INFORMATION RETRIEVAL TUGAS AKHIR DAN PERHITUNGAN KEMIRIPAN DOKUMEN MENGACU PADA ABSTRAK MENGGUNAKAN *VECTOR SPACE MODEL*

Putri Elfa Mas`udia
Jurusan Teknik Elektro, Program Studi Teknik Telekomunikasi
Politeknik Negeri Malang
Email: putri.elfa@polinema.ac.id

Martono Dwi Atmadja
Jurusan Teknik Elektro, Program Studi Teknik Telekomunikasi
Politeknik Negeri Malang
Email: martonce@yahoo.com

Lis Diana Mustafa
Jurusan Teknik Elektro, Program Studi Teknik Telekomunikasi
Politeknik Negeri Malang
Email: lisdianamustafa16@gmail.com

ABSTRAK

Pencarian pada database yang biasa dilakukan mahasiswa hanya mampu mencari judul yang sesuai berdasarkan kata kunci yang diinputkan, misalnya, jika kata kunci yang dimasukkan adalah "sistem cerdas" maka akan ditampilkan semua dokumen yang mengandung kata "sistem cerdas" namun sistem tidak bisa mengukur mana dokumen yang paling mirip. Untuk dapat melakukan pencarian berdasar substansi yang paling mirip, terdapat teknologi yang disebut *information Text Retrieval*. Dalam penelitian ini akan dikembangkan suatu sistem temu kembali informasi judul tugas akhir dan perhitungan kemiripan dokumen menggunakan *vector space model*. Sistem secara otomatis akan melakukan *indexing* secara *offline* dan temu kembali *(retrieval)* secara *real time*. Proses retrieval dimulai dengan mengambil query dari pengguna, menerapkan *stop word removal* sehingga dihasilkan *keyword* yang *compaq* tetapi dapatmewakili query tersebut, kemudian sistem menghitung kemiripan antarakeyword dengan daftar dokumen yang diwakili oleh term-term di dalam index. Dokumen akan ditampilkan diurutkan berdasarkan dokumen yang paling mirip.Dari hasil pengujian terlihat ketika *keyword* "android" dimasukkan maka akan tampil empat dokumen yang diurutkan sesuai tingkat kemiripannya, yaitu docId 3 dengan tingkat kemiripan 0.9512, docId 4 dengan tingkat kemiripan 0.5020, docId 2 dengan tingkat kemiripan 0.2671, docId 8 dengan tingkat kemiripan 0.1522.

Kata kunci: temu kembali, *vector space model, stopword.*

ABSTRACT

Querying data from the database with SQL syntax will produce results that exactly match with the condition specified in where clause. It will not be able to determine the other result that does not contain the condition specified in where clause, even it may be subtantially similar with the condition. It will not be able to determine the value of similarity of each result with the condition searched. Information retrieval brings the solution to overcome this problem.This research is aimed to develop information text retrieval for the title of the final project and calculate the similarity between document using vector space model. The system will automatically indexecah terms inside document on offline mode and will retrieve information on real time mode. The retrieveal process will begin by taking query from user, then removingthe stop word . Then the system will calculate the similarityof query with the documents represented by index value of each term.. The output will show the the most relevant documentthat has the biggest simalirity value, followed by the other adocuments that have smaller similarity value. From the testing by applying "android" keyword, the system shows four documents ordered by simalirty value descending. Document with docId 3 has simalirity value 0.9512, docId 4 has simalirity value 0.5020, docId 2 has simalirity value 0.2671 and docId 8 has simalirity value 0.1522.

Keywords: *information retrieval, vector space model, stopword.*

1. PENDAHULUAN

Tugas akhir merupakan prasyarat yang harus dipenuhi mahasiswa Politeknik Negeri Malang untuk lulus dan memperoleh gelar Amd. Mahasiswa biasanya mencari referensi judul dari jurnal atau tugas akhir dari kakak tingkat sebelumnya yang berhubungan dengan ide yang akan diajukan. Dalam proses pengajuan judul proposal, mahasiswa harus memastikan bahwa judul yang akan diajukan belum pernah diajukan sebelumnya. Tidak hanya kemiripan judul saja, tetapi juga kemiripan konten, metode yang digunakan, dan studi kasus. Hal ini bertujuan untuk menghindari adanya plagiasi.

