PENERAPAN ALGORITMA STEMMING NAZIEF & ADRIANI DAN SIMILARITY PADA PENERIMAAN JUDUL THESIS

Hafiz Ridha Pramudita

Magister Teknik Informatika STMIK AMIKOM Yogyakarta
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281
email : me@hafizridha.net

Abstract

Search information in the form of a document or a text known as Information Retrieval (IR) is a separation process documents that are relevant from a set of documents available. The increasing number of thesis documents are available allowing the similarity theme or topic of discussion was appointed as the thesis title. Stemming is a process that converts all the words in a text document into "rootword" or principal said. Rootwords are stored as an index. Each index which stored to be used as a comparison to the new document. By using the concept of Similarity, the index comparisons of old documents with new index documents can be obtained degree of similarity to certify the appropriateness of the thesis title.

Keywords :
Stemming, Simmilarity, Information Retrieval (IR)

Pendahuluan

Pencarian Informasi berupa teks atau dokumen yang dikenal dengan istilah *Invormation Retrieval (IR)* merupakan proses pemisahan dokumen-dokumen yang dianggap relevan dari sekumpulan dokumen yang tersedia. Semakin banyaknya jumlah dokumen *thesis* yang tersedia memungkinkan terjadinya kesamaan tema atau topik pembahasan yang diangkat sebagai judul *thesis.*

Algoritma *Stemming* adalah salah satu algoritma yang digunakan untuk meningkatkan performa IR dengan cara mentrasformasikan semua kata dalam teks dokumen kedalam *"rootword"* atau kata dasarnya. Algoritma *Stemming* untuk satu bahasa satu dan bahasa lainnya berbeda, sebagai contoh Bahasa Inggris memiliki morfologi yang berbeda dengan Bahasa Indonesia. Proses *stemming* untuk Bahasa Indonesia lebih kompleks karena terdapat lebih banyak variasi imbuhan yang harus dibuang untuk mendapatkan *rootword*[1]. Penggunaan algoritma Stemming dalam Bahasa Indonesia sebelumnya pernah diteliti oleh Nazief dan Adriani yang berdasarkan dari Algoritma *Stemming* Porter.

Similarity atau tingkat kesamaan dokumen lama dengan dokumen baru dihitung untuk mengetahui tingkat kesamaan topik melalui judul dan abstraksi dokumen yang diambil berdasarkan indexnya. *Similarity* digunakan untuk mencegah terjadinya duplikasi dan plagiatisme. Pada papper ini hanya akan dibahas mengenai konsep algoritma *stemming* dan *similiraity* pada penerimaan judul *thesis* dengan Bahasa Indonesia.

Thesis adalah istilah yang digunakan di Indonesia untuk mengilustrasikan suatu karya tulis ilmiah berupa paparan tulisan hasil penelitian sarjana S2 yang membahas suatu permasalahan/fenomena dalam bidang ilmu tertentu dengan menggunakan kaidah-kaidah yang berlaku.

Landasan Teori

Stemming

Stemming merupakan suatu proses yang terdapat dalam sistem IR yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (r*ootword*) dengan menggunakan aturan-aturan tertentu. Sebagai contoh, kata bersama, kebersamaan, menyamai, akan distem ke *root wordnya* yaitu "sama". Proses *stemming* pada teks Bahasa Indonesia berbeda dengan *stemming* pada teks berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia, selain sufiks, prefiks, dan konfiks juga dihilang-kan[2].

Similarity

Konsep *similarity* sudah menjadi isu yang sangat penting hampir setiap bidang ilmu pengetahuan.[3] Terdapat tiga macam teknik yang dibangun untuk menentukan nilai *similarity*:

1. Distance-based similarity measure

Distance-based similarity measure mengukur tingkat kesamaan dua buah objek dari segi jarak geometris dari variabel-variabel yang tercakup di dalam kedua objek tersebut. Metode ini meliputi : *Menkowski Distance, Manhatann/ City Block Distance, Eulidean Distance, Jaccard Distance, Dicee's Coefficient, Cosine similarity, Levenshtein Distance, Hamming distance, dan Soundex distance.*

2. Feature-based similarity measure

Feature-based similarity measure melakukan perhitungan tingkat kemiripan dengan merepresentasikan objek ke dalam bentuk *feature-feature* yang ingin dibandingkan. *Feature-based* ini banyak digunakan pada pengklasifikasian atau *patern matching* untuk gambar dan teks.

