IDENTIFIKASI KEMIRIPAN JUDUL TUGAS AKHIR PSTI DAN PSMI DI POLITEKNIK NEGERI MALANG

Bariroh Isriya Nur Aini[1], Dwi Puspitasari[2], Yan Watequlis Syaifudin [3]

Jurusan Teknik Elektro, Program Studi Teknik Informatika, Politeknik Negeri Malang

bariroh.isriya@gmail.com, dwi_istari@yahoo.com, yan_ws@yahoo.com

Abstrak

Membuat tugas akhir atau skripsi merupakan syarat yang harus dipenuhi oleh mahasiswa untuk kelulusan. Tak jarang karya yang dihasilkan memiliki kemiripan baik dari judul ataupun isi. Hal ini menjadi permasalahan tersendiri karena sebaiknya tugas akhir atau skripsi memiliki judul yang berbeda supaya karya yang dihasilkan semakin beragam. Untuk mengatasi permasalahan diatas perlunya dilakukan seleksi saat akan mengajukan tugas akhir atau skripsi. Seleksi awal dapat dilakukan melalui judul yang akan diajukan. Seleksi dilakukan dengan cara melihat serta membandingkan antara judul yang diusulkan dengan judul yang sudah pernah diajukan sebelumnya. Dalam penelitian ini teknik yang digunakan untuk menyelesaikan masalah adalah dengan menggunakan teknik *text mining*. Teknik *text mining* merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks. Proses penganalisisan teks guna mendapat informasi yang bermanfaat untuk tujuan tertentu. Proses data mining untuk data dokumen atau teks memerlukan lebih banyak tahapan, mengingat data teks memiliki karakteristik yang lebih kompleks daripada data biasa.

Kata kunci : kemiripan judul,*text mining,* TF-IDF, *Vector Space Model*

1.Pendahuluan

Salah satu bentuk karya tulis ilmiah yaitu tugas akhir dan skripsi. Tugas akhir dan skripsi merupakan syarat yang harus dipenuhi oleh mahasiswa sebelum mengakhiri masa studi. Pembuatan tugas akhir dan skripsi diawali dengan pemilihan judul sebelum proses pengerjaan. Kemungkinan adanya kesamaan antara judul yang akan diajukan dengan judul yang telah ada cukup besar. Dengan terbatasnya informasi maka pemilihan judul menjadi sedikit sulit. Hal ini dapat sebenarnya diselesaikan dengan adanya seleksi judul yang akan diajukan. Dengan dilakukan seleksi judul ini diharapkan judul yang akan diajukan berbeda dengan judul yang telah ada. Dengan demikian hasil karya menjadi lebih beragam dan berkembang.

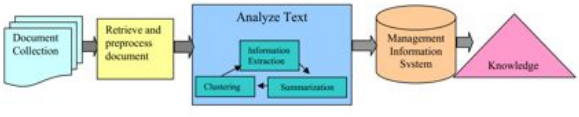
Teknik *text mining* diterapkan dalam tahap preproses. Sedangkan dalam tahap analisa untuk mengukur kemiripan antara satu judul dengan judul lainnya menggunakan kata kunci yang didapat dari inputan, dan algoritma yang digunakan adalah algoritma TF/IDF (*Term Frequency – Inversed Document Frequency*) dan algoritma Vector Space Model.

2.Metode

A. Text Mining

Text mining didefinisikan sebagai penemuan informasi baru yang belum diketahui sebelumnya secara terkomputerisasi dengan mengekstrak informasi dari beberapa sumber tertulis yang berbeda. *Text mining* digunakan untuk menangani data yang tidak terstuktur seperti email, dokumen, *ful- text,* file HTML dan lain sebagainya.

Text mining merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks. Proses penganalisisan teks guna mendapat informasi yang bermanfaat untuk tujuan tertentu. Proses data mining untuk data dokumen atau teks memerlukan lebih banyak tahapan, mengingat data teks memiliki karakteristik yang lebih kompleks daripada data biasa.



Gambar 2 Skema Text Mining

Pada proses *text mining* pertama dilakukan pengumpulan dokumen setelah itu *text mining tool* akan mendapatkan dokumen tertentu dan melakukan tahapan *preprocess* dengan mengecek format dan karakter. Tahap selanjutnya ialah ekstrak informasi dari dokumen yang ada. Pada umumnya proses ektrak informasi sama dengan seperti gambar, namun tak jarang proses sedikit berubah menyesuaikan dengan kebutuhan. Setelah proses ekstrak informasi, informasi akan disimpan dan dapat mengambil berbagai informasi yang dibutuhkan.

