

# Penanganan Kasus *Overstemming* dan *Understemming* dengan Modifikasi Algoritma *Stemming*

Stephanie Betha R.H

Informatika , AMIK Purnama Niaga Indramayu  
 Jl Ir.H.Djuanda No.256 Indramayu  
 ntephbetha@gmail.com

## Abstrak

Praproses merupakan suatu tahapan yang sangat penting dalam *data mining*. Kualitas data yang dihasilkan dari praproses dapat mempengaruhi hasil pengolahan *data mining*. Praproses teks terdiri dari beberapa tahapan, antara lain : *case folding*, tokenisasi, penghapusan *stopwords* dan *stemming*. *Stemming* berfungsi untuk mengembalikan kata ke bentuk dasarnya, meningkatkan *recall* dan memberikan hasil yang relevan pada proses pencarian kata. Oleh karena itu, proses *stemming* memegang peranan penting pada praproses teks. Salah satu masalah yang terjadi pada proses *stemming* yaitu *overstemming* dan *understemming*. *Overstemming* adalah pemotongan kata yang berlebihan menyebabkan situasi dimana suatu kata yang memiliki makna yang sangat berbeda, namun menghasilkan *stem* yang sama. Sedangkan, *understemming* adalah pemenggalan imbuhan yang terlalu sedikit dari seharusnya, hal ini menyebabkan kata-kata yang sama tidak menghasilkan *stem* yang sama. Oleh karena itu, untuk mengatasi permasalahan tersebut, maka akan dilakukan modifikasi pada proses *stemming*. Modifikasi ini dilakukan dengan menggabungkan dua algoritma *stemming* (*hybrid stemming*) yaitu algoritma *look-up table* kamus kata dan algoritma *affix removal* menggunakan *stemming* Porter. Modifikasi algoritma *stemming* ini akan diuji cobakan pada atribut judul dan kata kunci pada dokumen publikasi ilmiah. Hasil pengujian menunjukkan bahwa proses *stemming* dengan modifikasi algoritma *stemming* dapat meningkatkan nilai *recall* pada atribut judul, walaupun tidak terlalu signifikan. Atribut judul menghasilkan *recall* sebesar 89,9%. Sedangkan, atribut kata kunci menghasilkan *recall* sebesar 98,1%.

**Kata kunci :** praproses, *stemming*, modifikasi *stemming*

## Abstract

*Preprocess is a very important step in data mining. The quality of data generated from the preprocess can affect the results of data mining processing. Preprocess text consists of several stages, among others: case folding, tokenisasi, removal of stopwords and stemming. Stemming serves to return the word to its basic form, increasing the recall and providing relevant results in the word search process. Therefore, the stemming process plays an important role in the preprocessing of the text. One of the problems that occur in the stemming process is overstemming and understemming. Overstemming is an exaggerated word cut causing situations where a word has a very different meaning, but it produces the same stem. Whereas, understemming is a beheading of too little supposition, it causes the same words not to produce the same stem. Therefore, to overcome these problems, it will be modified on the stemming process. This modification is done by combining two stemming algorithms (hybrid stemming) that is the look-up algorithm of dictionary table and affix removal algorithm using stemming porter. The modification of this stemming algorithm will be tested on title and keyword attributes in scientific publication documents. The test results show that stemming process with modification of stemming algorithm can increase the recall value in the title attribute, although not very significant. The recall in an experiment using title attribute is 89,9%. Whereas, for keywords attribute, the recall is 98,1%.*

**Keywords:** *preprocess, stemming, stemming modification*

## I. PENDAHULUAN

Tamilselvi dan Brilly (2011) menyatakan bahwa praproses merupakan suatu proses yang sangat penting dalam tahapan *data mining*. Kualitas data yang dihasilkan dari praproses dapat mempengaruhi hasil pengolahan *data mining*. Penelitian Yates dan Neto (1999) menyatakan bahwa praproses teks terdiri dari beberapa tahapan, antara lain : *case folding*, tokenisasi, penghapusan *stopwords* dan *stemming*. Joshi dkk. (2016) menyatakan proses *stemming* berfungsi untuk

mengembalikan kata ke bentuk dasarnya, meningkatkan *recall* dan memberikan hasil yang relevan pada proses pencarian kata. Oleh karena itu, proses *stemming* memegang peranan penting pada praproses teks.