Pencarian pada database yang biasa dilakukan mahasiswa hanya mampu mencari judul yang sesuai berdasarkan kata kunci yang diinputkan, misalnya, jika kata kunci yang dimasukkan adalah "sistem cerdas" maka akan ditampilkan semua dokumen yang mengandung kata "sistem cerdas" namun sistem tidak bisa mengukur mana dokumen yang paling mirip. Untuk dapat melakukan pencarian berdasar substansi yang paling mirip, terdapat teknologi yang disebut *information Text Retrieval*. *Information text retrieval* adalah salah satu metode yang digunakan untuk menyimpan data dengan cara memprosesnya (menghilangkan *stop word*) dan menyimpan tiap kata beserta informasi dari kata tersebut (letak kata, jumlah bobot, dll). *Information retrieval* berfokus pada proses yang terlibat di dalam representasi, media penyimpanan, mencari dan menemukan informasi yang relevan dari informasi yang diinginkan oleh user. [6]

Dari latar belakang diatas, maka penulis ingin mengembangkan suatu sistem temu kembali informasi judul tugas akhir dan perhitungan kemiripan dokumen menggunakan *vector space model*. Sistem secara otomatis akan melakukan *indexing* secara offline dan temu kembali *(retrieval)* secara real time. Proses retrieval dimulai dengan mengambil query dari pengguna, menerapkan *stop word removal* sehingga dihasilkan *keyword* yang *compaq* tetapi dapatmewakili query tersebut, kemudian sistem menghitung kemiripan antarakeyword dengan daftar dokumen yang diwakili oleh term-term di dalam index. Dokumen akan ditampilkan diurutkan berdasarkan dokumen yang paling mirip.

Adapun rumusan masalah untuk penelitian ini adalah bagaimana merepresentasikan dokumen dan query menggunakan algoritma Tf/Idf ? bagaimana merancang sistem temu kembali untuk pencarian judul tugas akhir mahasiswa? Dan bagaimana perhitungan kemiripan dokumen dengan menerapkan algoritma Tf/Idf dan menggunakan *vector space model* ?

Dwija Wisnu [14] melakukan penelitian yang berjudul Perancangan *information retrieval* untuk pencarian ide pokok teks artikel berbahasa inggris dengan pembobotan vector space model. Dalam penelitiannya peneliti memanfaatkan *information retrieval* pada text mining untuk menemukan ide pokok dalam teks pada artikel berbahasa inggris untuk membantu pembaca untuk lebih mudah memahami isi artikel dan menghemat waktu yang dibutuhkan.

Cholifatul [3] melakukan penelitian yang berjudul Aplikasi *Information retrieval* untuk pembentukan thesaurus berbahasa Indonesia. Penelitian ini bertujuan untuk membangun perangkat lunak yang mampu menentukan thesaurus dari kata berbahasa Indonesia di bidang teknologi informasi dan komputer.

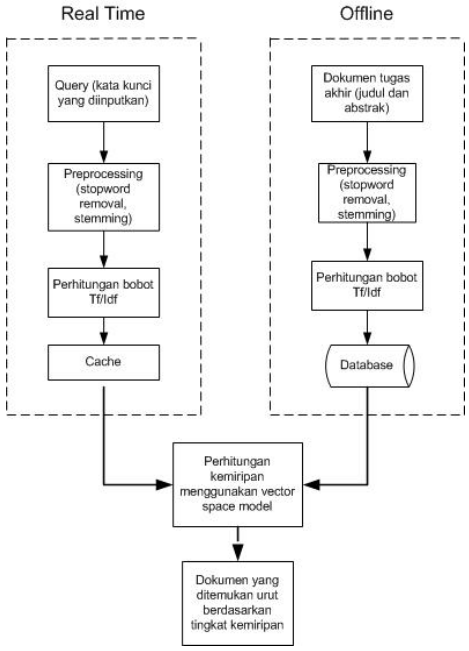
2. METODOLOGI PENELITIAN

2.1 Data

Data yang dibutuhkan dalam penelitian ini adalah data judul tugas akhir mahasiswa beserta abstraknya. Data tersebut berasal dari tugas akhir mahasiswa Teknik Telekomunikasi Politeknik Negeri Malang. Data yang digunakan untuk pengujian adalah 25 data.