B. Preprocess

Tahapan preproses merupakan tahapan mempersiapkan semua dokumen yang dibutuhkan. Dalam hal ini dibutuhkan koleksi judul yang sudah pernah diajukan. Koleksi judul yang ada harus sudah diproses menggunakan teknik *text mining*. Adapun tahapan- tahapan yang ada adalah sebagai berikut:

a. Tokenizing

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

b. Filtering

Pada tahap ini dilakukan proses filter atau penyaringan kata hasil dari proses tokenizing, dimana kata yang tidak relevan dibuang. Proses ini menggunakan pendekatan stoplist. Yang termasuk stoplist adalah "yang", "di", "dari", dan lain-lain.

c. Stemming

Stemming adalah proses untuk menggabungkan atau memecahkan setiap varian-varian suatu kata menjadi kata dasar. Stem (akar kata) adalah bagian dari akar yang tersisa setelah dihilangkan imbuhannya (awalan dan akhiran).

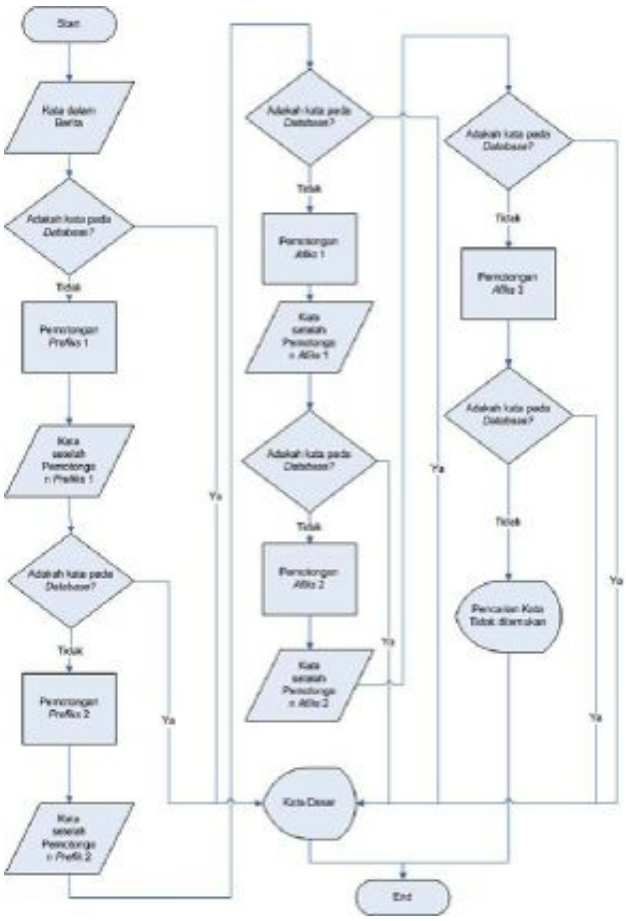
d. Tagging

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

e. Analizing

Pada tahap ini dilakukan proses perhitungan bobot (w) dokumen agar diketahui seberapa jauh tingkat similaritas antara keyword yang dimasukkan dengan dokumen.

Tahapan *preprocess* pada system ini dapat digambarkan seperti berikut:

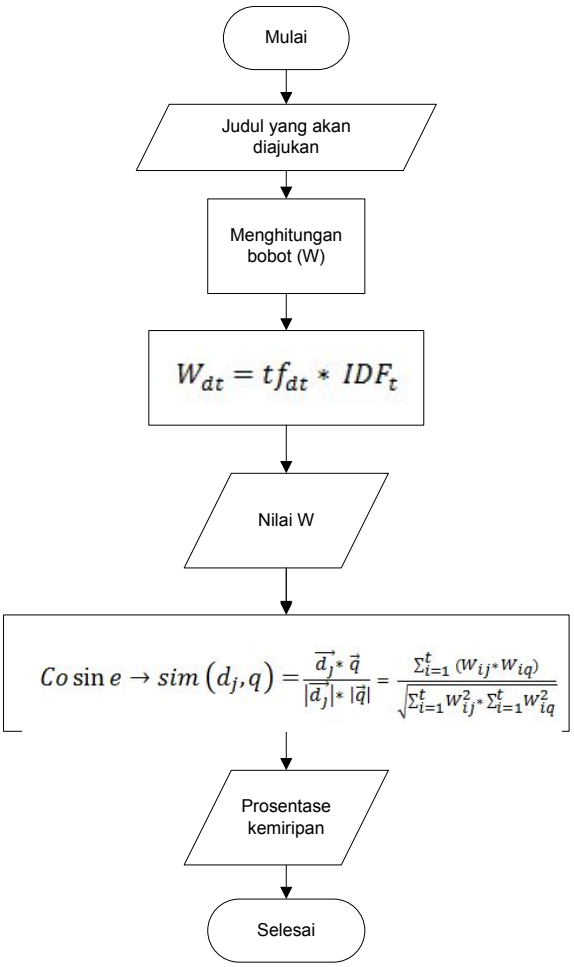


Gambar 3 Flowchart Stemming

C. Analisa

Proses analisa merupakan tahap terakhir dalam *preprocess*. Tahap ini menentukan seberapa jauh keterhubungan antar kata-kata antar judul yang ada. Dalam tahap ini akan dicari bobot tiap-tiap kata dari judul terhadap query yang dimasukkan. Tahap analisa ini menggunakan algoritma tfidf untuk pembobotan dan *Vector Space Model* untuk mengukur kemiripan.

Proses analisa dapat digambar seperti berikut:



Gambar 4 Flowchart analisa

1. TFIDF

Metode TF/IDF digunakan untuk menghitung bobot masing-masing dokumen terhadap kata kunci dengan formula :

$$W_{dt} = tf_{dt} * IDF_t$$

d = dokumen ke-d
t = kata ke-t dari kata kunci
W = bobot dokumen ke-d terhadap kata ke-t
tf = banyak kata yang dicari
IDF = *Inversed Dokumen Frequency*
IDF = $log_2\,(D/df)$
D = total dokumen
df = banyak dokumen yang mengandung kata yang dicari

Langkah awal yang dilakukan adalah penghitungan bobot judul yang akan diajukan dengan beberapa judul yang dianggap mendekati dengan judul yang telah ada. Setelah bobot diketahui maka penghitungan dilanjutkan menggunakan algoritma *Vector space model* (VSM) untuk mengukur tingkat kemiripan.

2. *Vector space model (VSM)*

Vector space model adalah suatu model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu query. Pada model ini, query dan judul dianggap sebagai vektor-vektor pada ruang n-dimensi, dimana n adalah jumlah dari seluruh term yang ada dalam *leksikon*. *Leksikon* adalah daftar semua term yang ada dalam indeks. Salah satu cara untuk mengatasi hal tersebut dalam model vector space adalah dengan cara melakukan perluasan vektor. Proses perluasan dapat dilakukan pada vektor query, vektor dokumen, atau pada kedua vektor tersebut.

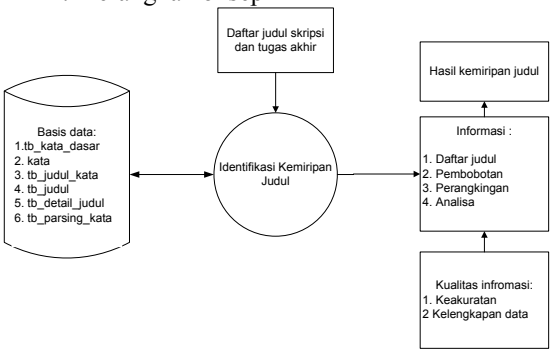
Pada algoritma vector space model digunakan rumus untuk mencari nilai cosinus sudut antara dua vector dari setiap bobot judul dan bobot dari query atau kata kunci. Formula yang digunakan adalah :

$$Co\sin e \rightarrow sim\,(d_j, q) = \frac{\vec{d_j} * \vec{q}}{|\vec{d_j}| * |\vec{q}|} = \frac{\Sigma_{i=1}^{t}(W_{ij} * W_{iq})}{\sqrt{\Sigma_{i=1}^{t} W_{ij}^2 * \Sigma_{i=1}^{t} W_{iq}^2}}$$

Dengan menggunakan *vector space model* hasil yang didapat akan lebih presisi dan dapat melakukan perangkingan.