Salah satu masalah yang terjadi pada proses *stemming* yaitu *overstemming* dan *understemming*. Karaa, W. (2013) menyatakan bahwa *overstemming* adalah pemotongan kata yang berlebihan menyebabkan situasi dimana suatu kata yang memiliki makna yang sangat berbeda, namun menghasilkan *stem* yang sama.

Misalnya, kata “*generalized, general, generous, generating*” menghasilkan stem “*gener*”. Selain itu, pemenggalan imbuhan yang terlalu sedikit dari seharusnya, hal ini menyebabkan kata-kata yang sama tidak menghasilkan *stem* yang sama (*understemming*). Misalnya, kata “*dryman*” dan “*drymen*” memiliki arti yang sama. Namun, kedua kata tersebut menghasilkan *stem* yang berbeda yaitu “*dryman*” dan “*drymen*”. Penelitian sebelumnya (Karaa, W, 2013) telah menangani kasus *overstemming* dan *understemming* dengan memodifikasi algoritma *stemming* Porter. Modifikasi algoritma *stemming* Porter dilakukan dengan membuat beberapa tabel aturan kata.

Penelitian ini juga bertujuan untuk mengatasi permasalahan kasus *overstemming* dan *understemming*. Kasus tersebut akan ditangani dengan memodifikasi pada proses *stemming*. Modifikasi ini dilakukan dengan menggabungkan dua algoritma *stemming* (*hybrid stemming*) yaitu algoritma *look-up table* kamus kata dan algoritma *affix removal* menggunakan *stemming* Porter. Algoritma *look-up table* kamus kata ini akan mengadopsi tabel aturan yang telah diusulkan oleh penelitian sebelumnya (Karaa, W, 2013).

Penelitian ini menggunakan atribut judul dan kata kunci pada dokumen publikasi. Atribut judul dan kata kunci ini akan mengalami proses *stemming* dengan modifikasi dan proses *stemming* tanpa modifikasi. Kedua proses tersebut akan menghasilkan perbandingan nilai *recall* pada judul dan kata kunci.

## II. METODE PENELITIAN

Proses *stemming* menjadi salah satu bagian pada praproses teks. Proses *stemming* dilakukan setelah proses *case folding*, tokenisasi dan penghapusan *stop words*. Beberapa tahapan yang harus dilalui dalam praproses teks, antara lain :

### 1. Case Folding

Proses *case folding* dilakukan dengan mengubah huruf besar menjadi huruf kecil. Hanya huruf ‘a’ sampai ‘z’ yang diterima. Karakter selain huruf dianggap sebagai delimiter sehingga karakter tersebut akan dihapus dari dokumen, misalnya, tanda baca (&#, : , - , ; ) dan angka (0-9).

### 2. Tokenisasi

Proses tokenisasi adalah proses pemecahan kalimat menjadi kata (token). Adanya proses ini string yang diinputkan akan terlihat lebih ringkas karena ditampilkan dalam bentuk tiap kata.

### 3. Penghapusan Stop Words

Penghapusan *stop words* dilakukan setelah melalui proses tokenisasi. Proses ini bertujuan untuk menghapus kata-kata yang sering muncul dan tidak dipakai dalam pemrosesan. Penghapusan *stop words* umumnya mengikuti standar daftar *stop words* yang sudah ada, khususnya dalam Bahasa Inggris. C, Fox (1992) telah membuat standar *stop words* pada *general text*, antara lain : *a, about, a, able,*

*about, across, after, all, almost, also* dan sebagainya.