2.2 Metode Pengolahan Data

Metode pengolahan data yang akan dilakukan pada penelitian ini terbagi menjadi dua, yaitu pengolahan data *offline* dan *real time*. Data yang diproses secara *offline* adalah data judul tugas akhir dan abstraknya yang kemudian dimasukkan dalam database, proses ini disebut dengan *indexing*. sedangkan pengolahan data Query dilakukan secara real time. Proses pengolahan data ditunjukkan pada Gambar 1.



Gambar 1. Proses Pengolahan Data

Gambar 1 menunjukkan proses Pengolahan Data secara offline dan Real time dengan tahapan Sebagai berikut:

2.2.1 Pengolahan Data Offline

1) Data yang terkumpul akan dilakukan *preprocessing. Preprocessing* meliputi penghilangan kata yang dianggap tidak penting *(stopword)* dan dilakukan *stemming*, yaitu mengubah kata ke bentuk dasarnya dengan cara menghilangkan imbuhan awal maupun imbuhan akhir. Dari proses ini akan dihasilkan daftar kata atau term yang lebih *compaq* tetapi tetap mewakili dokumen yang sedang diproses
2) Setelah dilakukan *preprocessing*, maka langkah selanjutnya adalah mengambil tiap kata/term dan menghitung jumlah kemunculannya pada dokumen tertentu.
3) Dilakukan pembobotan kata menggunakan rumus Tf/Idf.

$$W_{ij} = tf_{ij} * \log\left(\frac{N}{df_j}\right) \quad (1)$$

Dimana W_{ij} adalah bobot term j pada dokumen i
Tf_{ij} adalah frekuensi term j pada dokumen i
N adalah jumlah total dokumen yang dikoleksi
Df_j adalah jumlah dokumen yang mengandung term j

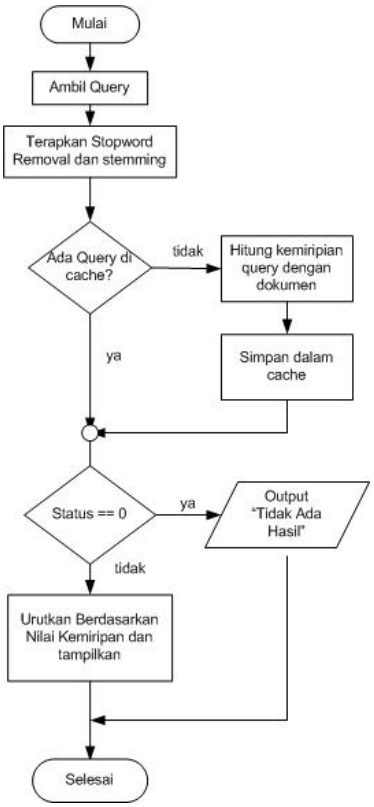
4) Tahapan indexing dilakukan untuk menyimpan tiap kata/term ke dalam database dengan atribut jumlah kemunculan dan bobot tiap term.

2.2.2 Pengolahan Data Real Time

Query yang dimasukkan oleh user juga akan diolah melalui beberapa proses yaitu :

1) Melakukan preprosesing terhadap *query* yang dimasukkan user yaitu menghilangkan stopword.
2) Setelah dilakukan *preprocessing*, maka langkah selanjutnya adalah mengambil tiap kata/term dan menghitung jumlah kemunculannya pada dokumen tertentu.
3) Dilakukan pembobotan kata menggunakan rumus Tf/Idf.
4) Tahapan indexing dilakukan untuk menyimpan tiap kata/term ke dalam database beserta bobot tiap term. Hal ini dilakukan dengan tujuan supaya *query* dengan kata yang sama tidak perlu dilakukan perhitungan lagi.

Setelah data selesai diolah secara offline dan realtime, maka akan dilakukan perhitungan similaritas (kemiripan) antara query permintaan user dengan dokumen yang tersimpan dalam database. Perhitungan dilakukan menggunakan *vector space model*. Kemudian hasilnya akan ditampilkan beberapa dokumen yang relevan dengan query secara urut berdasarkan kemiripan. Secara umum proses retrieval ditunjukkan dalam Gambar 2.

