APLIKASI DETEKSI KEMIRIPAN TUGAS PAPER

Anthony Anggrawan1, Azhari2,

1Tenaga Pengajar Teknik Informatika STMIK Bumigora Mataram
2Mahasiswa Teknik Informatika STMIK Bumigora Mataram
Jl Ismail Marzuki, Mataram, Lombok, NTB
[1]anthony.anggrawan17@gmail.com, [2]harrie_strong@yahoo.co.id

ABSTRACT

Information searching based on users' query, which is hopefully able to find the documents based on users' need, is known as Information Retrieval. This research uses Vector Space Model method in determining the similarity percentage of each student's assignment. This research uses PHP programming and MySQL database. The finding is represented by ranking the similarity of document with query, with mean average precision value of 0,874. It shows how accurate the application with the examination done by the experts, which is gained from the evaluation with 5 queries that is compared to 25 samples of documents. If the number of counted assignments has higher similarity, thus the process of similarity counting needs more time, it depends on the assignment's number which is submitted.

Keyword : Detection, Information Retrieval, Similarity, Vector Space Model.

I. PENDAHULUAN

Kemajuan Teknologi Informasi pendidikan pada masa sekarang ini berkembang dengan begitu pesat. Menurut Boediono (2012) dalam kompas.com (2012), pendidikan merupakan kunci pembangunan bangsa karena pendidikan mempunyai peranan yang besar dalam pembangunan suatu bangsa.

Dosen dihadapkan suatu masalah pada penilaian kualitas tiap mahasiswa sesuai dengan kriteria yang diinginkan kampus. Tugas paper merupakan salah satu bagian yang dijadikan acuan penilaian studi mahasiswa. Dengan berkembangnya sistem informasi, sudah banyak tugas yang di kumpulkan secara *softcopy*yang membutuhkan waktu yang cukup lama untuk pemeriksaan terlebih lagi jumlah mahasiswa yang diajar tidak sedikit.

Bisa terjadi sebuah karya paper/ilmiah merupakan hasil plagiat,baik yang dilakukan secara secara sengaja ataupun tidak sengaja dengan mengutip sebagian atau seluruh karya dan/atau karya ilmiah orang lain, tanpa menyatakan sumber secara tepat dan memadai.

Sebuah karya/tulisan dapat diketahui berapa besar plagiatnya dapat dideteksi dengan prinsip membandingkan dengan karya yang lainnya. Alasan inilah maka dalam studi ini dibangun sebuah Sistem Deteksi Kemiripan Paper atau tuliasan atau karya dengan menggunakan Metode *Vector Space Model,* di mana Model *VectorSpace* adalah "Model dalam IR (*Information Retrival*) yang berbasis *token* untuk memungkinkan *partial matching* dan pemeringkatan dokumen (pengindexan)". [3]

Adapun dokumen yang diuji tingkat persentase kemiripannya berupa *filepdf*, dimana *proses* deteksi plagiarismenya melalui tahapan *preprocessing*, yaitu

prosespenghapusan *stopword*, dan *stemming*dan selanjutnya dilakukan perhitungan pembobotan dan *cosine similarity*. Tujuan utama sistem ini adalah untuk mengetahui tingkat kemiripan atau plagiat suatu tugas paper. Aplikasi dari studi ini diharapkan mampu mendeteksi dan memberikan persentase kemiripan tugas paper dari proses tindakan plagiarisme mahasiswa. Aplikasi ini nantinya seseorang dapat dengan mudah memeriksa hasil dari sebuah paper persentase hasil yang di berikan program.