### 4. Stemming

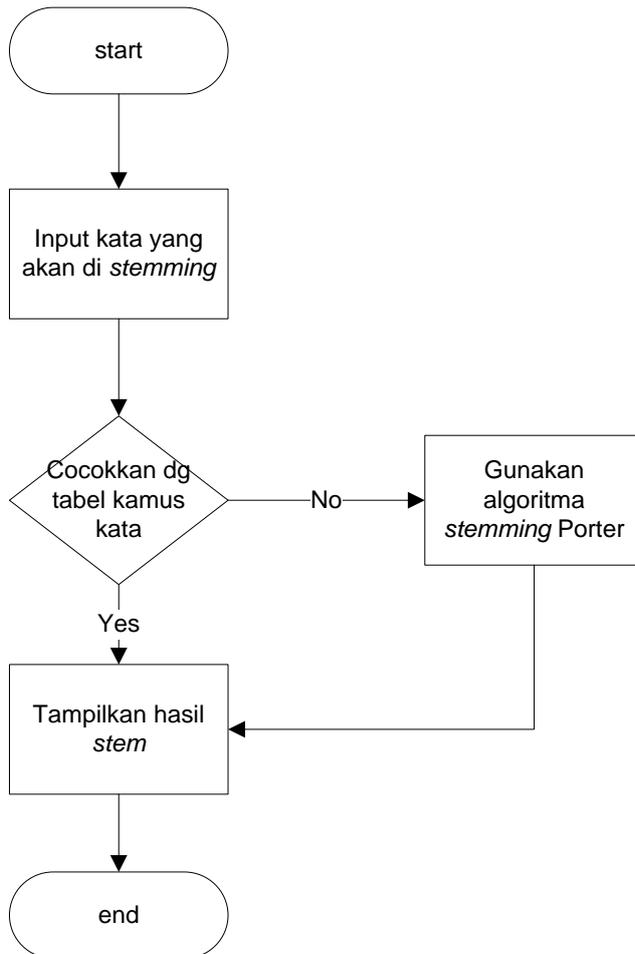
Joshi dkk. (2016) menyatakan proses ini berfungsi untuk mengembalikan kata ke bentuk dasarnya, meningkatkan *recall* dan memberikan hasil yang relevan pada proses pencarian kata. Manning dkk. (2009) menyatakan bahwa *recall* adalah rasio antara dokumen relevan yang berhasil ditemukan-kembalikan dari seluruh dokumen relevan yang ada di dalam sistem. Manning dkk. (2009) juga menyatakan bahwa *stemming* bertujuan untuk mengurangi bentuk infleksi dan bentuk turunan dari suatu kata ke bentuk dasar umum. Misalnya, kata “*cars, car’s, car*” memiliki bentuk dasar yang sama yaitu “*car*”. Permasalahan utama pada algoritma *stemming* adalah bagaimana mendapatkan kata dasar yang benar dari suatu kata jadian (Mandala dkk, 2004). Salah satu masalah yang terjadi pada proses *stemming* yaitu *overstemming* dan *understemming*.

Penanganan kasus *overstemming* dan *understemming* dilakukan dengan menggabungkan dua algoritma *stemming* (*hybrid stemming*) yaitu algoritma *look-up table* kamus kata dan algoritma *affix removal* menggunakan *stemming* Porter. Penelitian Dogra dkk. (2013) menyatakan bahwa *look up table* pada dasarnya adalah menyimpan *root words* pada tabel di database beserta bentuk turunan kata tersebut. Kata-kata dimasukkan ke dalam tabel secara manual. Daftar kata yang telah masuk ke dalam *database* dapat mengurangi kesalahan dalam *overstemming* dan *understemming*. Ketika pengguna memasukkan kata infleksi, maka *stemmer* akan mencari keberadaan kata infleksi tersebut dalam *database*. Infleksi merupakan perubahan bentuk kata yang tidak mengubah arti kata tersebut (Karaa, W, 2013).

*Look up table* kamus kata pada penelitian ini dilakukan dengan membuat daftar kata yang masuk ke dalam tabel aturan. Tabel aturan (Karaa, W, 2013) merupakan tabel-tabel yang memuat aturan pemotongan kata pada proses *stemming*. Daftar kata ini dibuat berdasarkan kamus kata yang diperoleh dari website *morewords*. *Website* ini digunakan sebagai pilihan kamus kata karena memiliki kata yang bersumber dari *Enhanced North American Benchmark Lexicon* (ENABLE2K) yang memiliki kata lebih dari 173.528. Daftar kata yang digunakan beserta kata hasil turunannya dimasukkan ke dalam tabel dalam *database* secara manual.

Gambar 1 menjelaskan tentang alur penanganan kasus *overstemming* dan *understemming* yang tidak tertangani oleh algoritma *stemming* Porter. Apabila kata termasuk ke dalam kasus *overstemming* dan *understemming* yang ada pada tabel aturan, maka kata tersebut akan menghasilkan *stem* sesuai dengan tabel kamus kata. Namun, apabila kata tidak termasuk ke dalam daftar tabel aturan, maka kata tersebut akan mendapat perlakuan *stemming* menggunakan algoritma Porter yang telah ada.