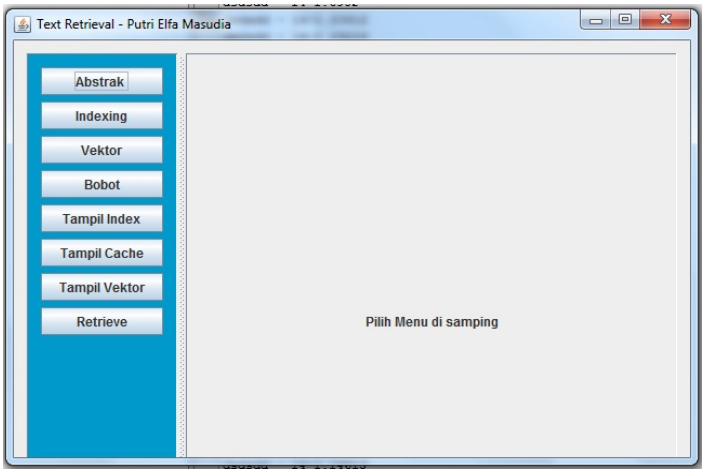


Gambar 2. Proses Retrieval

3. HASIL DAN PEMBAHASAN

3.1 Tampilan Umum Sistem

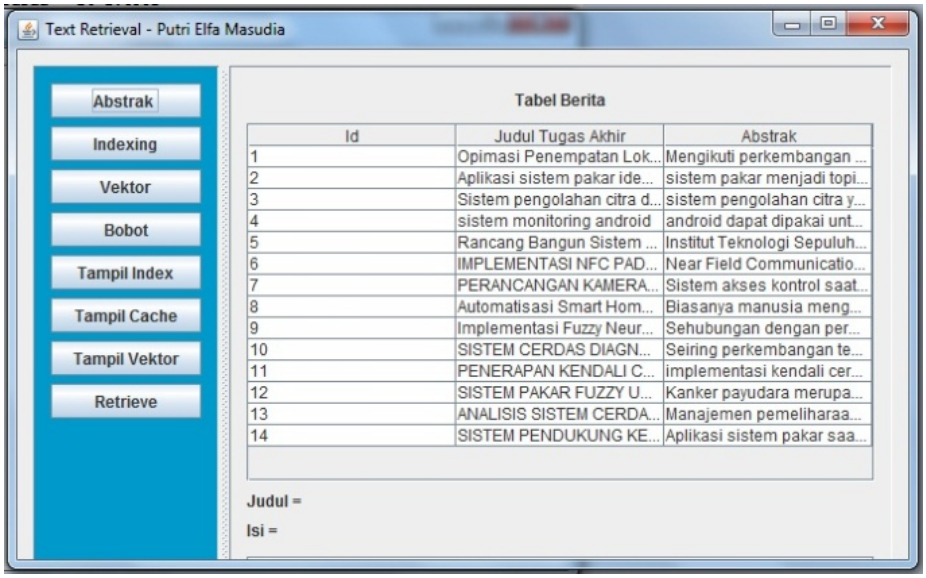
Pada sistem ini terdapat 6 sub menu, yaitu Abstrak ,indexing, vektor, bobot, Tampil index, Tampil cache, Tampil vector, Retrieve, untuk proses pencarian. Adapun menu utama ditunjukkan pada Gambar 3.