3. Probabilistic-based similarity measure

Probabilistic-based similarity measure mengukur tingkat kemiripan dua objek dengan me-representasikan dua set objek yang dibandingkan dalam bentuk *probability*. Metode ini mencakup *Kullback Leibler Distance dan Posterior Probability*.

Perhitungan *Similarity* pada jarak antara dua entitas informasi adalah syarat inti pada semua kasus penemuan informasi, seperti pada *Information Retrieval* yang kemudian digunakan untuk pendeteksi *plagiarisme*.

Mengukur kesamaan semantik (*Semantic Similarity)* merupakan tugas yang sulit dalam membandingkan kesamaan kata-kata dalam dokumen. Kesamaan kata-kata dalam dokumen diukur berdasarkan indeks kesamaan jumlah kata yang mungkin. Berdasarkan penelitian Alsamadi, untuk menentukan jumlah kata dapat diukur dengan algoritma :

$$T = \sum_{i=0}^{n} w, p \qquad \text{……..……..(1)}$$

Dimana w adalah kata-kata dan p adalah posisi yang membedakan kata-kata secara individual. Namun kesamaan 100% dalam dokumen jarang berlaku dalam lingkup plagiarisme. *Stop words* atau kata umum juga harus dihilangkan dan diabaikan karena kata umum akan membuat tidak relevan dalam menghitung kesamaan dokumen hal itu dapat mendistorsi perhitungan kesamaan dokumen[4].

Penelitian Terdahulu

Algoritma *stemming* untuk beberapa bahasa telah dikembangkan, seperti Algoritma *Porter* untuk teks berbahasa inggris, Algoritma *Porter* untuk teks berbahasa Indonesia, Algoritma Nazief & Adriani untuk teks berbahasa Indonesia [5].

Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah *root word*. Maka algoritma berhenti.
2. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika
   tidak maka ke langkah 3a.
   a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
   a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
   b. For i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan *Recoding*.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

1. Jika awalannya adalah: “di-”, “ke-”, atau “se-” maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
2. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-” maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
3. Jika dua karakter pertama bukan “di-”,“ke-”,“se-”,“te-”, “be-”, “me-”, atau “pe-” maka berhenti.
4. Jika tipe awalan adalah “none” maka berhenti. Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada.

Tabel 1. Kombinasi Awalan Akhiran Yang Tidak Diijinkan

Awalan	Akhiran yang tidak diijinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

Tabel 2. Cara Menentukan Tipe Awalan Untuk Kata Yang Diawali Dengan “te-”

Following Characters				Tipe Awalan
Set 1	Set 2	Set 3	Set 4	
“r-“	“-r-“	-	-	none
“r-“	Vowel	-	-	ter-luluh
“r-“	not (vowel or “-r-”)	“-er-“	vowel	ter
“r-“	not (vowel or “-r-”)	“-er-“	not vowel	ter-
“r-“	not (vowel or “-r-”)	not “-er-“	-	ter
not (vowel or “-r-”)	“-er-“	vowel	-	none
not (vowel or “-r-”)	“-er-“	not vowel	-	te

Tabel 3. Jenis Awalan Berdasarkan Tipe Awalannya

Tipe Awalan	Awalan yang harus dihapus
di-	di-
ke-	ke-
se-	se-
te-	te-
ter-	ter-
ter-luluh	ter

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan dibawah ini:

1. Aturan untuk reduplikasi.

Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word*

adalah bentuk tunggalnya, contoh : "buku-buku" *root word*-nya adalah "buku".
Kata lain, misalnya "bolak-balik", "berbalas-balasan, dan "seolah-olah". Untuk mendapatkan *root word-nya*, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama maka diubah menjadi bentuk tunggal, contoh: kata "berbalas-balasan", "berbalas" dan "balasan" memiliki *root word* yang sama yaitu "balas", maka *root word* "berbalas-balasan" adalah "balas". Sebaliknya, pada kata "bolak-balik", "bolak" dan "balik" memiliki *root word* yang berbeda, maka *root word*-nya adalah "bolak-balik".
2. Tambahan bentuk awalan dan akhiran serta aturannya.
Untuk tipe awalan "mem-", kata yang diawali dengan awalan "memp-" memiliki tipe awalan "mem-". Tipe awalan "meng-", kata yang diawali dengan awalan "mengk-" memiliki tipe awalan "meng-". Algoritma kedua yang digunakan dalam sistem ini adalah Algoritma Porter. Adapun langkah-langkah algoritma ini adalah sebagai berikut:

1. Hapus *Particle*.
2. Hapus *Possesive Pronoun*.
3. Hapus awalan pertama. Jika tidak ada lanjutkan ke langkah 4a, jika ada cari maka lanjutkan ke langkah 4b.
4. a. Hapus awalan kedua, lanjutkan ke langkah 5a.
   b. Hapus akhiran, jika tidak ditemukan maka kata tersebut diasumsikan sebagai *root word*. Jika ditemukan maka lanjutkan ke langkah 5b.
5. a.Hapus akhiran. Kemudian kata akhir diasumsikan sebagai *root word.*
   b.Hapus awalan kedua. Kemudian kata akhir diasumsikan sebagai *root word.*

Tabel 4. Aturan Untuk *Inflectional Particle*

Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-kah	NULL	2	NULL	bukukah
-lah	NULL	2	NULL	pergilah
-pun	NULL	2	NULL	bukupun

Tabel 5. Aturan Untuk *Inflectional Possesive Pronoun*

Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-ku	NULL	2	NULL	bukuku
-mu	NULL	2	NULL	bukumu
-nya	NULL	2	NULL	bukunya

Tabel 6. Aturan Untuk *First Order Derivational Prefix*

Awalan	Replacement	Measure Condition	Additional Condition	Contoh
meng-	NULL	2	NULL	mengukur → ukur
meny-	S	2	V…*	menyapu → sapu
men-	NULL	2	NULL	menduga → duga
mem-	P	2	V…	memaksa→ paksa
mem-	NULL	2	NULL	membaca→ baca
me-	NULL	2	NULL	merusak → rusak
peng-	NULL	2	NULL	pengukur→ ukur
peny-	S	2	V…	penyapu → sapu
pen-	NULL	2	NULL	penduga → duga
pem-	P	2	V…	pemaksa → paksa
pem-	NULL	2	NULL	pembaca → baca
di-	NULL	2	NULL	diukur → ukur
ter-	NULL	2	NULL	tersapu → sapu
ke-	NULL	2	NULL	kekasih → kasih

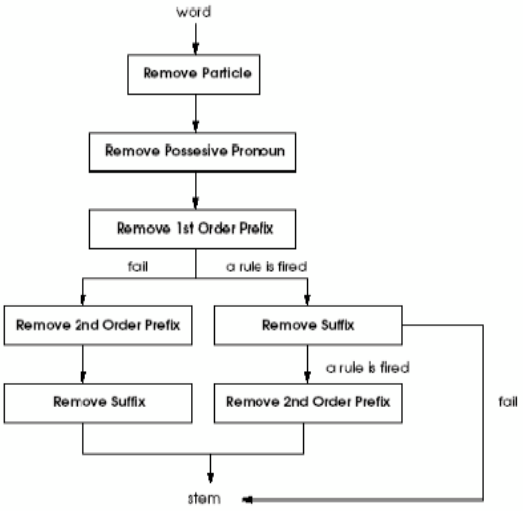
Tabel 7. Aturan Untuk *Second Order Derivational Prefix*

Awalan	Replacement	Measure Condition	Additional Condition	Contoh
ber-	NULL	2	NULL	*berlari → lari*
bel-	NULL	2	*Ajar*	*belajar → ajar*
be-	NULL	2	*k*er*	*bekerja → kerja*
per-	NULL	2	NULL	*perjelas → jelas*
pel-	NULL	2	*Ajar*	*pelajar → ajar*
pe-	NULL	2	NULL	*pekerja → kerja*

Tabel 8. Aturan Untuk *Derivational Suffix*

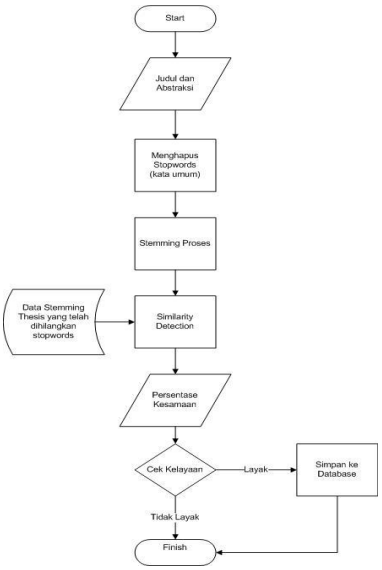
Akhiran	Replacement	Measure Condition	Additional Condition	Contoh
-kan	NULL	2	Prefix bukan anggota {ke, peng}	tarikkan → tarik, mengambilkan → ambil
-an	NULL	2	prefix bukan anggota {di, meng, ter}	makanan → makan, perjanjian → janji
-i	NULL	2	prefix bukan anggota{ber, ke, peng}	Tandai → tanda, mendapati → dapat

Proses stemming menggunakan Algoritma *Porter* dapat dilihat pada Gambar 1.