Rangkaian proses berikut ini merupakan langkah dalam menangani kasus *stemming* pada penelitian ini :



Gambar 1. Alur penanganan kasus *overstemming* dan *understemming*

Kasus *overstemming* dan *understemming* akan ditangani menggunakan beberapa tabel aturan yang diusulkan oleh Karaa, W. (2013) sebagai berikut :

1. Aturan Pertama

Aturan pertama menangani kasus bentuk jamak pada kata tidak beraturan yang tidak ditangani pada algoritma *stemming* Porter. Tabel 1 menunjukkan daftar kategori kasus yang ditangani pada aturan pertama :

2. Aturan Kedua

Aturan kedua menangani kata-kata yang memiliki *stem* “*gener*” (Porter, M, 2002) dan kata yang memiliki awalan “*gene*”. Beberapa kata yang ada pada Tabel 2 sebelum mengalami modifikasi algoritma *stemming*, akan mengalami kasus *overstemming*. Semua kata tersebut menghasilkan *stem* yang sama yaitu “*gener*”. Berikut ini adalah daftar kata yang memiliki *stem* “*gener*” dengan perubahan hasil *stem*-nya. Tabel 2 menunjukkan kasus yang ditangani pada aturan kelima.

TABEL 1. DAFTAR KATEGORI KASUS YANG DITANGANI PADA ATURAN PERTAMA

No	Kategori	Original Words	Aturan	Hasil Stem Modifikasi	Total Kata
1	Kata yang berakhiran children adalah bentuk jamak dari child	child/children	Mengubah children menjadi child	child/child	6 kata
2	Kata yang berakhiran -men adalah bentuk jamak dari -man	dryman/drymen	mengubah -men menjadi -man	dryman/dryman	429 kata
3	Kata yang berakhiran -ci adalah bentuk jamak dari -cus	abacus/abaci	mengubah -ci menjadi -cus	abacu/abacu	35 kata
4	Kata yang berakhiran -eaux adalah bentuk jamak dari -eau	plateau/plateaux	mengubah -eaux menjadi -eau	plateau/ plateau	29 kata
5	Kata yang berakhiran -mata adalah bentuk jamak dari -ma	automa/automata	mengubah -mata menjadi -ma	automa/automa	108 kata
6	Kata yang berakhiran -trices adalah bentuk jamak dari -trix	matrix/matrices	mengubah -trices menjadi -trix	matrix/ matrix	21 kata
7	Kata yang berakhiran -ses adalah bentuk jamak dari -sis	analysis/analyses	mengubah -sis menjadi -s	anals/analys	492 kata

TABEL 2. DAFTAR KATEGORI KASUS YANG DITANGANI PADA ATURAN KEDUA

Kata	Hasil stem sebelum modifikasi	Hasil stem modifikasi	Total Kata
<i>generate</i>	<i>Gener</i>	<i>generat</i>	76 kata
<i>generates</i>	<i>Gener</i>		
<i>generated</i>	<i>Gener</i>		
<i>generating</i>	<i>Gener</i>		
<i>general</i>	<i>Gener</i>	<i>general</i>	
<i>generally</i>	<i>Gener</i>		
<i>generic</i>	<i>Gener</i>	<i>generic</i>	
<i>generically</i>	<i>Gener</i>		
<i>generous</i>	<i>Gener</i>	<i>generous</i>	
<i>generously</i>	<i>Gener</i>		

3. Aturan Ketiga

Aturan ketiga menangani kata-kata yang memiliki akhiran -y dan tidak mengandung vokal pada kata tersebut yang tidak tertangani oleh algoritma *stemming* Porter. Tabel 3 menunjukkan kasus yang ditangani pada aturan kelima.

TABEL 3. DAFTAR KATEGORI KASUS YANG DITANGANI PADA ATURAN KETIGA

Original Words	Hasil stem sebelum modifikasi	Hasil Stem Modifikasi	Total Kata
<i>cry/cries/cried/crying</i>	<i>cry/cri/cri/cry</i>	<i>cry/cry/cri/cry</i>	20 kata
<i>dry/dyes/dyed/dying</i>	<i>dye/dyes/dyed/dying</i>	<i>dy/dy/dy/dy</i>	

TABEL 4. DAFTAR KATEGORI KASUS YANG DITANGANI PADA ATURAN KETIGA

No	Original Words	Hasil stem sebelum modifikasi	Aturan	Hasil Stem Modifikasi	Total Kata
1	<i>focus/focuses/focused/focusing</i>	<i>focus/focus/focus</i>	mengubah -sed menjadi -s mengubah -sing menjadi -s	<i>focu/focu/focu</i>	28 kata