Studi ini uji coba dilakukanpadatugas paper mahasiswa STMIK Bumigora. Sistem yang dibangun bersifat *multiuser* berbasis Web dengan gunakan bahasa pemrograman *PHP* dan *databaseMySQL*. Paper yang diuji cobakan berupa *filePDF* dan berbahasa indonesia. Algoritma *Steaming* menggunakan sastrawi dan teknik pembobotan term menggunakan TF-IDF(*Term Frekuensi – InverseDokumen Frekuensi*). Adapun proses perhitungan Kemiripan Menggunakan *cosine similarity* yaitu dalam persentase kemiripan

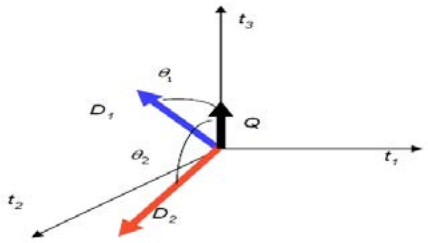
dan metode evaluasi yang digunakan *precision dan recall.*

Secara umum, manfaat dari studi ini adalah pertama, terciptanya sebuah AplikasiPendeteksi Kemiripan untuk menentukan tingkat kemiripan tugas paper di STMIK Bumigora berbasis *WEB*; kedua, menerapkanmetode*VectorSpaceModel* (VSM) untuk menghasilkan karya aplikasi Pendeteksi Kemiripan Paper.

II. METODOLOGI

IR (*Information Retrieval*) adalah menemukan matrial (yang biasanya berbentuk dokumen) dari *domain* yang tidak terstruktur (biasanya berbentuk teks) berupa kebutuhan informasi yang memuaskan dari koleksi yang besar. Istilah data tidak terstruktur berhubungan dengan data yang tidak jelas, bersifat *simantik*, mudah untuk dipahami oleh struktur komputer. Pencari informasi kembali sebenarnya sudah lama terjadi dalam proses manual. Contoh nyata dari proses pencarian kembali informasi adalah ketika seorang pegawai yang bekerja sebagai kasir yang memberikan gaji pada pegawai lain mencari informasi jam kerja dari pegawai lain melalui absensi harian. Contoh lain adalah seorang mahasiswa yang mencari bahan untuk penelitian di dalam perpustakaan. Perkembangan dunia jaringan yang dapat menciptakan hubungan antar negara melalui Internet ikut berperan akan kebutuhan pencarian informasi karena informasi begitu mudah didapatkan karena seakan-akan dunia telah menjadi satu dengan adanya *Internet*. [2]

Vector Space Model (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan *(similarity)* term dengan cara pembobotan *term*. Dokumen dipandang sebagi sebuah vektor yang memiliki *magnitude* (jarak) dan *direction* (arah). Pada *Vector Space Model,* sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang *vektor*. *Relevansi* sebuah dokumen ke sebuah *query* didasarkan pada similaritas diantara *vektor* dokumen dan *vektorquery*. [1]



Gambar 1. Ilustrasi *VektorSpaceModel* [1]

dimana

ti = Kata di *database*

D_i = Dokumen

Q = Kata Kunci

Cara kerja dari *vector space model* adalah dengan menghitung nilai *cosines* sudut dari dua *vektor*, yaitu *vektor* kata kunci terhadap *vektor* tiap dokumen. Perhitungan *vectorspacemodel* menggunakan persamaan (1),(2) dan (3)

$$Cosine\ \theta_{D_i} = Sim(Q, D_i)$$ 1

Q = query (kata kunci)

D_i = dokumen ke-i

$$Sim(Q, D_i) = \frac{\sum_j w_{i,j}\, w_{q,j}}{\sqrt{\sum_j w_{i,j}^2}\sqrt{\sum_i w_{q,j}^2}}$$ 2

D_i = dokumen ke-i

Q = query (kata kunci)