D. Rancangan Sistem

1. Kerangka konsep



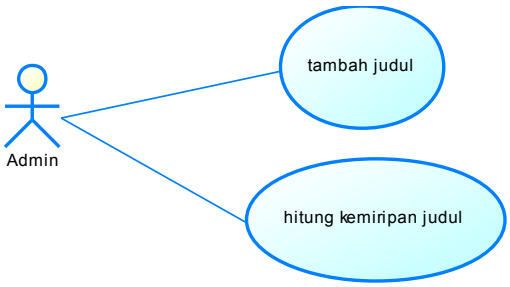
Gambar 2. Kerangka konsep

Sesuai dengan kerangka konsep diatas, diperlukannya daftar judul tugas akhir dan skripsi sebagai sumber data. Dan dibutuhkan table tb_katadasar, kata, tb_judul,

tb_judul_kata, td_detail_judul dan tb_parsing_kata untuk menampung dan mengolah data pada aplikasi atau system. Dari data yang terkumpul informasi yang didapat berupa daftar judul dan pembobotan, perangkingan dan analisa serta menghasilkan prosentase kemiripan judul.

E. Pemodelan Sistem

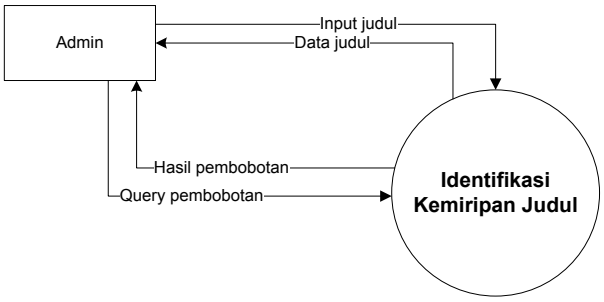
a. Use Case Diagram



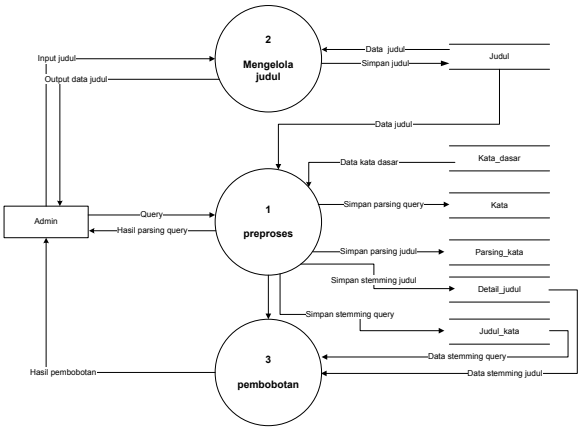
Gambar 3. Use case

Pelaku dalam sistem ini ialah admin dimana admin dapat melakukan proses tambah judul serta melakukan penghitungan kemiripan judul yang akan diajukan. Admin adalah panitia penyelenggara skripsi dan tugas akhir.

b. Data Flow Diagram



Gambar 4. DFD level 0



Gambar 5. DFD Level 1

3.Hasil

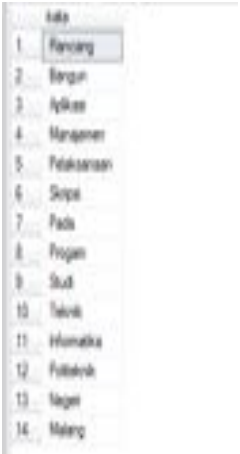
Setelah dilakukan pengujian terhadap system yang dibuat, diketahui bahwa sistem dapat berjalan lancar dan menghasilkan nilai yang dapat digunakan untuk perangkingan.

4.Pembahasan

A. Tahap Tokenisasi

Pada proses *preprocessing* dokumen terdapat tahapan tokenisasi. Tahap tokenisasi ialah tahapan pemotongan string inputan berdasarkan tiap kata yang menyusun. Sebagai contoh inputan berupa kalimat "Rancang Bangun Aplikasi Manajemen Pelaksanaan Skripsi Pada Progam Studi Teknik Informatika Politeknik Negeri Malang"

Hasil tokenisasi sebagai berikut:



Tabel 1. Hasil Tokenisasi

B. Tahap *filtering*

Tahap ini akan melakukan proses pengambilan kata-kata penting dari hasil tokenisasi. Dalam hal ini dapat menggunakan algoritma *stop list* (membuang kata yang kurang penting) atau *word list* (menyimpan kata yang penting)