4. Aturan Keempat

Aturan keempat menangani kata-kata yang memiliki akhiran -s (bukan -ss) dan bentuk *participle* dari kata tersebut. Algoritma *stemming* Porter akan menghapus akhiran -s pada kata yang memiliki akhiran -s. Misalnya, kata “*focus*” menjadi “*focu*”. Namun, apabila kata berakhiran -s tersebut berada dalam bentuk *past* atau *present participle* maka *stemming* Porter hanya akan menghapus akhiran -ed atau -ing. Misalnya, kata “*focused*”, “*focusing*” menjadi “*focus*” dan “*focus*”. Tabel 4 menunjukkan kasus yang ditangani pada aturan kelima.

5. Aturan Kelima

Algoritma *stemming* Porter tidak menangani *present* atau *past participle derivations*. Misalnya, kata “*studiedly*” menghasilkan *stem* “*studiedli*” seharusnya “*study*”. Aturan kelima akan menangani kata-kata pada kategori tersebut. Tabel 5 menunjukkan kasus yang ditangani pada aturan kelima.

TABEL 5. DAFTAR KATEGORI KASUS YANG DITANGANI PADA ATURAN KETIGA

No	Kategori	Original Words	Hasil stem sebelum modifikasi	Aturan	Hasil Stem Modifikasi	Total Kata
1	Kata yang berakhiran -iedly atau -iedness berelasi dengan kata yang berakhiran -ied	<i>study/studied/studiedness/studiedly</i>	<i>studi/studi/studied/studiedli</i>	mengubah -ly menjadi -ied mengubah -ss menjadi -ied	<i>study/study/study</i>	13 kata
2	Kata yang berakhiran -edly atau -edness berelasi dengan kata yang berakhiran -ed	<i>amaz/amazed/amazedly/amazedness</i>	<i>amaz/amaz/amazedli/amazed</i>	mengubah -ly menjadi -ed mengubah -ss menjadi -ed	<i>amaz/amaz/amaz</i>	439 kata

III. HASIL DAN PEMBAHASAN

Penanganan kasus *overstemming* dan *understemming* dilakukan berdasarkan kamus kata yang terdapat dalam *database*. Kamus kata ini disusun menggunakan aturan yang diadopsi dari penelitian Karaa, W. (2013) dan daftar kata dari [www.morewords.com](http://www.morewords.com). Penanganan kasus ini diimplementasikan pada atribut judul dan kata kunci dokumen publikasi dari data DBLP serta dokumen publikasi Kelompok Keilmuan RPLP dan Informatika di Sekolah Teknik Elektro dan Informatika (STEI-ITB). Jumlah dokumen publikasi yang digunakan sebanyak 3.012 publikasi ilmiah. Dokumen publikasi ini sudah diklasifikasikan ke dalam 7 kategori bidang penelitian yaitu *AI and Image Processing (AI)*, *Computational Theory and Mathematics (CT)*, *Computer Software (CS)*, *Data Format (DF)*, *Distributed Computing (DC)*, *Information System (ISM)*, *Library and Information Science (LIS)*.

Tabel 6 dan 7 menunjukkan jumlah kata yang mengalami *overstemming* dan *understemming* pada atribut judul dan kata kunci pada dokumen publikasi.

TABEL 6. JUMLAH KASUS OVERSTEMMING YANG DITANGANI

Atribut	Jml Kata <i>Overstemming</i>	Total Kata	Presentase
Kata kunci	29	15.341	0,19%
Judul	1430	17.743	8,06%

Tabel 6 menunjukkan bahwa kata judul memiliki jumlah kasus *overstemming* yang lebih banyak dari kata kunci. Hal ini disebabkan oleh kata kunci lebih banyak mengandung frase (gabungan kata) sehingga banyak yang tidak mengalami perubahan.