J = Kata diseluruh dokumen

$$Cosine\ \theta_{D_i} = \frac{Q \cdot D_i}{|Q| * |D_i|}$$ 3

dimana

D_i = dokumen ke-i

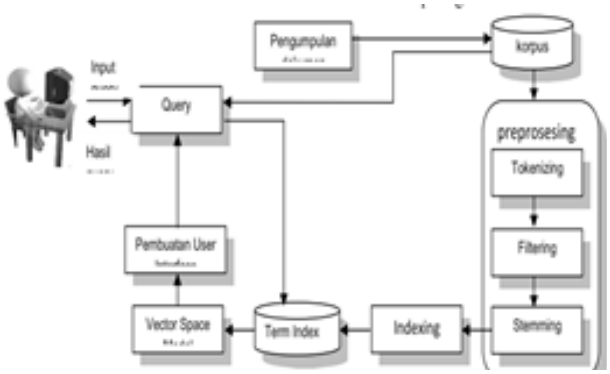
Q = query (kata kunci)

|Q| = Vektor Q

$|D_i|$ = Vektor Di

Sistem temu kembali informasi menggunakan metode *Vector Space Model* sebagai suatu sistem memiliki beberapa proses (modul) yang membangun system secara keseluruhan. Modul *system* temu kembali informasi terdiri dari : modul pengumpulan dokumen, modul tokenisasi (*tokenizing*), modul pembuangan *stopword* (*filtering*), modul Pengubahan kata dasar (*stemming*), modul Pengindeksan kata (*indexing*), modul *Vector Space Model* (*term similarity*) dan modul pembuatan *user interface.* [1].

Arsitektur sistem temu kembali informasi bisa dilihat pada gambar 2.3.



Gambar 2.3 Arsitekture Sistem Temu Kembali Informasi [1]

Pada gambar diatas dapat dijelaskan bahwa yang dilakukan oleh *user* adalah memasukkan *query* untuk mencari dokumen yang akan dicari. Dari query yang dimasukkan oleh user akan dilakukan pengindexan, dimana sebelumnya itu korpus dari kumpulan dokumen sudah dilakukan *prosessing* yang kemudian akan di cocokkan dengan *query* melalui metode *Vektor Space Model* untuk dilakukan proses pecocokan dari perhitungan yang sudah ada. Hasil yang diterma oleh *user* berupa file dokumen yang sudah terurut berdasarkan kemiripan *query* dengan dokumen.

III. HASIL DAN PEMBAHASAN

1. ***Indexing***

Sebuah dokumen elektronik biasanya berbentuk *file* yang didalamnya terdapat kumpulan kata. Kumpulan kata itu akan dibentuk sebuah pola pengenalan dokumen yang biasa disebut proses indexing. Proses *indexing* dilakukan dengan *tokenization*, *stopword* dan *stemming.* [2]

2. ***Tokenization***

Memberikan urutan karakter dan mendefinisikan unit dokumen, *tokenization* adalah mencacah kalimat kedalam bagian-bagian . Proses tersebut dimulai dengan membaca dokumen yang dimiliki, dilanjutkan dengandipecah perkata.

Berikut adalah contoh *tokenization*:

Kalimat didalam dokumen:

Saya sedang belajar *Information Retrieval*

Hasil *tokenization*:

Saya	sedang	belajar	*Information*	*Retrival*

Terlihat dari contoh diatas terdapat kalimat "Saya sedangbelajar *Information Retrieval*" kemudian proses *tokenization*dilakukan dengan memecah kata dalam kalimat tersebutmenjadi 5 pecahan yaitu saya, sedang, belajar, *Information* dan *Retrieval*. [2]

3. ***Stopword***

Dalam sebuah dokumen terdapat banyak kata yang bukan kata kunci di dalam dokumen atau kata-kata tambahan hanya untuk menghubungkan kata, contohnya adalah kata penghubung dan juga terdapat tanda-tanda baca. Dalam proses *indexing* dilakukan proses untuk menghilangkan kata-kata tersebut untuk mengurangi proses peng-*index*-an dan mengurangi kata-kata dan tanda baca yang nantinya tidak berkaitan langsung dengan kata kunci. Selain untuk mengungari proses *indexing* proses tersebut dilakukan agar penerapan perhitungan kesamaan dokumen dengan dokumen yang dicari terdapat kesesuaian karena berkurangnya *noise* kata penghubung dan tanda baca yang jika tidak dihilangkan akan masuk kedalam perhitungan. [2]

4. ***Stemming***

Stemming adalah proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya (stem).*Stemming* akan menghilangkan kata imbuhan pada kata-kata sehingga yang terbentuk adalah kata dasarnya saja.