Gambar 3. Tampilan Umum Sistem

3.2 Tampilan Abstrak

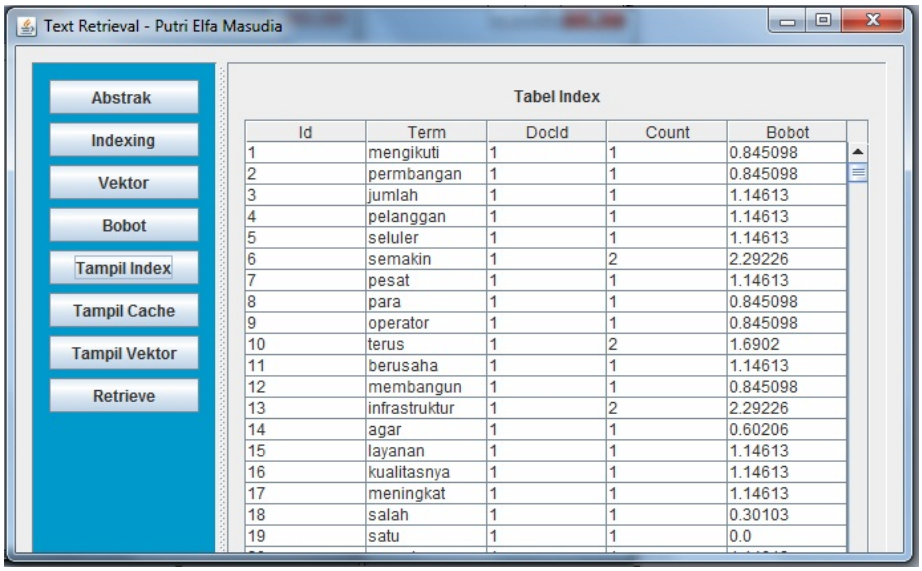
Menu abstrak ini digunakan untuk menampilkan semua data yang telah dimasukkan ke dalam database. Format data yang dimasukkan adalah id, judul tugas akhir, dan abstrak. Menu Abstrak dapat dilihat pada Gambar 4.



Gambar 4. Tampilan Daftar Abstrak

3.3 Tampil Index

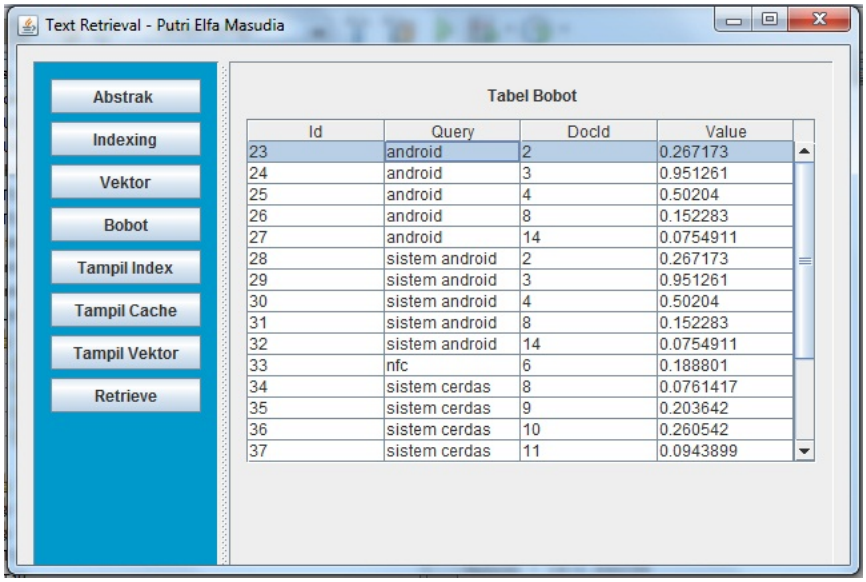
Pada menu indexing dilakukan proses untuk memasukkan tiap term kedalam tabel index. Hasil dari perhitungan indexing terdapat pada menu Tampil index yang ditunjukkan pada Gambar 5