Tabel 2. Hasil *filtering*

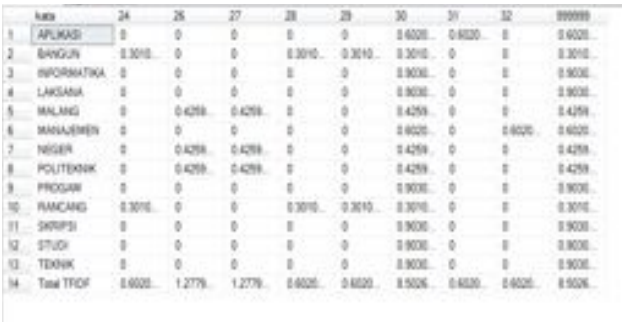
C. Tahap *Stemming*

Tahap ini merupakan tahapan mencari kata dasar dari tiap kata hasil tokenisasi.

Tabel 3. Hasil *Stemming*

D. Pembobotan

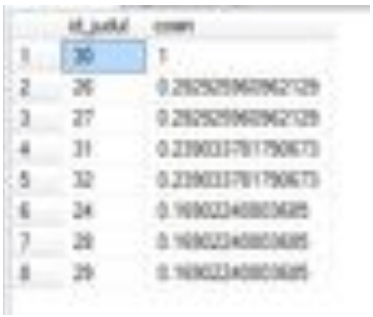
Pada algoritma TF/IDF digunakan rumus untuk menghitung bobot (W) masing-masing judul terhadap kata kunci dengan query 'Rancang Bangun Aplikasi Manajemen Pelaksanaan Skripsi Pada Progam Studi Teknik Informatika Politeknik Negeri Malang'. Hasil penghitungan nilai bobot *tfidf* adalah seperti berikut:



Tabel 4. Hasil penghitungan *tfidf*

E. Perangkingan

Nilai penghitungan bobot *tfidf* dari kata pada tiap-tiap judul dan query digunakan untuk menghitung nilai kemiripan dengan menggunakan algoritma *Vector Space Model.* Hasil penghitungan menggunaka *VSM* adalah seperti berikut:



Tabel 5. Hasil penghitungan *VSM*

Dari hasil yang didapat jika nilai penghitungan *VSM* besar maka judul tersebut memiliki tingkat kemiripan yang tinggi. Begitu juga sebaliknya, semakin rendah nilai penghitungan *VSM* maka tingkat kemiripan rendah. Jika nilai penghitungan VSM bernilai 1 maka dapat disimpulkan bahwa tingkat kemiripan 100%.

5.Kesimpulan dan Saran

A. Kesimpulan

Dari hasil implementasi system dapat disimpulkan bahwa :

1. Algoritma tfidf dapat diterapkan dalam tahap pembobotan kata pada judul terhadap query
2. Algoritma VSM dapat diterapkan dalam proses mengukur kemiripan antar judul dengan query serta melakukan perangkingan.
3. Semakin besar nilai pengukuran VSM, maka judul tersebut dapat dinyatakan mirip.

4. Nilai VSM tidak boleh lebih dari 1, jika nilai yang dihasilkan 1 maka dapat dinyatakan judul tersebut sama.

B. Saran

Saran penulis yang diusulkan untuk penelitian dan pengembangan selanjutnya adalah data pengujian yang digunakan dapat diperluas, tidak hanya menggunakan judul sebagai parameter mengukur kemiripan melainkan pada *abstrak* atau latar belakang dari skripsi dan tugas akhir.

6.Daftar Rujukan

Fan, Weiguo., et al.2005. Tapping The Power of Text Mining. (http://filebox.vt.edu)

Iwan Arif, *Text Mining*, http://lecturer.eepis-its.edu/~iwanarif/kuliah/dm/6Text%20Mining.pdf, [diakses pada Mei 2014]

Herwansyah , Adhit. Aplikasi Pengkategorian Dokumen Dan Pengukuran Tingkat Similaritas Dokumen Menggunakan Kata Kunci Pada Dokumen Penulisan Ilmiah Universitas Gunadarma.

Rosiani, Ulla Delfana. Puspitasari, Dwi. Andoko, Banni Satria. 2013: Penerapan Algoritma Nazief & Adriani Pada Preproses Sistem Pengukuran Tingkat Kemiripan Judul Tugas Akhir Di Program Studi Manajemen Informatika Politeknik Negeri Malang.

Ardi. 2010: Text Mining Untuk Akuisisi Pengetahuan Secara Otomatis Pada Sistem Pakar.