5. ***Algoritma TF-IDF(TermFrequensi- Inversed Dokumen Frequensi)***

Penggunaan algoritma tfidf dalam proses hierarchical template maching **:** TFIDF (*Term Frekuensi Inverse Document Frequency*) dikenal sebagai algoritma yang didasarkan pada nilai statistik kemunculan suatu *template* dalam dokumen. [4]

6. ***Cosine Similarity***

Pada metode *cosine similarity*, semakin besar sudut antara dua koordinat kata maupun dokumen yang dihitung, maka semakin kecil kemiripan antara dua kata maupun dokumen yang dihitung tersebut. Sedangkan jika semakin kecil maka semakin besar kemiripannya. Metode *cosine similarity* bekerja dengan cara menghitung nilai kosinus dari kedua sudut koordinat kata maupun dokumen pada sebuah dimensi. [5]

Dalam imlementasi *VektorSpaceModel* tersebut telah dibuat alur dari Aplikasi Deteksi Kemripan Tugas paper sebagai berikut :

1. Tugas paper dikumpulkan dalam bentuk file PDF.

   Hanya tugas *file* dalam bentuk *file* PDF yang dapat di proses oleh aplikasi, mahasiswa hanya akan mengupload *file* dalam bentuk PDF.

2. Proses konversi oleh sistem dari *file* PDF ke bentuk teks.

Dari *file* PDF yang dikumpulkan oleh mahasiswa akan dilakukan konversi ke bentuk teks yang bertujuan untuk dapat dilakukan proses selanjutnya, karena hanya bentuk teks yang dapat di proses untuk dilakukan tahapan selanjutnya

3. Teks Normalisa.

   Pada tahapan teks normalisasi dilakukan penghilangan tanda baca yang ada pada file PDF yang sudah diubah ke bentuk teks.

4. Tokenizing.

   Tekenizing melakukan proses pemecahan kalimat sehingga menjadi bagiaan-bagian yang sudah di jelaskan dalam implementasi Vektor Space Model.

5. Stopword.

   Stopword dilakukan menghilangkan tanda penghubung yang sudah ada dalam database sehingga kata-kata tersebut akan dicocokkan dalam database stopword apa bila kata itu sama dengan kata yang ada di stopword maka akan dilakukan proses penghapusan kata penghubung.

6. Stemming.

   Proses stemming digunakan untuk menghilangkan kata imbuhan sehingga menghasilkan kata dasar saja.

7. Term Dokumen Matriks.

   Berisikan kata yang sudah dilakukan proses stopword dan stemming.

8. TF-IDF.

   Perhitungan dimana jumlah kemuculan kata itu dalam dokumen itu seperti dijelaskan pada implementasi Vektor Space Model.

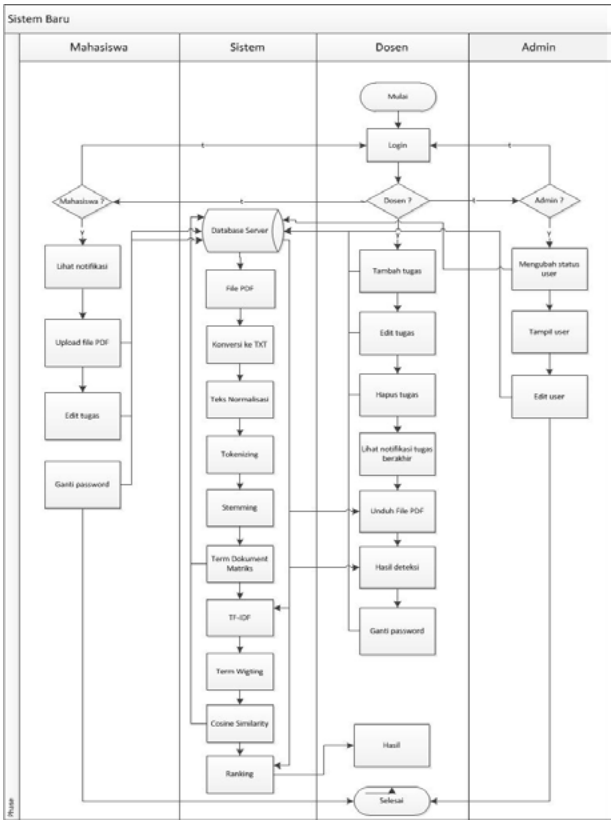
9. Cosine Similarity.

