924
privasi	1	0.1666	1.5849	0.2641
tahu	2	0.3333	1.5849	0.5283

Term	Total	TF	IDF	TF-IDF
tidak	4	0.6666	0.5849	0.3899
tuju	1	0.1666	1.5849	0.2641
hanya	1	0.1666	1.5849	0.2641
pesan	5	0.8333	1.5849	1.3207
terima	1	0.1666	1.5849	0.2641
saja	1	0.1666	1.5849	0.2641
proses	1	0.1666	0.5849	0.0974
aman	2	0.3333	1.5849	0.5283
akurat	1	0.1666	1.5849	0.2641
jadi	1	0.1666	0.5849	0.0974
prioritas	1	0.1666	1.5849	0.2641
utama	1	0.1666	1.5849	0.2641
suatu	3	0.5	0.5849	0.2924
mampu	1	0.1666	1.5849	0.2641
sembunyi	1	0.1666	1.5849	0.2641
media	5	0.8333	1.5849	1.3207
akses	1	0.1666	1.5849	0.2641
semua	1	0.1666	1.5849	0.2641
namun	1	0.1666	1.5849	0.2641
mereka	1	0.1666	1.5849	0.2641
sadar	1	0.1666	1.5849	0.2641
bahwa	1	0.1666	0.5849	0.0974
telah	1	0.1666	1.5849	0.2641
sisip	2	0.3333	1.5849	0.5283
untuk	1	0.1666	1.5849	0.2641
sikap	1	0.1666	1.5849	0.2641
masalah	1	0.1666	0.5849	0.0974
gabung	1	0.1666	1.5849	0.2641
kriptografi	3	0.5	1.5849	0.7924
pasti	1	0.1666	1.5849	0.2641
akan	2	0.3333	0.5849	0.1949
berada	1	0.1666	1.5849	0.2641
acak	1	0.1666	1.5849	0.2641
enkripsi	3	0.5	1.5849	0.7924
baca	1	0.1666	1.5849	0.2641
data	1	0.1666	1.5849	0.2641
standard	1	0.1666	1.5849	0.2641
guna	1	0.1666	0	0
format	1	0.1666	1.5849	0.2641
uncompressed	1	0.1666	1.5849	0.2641
avi	1	0.1666	1.5849	0.2641

Vector Length = $\sqrt{\sum (TF - IDF)^2}$ = 4.066151

Table 4: TF-IDF Calculation of Query

Term	Total	TF	IDF	TF-IDF
olah	1	1	0	0
citra	1	1	0.5849	0.5849
digital	1	1	0.5849	0.5849

Vector Length = $\sqrt{\sum (TF - IDF)^2}$ = 0.82726

Conclusion:

The conclusion that we can get for this search engine design are

(1)Search engine are made to perform search by query that consists of several words. (2)This search engine can help students who want to search reference in thesis topics of Computer Science & Mathematics. (3)Searching process is more accurate because data that searched are based on abstract, so the scope became wider. (4)Vector space model of TF-IDF weighting can provide search results that sorted by their similarity with query.

REFERENCES

Anton, H., C. Rorres, 2005. *Elementary Linear Algebra*. (9th Edition). New York: John Wiley & Sons

Cios, K.J., W. Pedrycz, R.W. Swiniarski, L.A. Kurgan, 2007. *Data Mining A Knowledge Discovery Approach*. New York: Springer.

Connolly, T.M., C.E. Begg, 2002. *Database Systems: a Practical Approach to Design, Implementation, and Management*. (3rd Edition). Harlow: Addison-Wesley.

Dawkins, P., 2011. *Paul's Online Math Notes.* Retrieved 1 November 2011 from http://tutorial.math.lamar.edu/Classes/CalcII/DotProduct.aspx

Eaglestone, B., M. Ridley, 2001. *Web Database Systems.* London: McGRAW-HILL.

Garcia. E., 2005. *Document Indexing Tutorial.* Retrieved 27 October 2011 from http://www.miislita.com/information-retrieval-tutorial/indexing.html

Husni, 2010. *Information Retrieval*. Retrieved 25 October 2011 from http://husni.trunojoyo.ac.id/wp-content/uploads/2010/03/Husni-IR-dan-Klasifikasi.pdf

Husni, 2010. *Sistem Temu-Balik Informasi*. Retrieved 25 October 2011 from http://husni.trunojoyo.ac.id/wp-content/uploads/2010/03/STBI2010-02.pdf

Kowalski, G., M.T. Maybury, 2000. *Information Storage and Retrieval Systems : Theory and Implementation*. (2nd edition). Massachusetts: Kluwer Academic Publishers.

Mathiassen, L., A. Munk-Madsen, P.A. Nielsen, & J. Stage, 2000. *Object Oriented Analysis & Design.* Aalborg: Marko Publishing.

Polettini, N., 2004. *The Vector Space Model in Information Retrieval – Term Weighting Problem.* Povo: University of Trento. (Manuscript)

Pressman, R.S., 2010. *Software Engineering : A Practitioners Approach.* (7th Edition). New York : McGraw-Hill.

Ramos, J., 2003. *Using TF-IDF to Determine Word Relevance in Document Queries*. The First instructional Conference on Machine Learning (iCML-2003), 3-8 December 2003 , Piscataway, NJ USA

Shneiderman, B., C. Plaisant, 2010. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. (5th Edition). Boston: Addison-Wesley.

Sommerville, I., 2007. *Software Engineering.* (8th Edition). Harlow: Pearson Education Limited.