Gambar 1. Algoritma *Porter*

Analisa dan Perancangan
Dalam pengajuan judul Thesis pada suatu instansi perguruan tinggi hal yang paling mendasari adalah judul yang diangkat dan abstraksi yang menggambarkan judul tersebut. Penulisan abstraksi rata-rata terdiri dari 200 sampai 250 kata. Untuk menghindari kesamaan tema yang diangkat diperlukan sebuah *Information Retrieval* yang menghitung jumlah kesamaan kata dasar dari abstraksi tersebut. Alur *flowchart* sistemnya adalah seperti berikut.



Gambar 2. *Flowchart* Sistem Pendeteksi Kesamaan Judul dan Abstraksi *Thesis.*

Dua kriteria utama untuk mengukur kinerja kesamaan dokumen adalah akurasi dan kinerja atau kecepatan. Banyak *tools* untuk menghitung kesamaan dokumen saat ini relatif lambat dan membutuhkan banyak sumber daya. Dalam hal ini, kecepatan atau kinerja tergantung pada jumlah database abstraksi yang sudah di*stemming* dan dihilangkan *stopwords*-nya.

Pada proses *similarity detection* yang dibandingkan adalah abstraksi yang telah dilakukan *stemming* dan dihilangkan kata umumnya (*stopwords)*, pengecekan ini akan melibatkan seluruh database abstraksi lama yang telah tersimpan sehingga akan membutuhkan waktu cukup lama untuk mengkalkulasi persentase tingkat kesamaan terhadap masing-masing abstraksi.

Apabila persentase memenuhi tingkat kelayakan maka abstraksi baru yang telah d*i stemming* dan dihilangkan *stopwords* akan disimpin dan digunakan untuk pengecekan abstraksi berikutnya sehingga tema yang sudah diangkat dalam *thesis* tidak dapat diangkat lagi kecuali dengan pengembangan tertentu.

Berikut adalah contoh implementasi sistem:

1. Contoh kalimat asli :

similarity atau tingkat kesamaan dokumen lama dengan dokumen baru dihitung untuk mengetahui tingkat kesamaan topic melalui judul dan abstraksi dokumen yang diambil berdasarkan indexnya similarity digunakan untuk mencegah terjadinya duplikasi dan plagiarisme pada papper ini hanya akan dibahas mengenai konsep algoritma stemming dan similiraity pada penerimaan judul thesis dengan bahasa indonesia

Gambar 3. Contoh kalimat asli

2. Contoh kalimat setelah dihilangkan *stopwords* :

similarity kesamaan dokumen dokumen dihitung kesamaan topic judul abstraksi dokumen diambil indexnya similarity mencegah terjadinya duplikasi plagiarisme papper dibahas konsep algoritma stemming similiraity penerimaan judul thesis bahasa indonesia

Gambar 4. Kalimat setelah dihilangkan *Stopwords*

3. Contoh proses Stemming kalimat :

Stemming :
1) similarity=>similarity
2) kesamaan=>sama
3) dokumen=>dokumen
4) dokumen=>dokumen
5) dihitung=>hitung
6) kesamaan=>sama
7) topic=>topic
8) judul=>judul
9) abstraksi=>abstraksi
10) dokumen=>dokumen
11) diambil=>ambil
12) indexnya=>index
13) similarity=>similarity
14) mencegah=>cegah
15) terjadinya=>jadi
16) duplikasi=>duplikasi
17) plagiarisme=>plagiarisme

Gambar 5. Contoh *Stemming* kalimat dengan Algoritma Nazief & Adriani

18) papper=>papper
19) dibahas=>bahas
20) konsep=>konsep
21) algoritma=>algoritma
22) stemming=>stemming
23) similiraity=>similiraity
24) penerimaan=>terima
25) judul=>judul
26) thesis=>thesis
27) bahasa=>bahasa
28) indonesia=>indonesia

Gambar 6. Contoh *Stemming* kalimat dengan Algoritma Nazief & Adriani (lanjutan)

4. Contoh kalimat akhir yang telah di hilangkan *stopwords* dan dilakukan stemming.

kalimat yang udah di stemming dan remove stopwords :
similarity sama dokumen dokumen hitung sama topic judul abstraksi dokumen ambil index similarity cegah jadi duplikasi plagiarisme papper bahas konsep algoritma stemming similiraity terima judul thesis bahasa indonesia

Gambar 6. Contoh kalimat akhir yang akan dihitung *similarity*-nya.