   Menghitung seberapa besar sudut antara kata kunci dengan dokumen apabila sudut dokumen lebih dekat dengan sudut kata kunci maka persentase semakin besar.

10. Perangkingan.

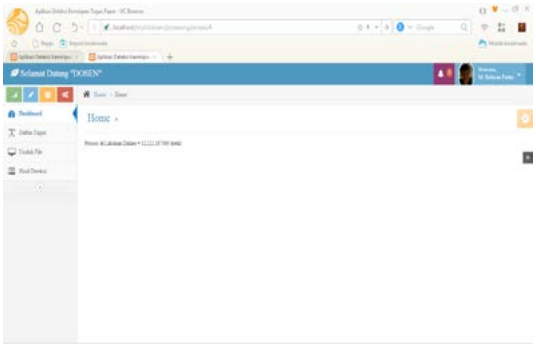
    Pengubahan nilai cosine ke dalam bentuk persentase sehingga dokumen dapat terurut sesuai dengan hasil persentasenya.

Alur Aplikasi Deteksi Kemiripan Tugas Paper dari penjelasan di atas bisa dilihat pada gambar 2.



Gambar 2. Alur Studi Kemiripan Tugas Paper

Penerapan *VektorSpace* Model yang sudah dibuat dalam bentuk aplikasi dapat dilihat pada gambar 3 sebagai berikut



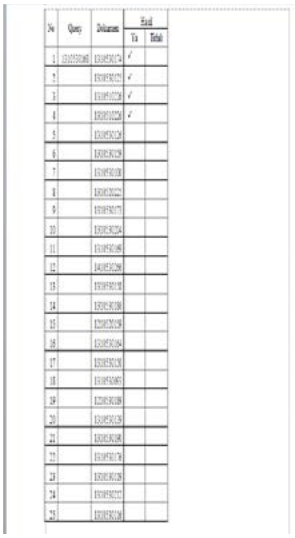
Gambar 3. Hasil Penerapan

*VektorSpace*Model

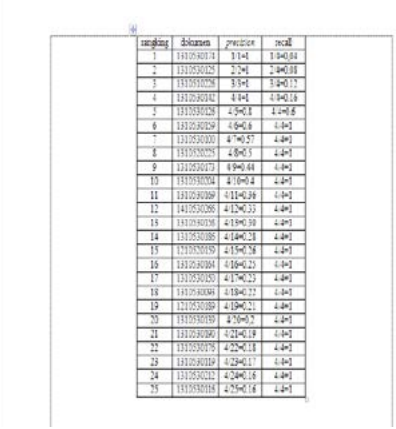
Dari 26 tugas paper yang dijadikan percobaan, dari 26 tugas paper tersebut 1 tugas paper yang urutan pertama menjadi kata kunci dan urutan 2-26 tugas paper menjadi dokumen, proses itu dilakukan berulang sehingga tugas paper yang urut 26 dibandingkan dengan tugas paper yang nomor urut 1-25.

Ujicoba yang telah dilakukan untuk mendatkan

hasil yang sesuai di bentuk sebuah table relepansi yang dapat dilihat di lihat pada tabel 1 sebagai berikut.