Gambar 5. Tampilan Tabel Index

3.4 Tampil Cache

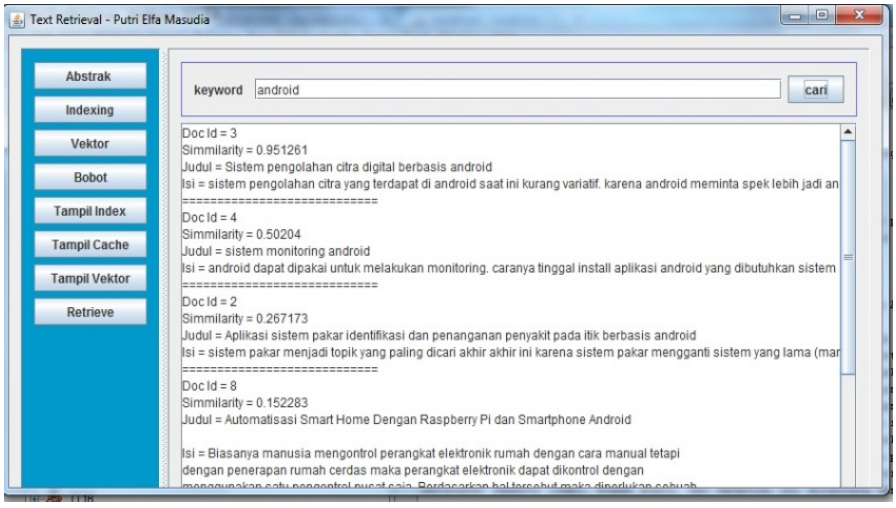
Menu tampil cache digunakan untuk menampilkan tabel cache, table cache adalah tabel yang menyimpan *keyword* yang sudah pernah dicari/dimasukkan. Jadi jika ada proses pencarian dengan keyword serupa, maka sistem tidak perlu menghitung ulang. Menu Tampil cache ditunjukkan pada Gambar 6



Gambar 6. Menu Tampil Cache

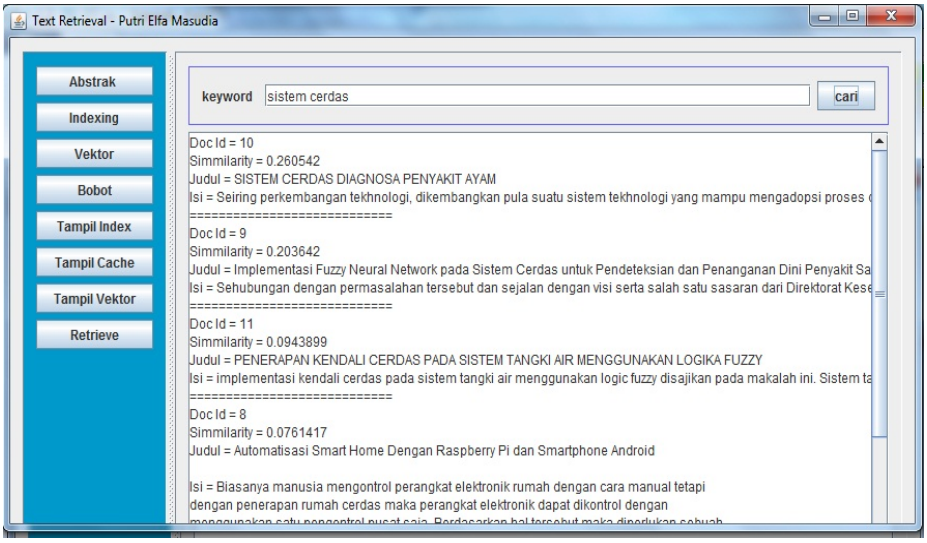
3.5 Retrieve

Menu retrieve digunakan untuk melakukan pencarian dokumen, caranya tinggal mengetikkan keyword yang akan dicari. Keyword bisa satu kata atau lebih, kemudian klik tombol cari. Hasil pencarian ditunjukkan pada Gambar 7



Gambar 7. *Retrieve Keyword* Android

Pada saat user mengetikkan keyword android dan klik tombol cari, maka sistem akan melakukan perhitungan kemiripan dokumen berdasarkan keyword. Hasilnya ditampilkan 4 dokumen yang diurutkan berdasarkan tingkat kemiripan yaitu DocId 3, DocId 4, DocId 2 dan DocId 8. Keempat dokumen ini membahas tentang android, akan tetapi DocId 3 membahas lebih banyak tentang android dibanding dokumen yang lain.



Gambar 8. Retrieve Keyword Sistem Cerdas

Keyword juga dapat diisi lebih dari satu kata, misal sistem cerdas. Pada retrieve dengan keyword sistem cerdas, terdapat 4 dokumen yang ditampilkan. Dokumen terakhir yang ditampilkan memang tidak terdapat *keyword* "sistem cerdas" pada judulnya.akan tetapi pada abstrak terdapat *keyword* "cerdas" oleh karena itu sistem tetap bisa menampilkan walau tingkat kemiripannya kecil.