5. Contoh hasil akhir perhitungan similarity

Tabel 9. Hasil akhir perhitungan

Dokumen Ke-	jumlah kata Dokumen Lama	jumlah kata sama	jumlah kata berbeda	persentase
1	100	2	56	3.448275862069
2	40	4	54	6.8965517241379
3	85	10	48	17.241379310345
4	78	4	54	6.8965517241379
5	91	9	49	15.51724137931
6	86	11	47	18.965517241379
7	89	3	55	5.1724137931034
8	101	1	57	1.7241379310345
9	81	4	54	6.8965517241379
10	85	2	56	3.448275862069

Berdasarkan hasil akhir tersebut maka abstraksi baru akan dibandingkan dengan semua abstraksi yang ada di dalam *database*, abstraksi yang berada didalam database sudah dilakukan *Stemming* dan penghapusan *stopwords* sehingga terlihat pada tabel 9, jumlah kata yang dimaksud adalah jumlah kata yang sudah dilakukan *stemming* dan penghapusan *stopwords* untuk selanjutnya dilakukan perhitungan *Semantic Similarity* abstraksi baru terhadap tiap-tiap abstraksi lama.

Penulis melakukan uji coba dengan pengambilan 10 abstraksi secara acak milik *bimbingan.amikom.ac.id* sehingga tingkat persentase kesamaannya juga beragam. Untuk menentukan seberapa tinggi tingkat persentase yang diperbolehkan merupakan kebijakan suatu instansi yang terikat akan tetapi pada kaidahnya kurang dari 50% dapat diterima.

Hal yang perlu diperhatikan dalam proses *stemming* dan penghilangan *stopwords* adalah kualitas kamus yang digunakan, semakin lengkap kamus kata dasar untuk stemming dan kamus kata umum untuk proses penghapusan *stopwords.*

Hasil dari algoritma ini dibandingkan dengan manual verifikasi menunjukkan bahwa kesamaan judul dan

abstraksi pada *thesis* menujukkan bahwa penggunaan algoritma ini lebih cepat dibandingkan dengan melakukan pengecekan manual satu persatu.

Penutup

A. Kesimpulan

Berdasarkan penelitian terdahulu dan analisis penulis, maka algoritma Adriani & Nazief dan Algoritma *Similarity* dapat digunakan untuk pengecekan judul dan abstraksi *thesis*, apakah judul dengan tema tersebut sudah pernah diajukan atau belum. *Stemming* berfungsi untuk mengumpulkan index judul dan abstraksi *thesis* sebagai *database* sehingga dapat dilakukan pengecekan dengan menggunakan algoritma *similarity*.

B. Saran

Pada bagian ini penulis ingin menyarankan bahwa penerapan algoritma untuk mendeteksi plagiarism ataupun kesamaan pada judul *thesis* dan abstraksi tidak hanya sebatas menggunakan algoritma *Stemming* dan *Similarity*, masih terdapat banyak algoritma lain yang menunjang untuk menuju kearah tersebut.

Daftar Pustaka

[1] Agusta, Ledy, Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana

[2] Fadillah Z. Tala, A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia, Netherland, Universiteit van Amsterdam

[3] B. Zaka, "Theory and Applications of Similarity Detection Techniques, : Fraz University of Technology. 2009 B. Zaka, "Theory and Applications of Similarity Detection Techniques, : Fraz University of Technology. 2009

[4] Alsamadi, Izzat. Saleh, Zakaria Issa, Documents Similarities Algorithms for Research Papers Authenticity. IT Faculty, Yarmouk University. Jordan.

[5] Nazief, Bobby dan Mirna Adriani, Confix-Stripping: Approach to Stemming Algorithm for Bahasa Indonesia, Fakulty of Computer Science University of Indonesia.