Tabel 1. Hasil Pemilihan dokumen relevan dengan query



Tabel 2. Perhitungan nilai *Precison* dan *Recal* dari table relevan

Ke dua tabel diatas adalah salah satu contoh dari 1 dokumen yang menjadi kata kunci dari 25 dokumen yang di ambil dari 26 tugas paper mahasiswa, dengan demikian nilai perhitungan dari masing-masing dokumen yang menjadi kata kunci akan mendapatkan nilai *precision* dan *recall* di mana nilai-nilai tersebuat akan dihitung nilai *mean average precision* untuk menghitung apakah aplikasi sedah sesuai dengan yang diharapkan.

Dalam menghitung nilai precision dan recall dari masing-masing dokumen dimana jumlah sampelnya ada 5 maka untuk nilai yang di ambil adalah nilai *precision* dimana dokumen yang relevan saja yang di ambil. Nilai *Mean average precision*= (1+0,96+0,91+0,75+0,75)/5=4,37/5=0,874. Nilai (1+0,96+0,91+0,75+0,75) didapatkan dari nilai rata-rata dari masing-masing kata kunci sedangkan nilai 5 adalah total dari kata kunci yang menjadi sample, dan nilai 0,874 adalah nilai dari rata-rata *precision* dari lima *query* menjadi sample terhadap 25 dokumen lainnya terhadap akurasi antara program dengan pakar yang mengoreksi secara manual.

IV. SIMPULAN DAN SARAN

Adapun simpulan dari hasil studi ini adalah:

1. Metode *vektorspace* model dapat di gunakan untuk mendeteksi tugas paper mahasiswa.
2. Aplikasi membutuhkan waktu yang lama tergantung jumlah tugas yang akan diproses hasil kemiripannya, dalam proses perhitungan kemiripan uji coba dilakukan 26 tugas dangan 30 tugas yang mempunyai selisih waktu±3 jam 20 menit dalam proses perhitungan kemiripan.
3. *Nilai Mean average precision* yang diperoleh adalah 0,874 yang digunakan untuk membandingkan seberapa akurat aplikasi terhadap pemeriksaan menggunakan aplikasi yang dibandingkan dengan pakar yang memeriksa secara manual, pada pengujian menggunakan 5 dokumen sebagai kata kunci terhadap 25 dokumen.

Diharapakan studi ini bermanfaat bagi berbagai pihak yang membutuhkan, dan diharapakan ada pengembangan lebih lanjut sebagai berikut:

1. Dalam proses *stemming* masih belum sempurna karena masih ada kata-kata yang tidak bisa di *stemming*
2. Mengembangkan aplikasi agar dapat menunjukkan bagian yang memiliki kemiripan berdasarkan persentase kemiripan yang dihasilkan.
3. Dalam proses normalisasi database masih ada redudansi, apabila redudansi di hilangkan maka tidak akan bisa melakukan proses perhitungan.
4. Secara umum hasil studi ini masih banyak keterbatasan dan kekurangan yang perlu untuk ditambahkan sehingga saran dan kritik sangat diharapkan demi penyempurnaan studi ini.

Daftar Pustaka

[1] F. Amin, "Implementasi Search Engine (Mesin Pencari) Menggunakan Metode Vector Space Model", hal. 45-58, Januari 2011.

[2] Sahrun Alim Tri Bawono, *Information Retrieval Meningkatkan Pencarian Data yang Relevan*, Yogyakarta: Universitas Gadjah Mada, 2014.

[3] Dinz. Information Retrieval (Methods, Recall and Precision, Web-Crawler), 2008. [Online]. Available : *http://catatan-dinz.net/riset-pengembangan/information-retrieval-methods-recall-and-precision-web-crawler/.* [Accessed : Jul. 17, 2015].

[4] Irwan Pahendra Anton Saputra, *Penggunaan Algoritma Tfidf Dalam Proses Hierarchical Template Maching* : Konferensi Nasional Sistem dan Informatika, November 2011..

[5] Michael J. Shaw, *E BUSINNESS MANAJEMENT.* New York: Kluwer Academic Publishers, 2002.