4. KESIMPULAN

Dari perancangan dan implementasi yang telah dilakukan, maka dapat dibuat kesimpulan sebagai berikut :

1) Representasi dokumen dilakukan dengan menerapkan preprocessing, yaitu menghilangkan kata-kata yang tidak penting (*stopword*), kemudian dilakukan *indexing,* yaitu menghitung bobot (Tf/Idf) tiap term kemudian dimasukkan dalam tabel index sebagai representasi dokumen. Query yang dimasukkan oleh user juga akan diproses dengan cara yang sama (direpresentasikan dahulu) sebelum di hitung tingkat kemiripannya.
2) Proses perhitungan kemiripan dokumen dilakukan dengan mengetikkan *keyword, keyword* bisa terdiri dari satu kata atau lebih. *Keyword* yang dimasukkan juga akan diproses sama seperti dokumen. Kemudian dihitung tingkat kemiripan keyword dengan abstrak yang sesuai menggunakan rumus *vector space model* (VSM).
3) Dari hasil pengujian terlihat ketika keyword "android" dimasukkan maka akan tampil empat dokumen yang diurutkan sesuai tingkat kemiripannya, yaitu docId 3 dengan tingkat kemiripan 0.9512, docId 4 dengan tingkat kemiripan 0.5020, docId 2 dengan tingkat kemiripan 0.2671, docId 8 dengan tingkat kemiripan 0.1522

Terdapat banyak metode untuk klasifikasi, diharapkan untuk para pengembang dapat menggunakan metode tersebut untuk objek yang sama dan membandingkan metode klasifikasi mana yang paling baik dalam kasus klasifikasi tugas akhir untuk menentukan dosen pembimbing.

DAFTAR PUSTAKA

[1] Baeza, Yates and Ribeiro, Neto. 1999. *Modern Information retrieval*. Harlow. Addison-Wesley.
[2] Bunafit, Nugroho. 2008, Aplikasi *Pemrograman Web Dinamis Dengan PHPdan MySQL*. Gava Media. Yogyakarta.
[3] Cholifah, 1978. "Aplikasi Informasi Retrieval Untuk Pembentukan Tesaurus Berbahasa Indonesia Secara Otomatis".*Scan vol II no.1, ISSN : 1978-0087*
[4] Frakes, W.B. dan Baeza. R. 1992, *Information retrieval Data Structure and Algorithms*, New Jersey : Prentice Hall.
[5] Grossman, D., 1992, *IR Book*, http://www.ir.iit.edu/~dagr/cs529/files/ir_book/ [7 Maret 2002]
[6] Ingwersen, P, 1992, *Information retrieval Interaction*, London, Taylor Graham Publishing. http://www.db.dk/pi/iri [29 Agustus 2005]
[7] Rijsbergen, C.J. van., 1979, *Information retrieval, Second Edition*. Butterworths, London.

[8] Salton, G., 1989, *Automatic Text Processing : The transformation, Analysis, and Retrieval Information by Computer*, Massachusetts, Addison-Wesley.
[9] purwanti, endah,. 2015. "Klasifikasi Dokumen Temu Kembali dengan K-Nearest Neighbour". E-ISSN 2442-5168, Vol 1, No.1.
[10] Salton, G. & Buckley, C., 1987, *Term Weighting Approaches in Automatic Text Retrieval*, *Technical* Report No. 87-881, Departement of Computer Science Cornell University Ithaca, New York.
[11] Turney, P.D. Pantel, Patrick, 2010, "From Frequency To Meaning : Vector Space Model For Semantic" *Journal of Artificial Intelligence Reseach,* Vol 37, pp.141-188
[12] Welling, Luke and Thomson, Laura, 2001, *PHP and MySQL Web Development*.1st Edition. United States of America :Sams Publishing
[13] Witten et all, 1999, *Managing Gigabytes: Compressing and Indexing Document dan Images Second Edition*, San Fransisco, Morgan Kaufmann Publishers.
[14] Wisnu, dwija & Hetami, Anandhini. 2015," Perancangan *Information retrieval* (IR) untuk Pencarian Ide Pokok Teks Artikel Berbahasa Inggris Dengan Pembobotan *Vector Space Model".Jurnal ilmiah Teknologi dan Informasi ASIA,* Vol 9 No.1.