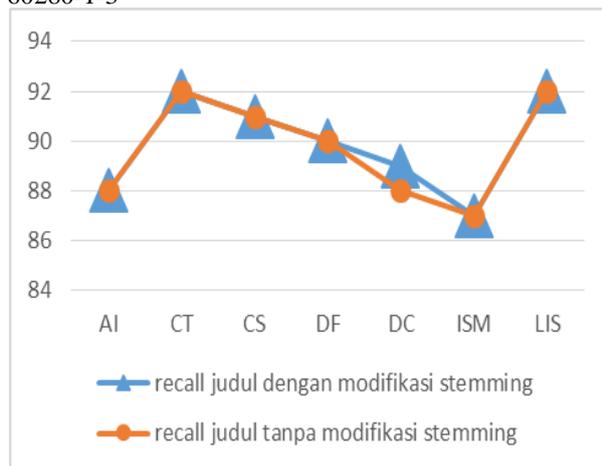
TABEL 7. JUMLAH KASUS UNDERSTEMMING YANG DITANGANI

Atribut	Jml Kata <i>Understemming</i>	Total Kata	Presentase
Kata kunci	162	15.341	1,06%
Judul	1397	17.743	7,87%

Tabel 7 menunjukkan bahwa kata kunci memiliki total presentase kasus *understemming* yang lebih rendah dibandingkan kata judul. Hal ini disebabkan oleh total kata judul yang lebih banyak dari kata kunci, sehingga kemungkinan kata judul yang mengalami *stemming* dengan modifikasi pun lebih banyak.

Kedua tabel di atas menunjukkan bahwa kemungkinan jenis aturan kata masih kurang mampu untuk menangani kasus *overstemming* dan *understemming* pada atribut judul dan kata kunci. Jenis aturan kata yang lebih bervariasi memungkinkan kedua kasus tersebut lebih banyak yang tertangani.

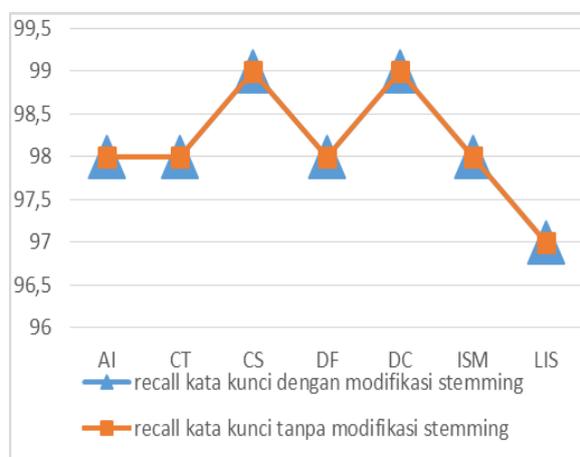
Gambar 2 menunjukkan hasil *recall* penggunaan atribut judul pada *stemming* dengan modifikasi dan *stemming* tanpa modifikasi.



Gambar 2. Grafik perbandingan nilai recall pada judul

Gambar 2 menunjukkan bahwa recall yang dihasilkan oleh stemming dengan modifikasi memiliki nilai yang hampir sama besarnya dengan stemming tanpa modifikasi. Kedua proses tersebut menghasilkan nilai rata-rata recall sebesar 89,9% dan 89,7%. Perbedaan nilai recall kata judul terjadi pada kategori DC. Nilai recall kategori DC pada stemming dengan modifikasi memiliki nilai yang lebih tinggi dibandingkan recall kategori DC pada stemming tanpa modifikasi.

Gambar 3 menunjukkan hasil recall penggunaan atribut kata kunci pada stemming dengan modifikasi dan stemming tanpa modifikasi.



Gambar 3. Grafik perbandingan nilai recall pada kata kunci

Gambar 3 di atas menunjukkan bahwa recall atribut kata kunci yang dihasilkan oleh stemming dengan modifikasi memiliki nilai yang sama besarnya dibandingkan stemming tanpa modifikasi, rata-rata nilai recall sebesar 98,1%.

Stemming dengan modifikasi pada atribut judul tidak terlalu mengakibatkan adanya peningkatan hasil recall dari proses stemming tanpa modifikasi. Hal ini disebabkan oleh atribut judul hanya memiliki sedikit variasi kata yang terkandung pada aturan stemming dengan modifikasi. Selain itu, kata yang mengalami stemming dengan modifikasi kemungkinan termasuk kata yang bersifat umum (memiliki nilai *IDF* kecil) pada judul sehingga tidak terlalu mempengaruhi peningkatan recall. Kata yang bersifat umum artinya kata yang memiliki frekuensi kemunculan di beberapa

kategori, sehingga menghasilkan nilai *IDF* yang kecil. Selain itu, jenis aturan kata kurang banyak variasinya, sehingga kurang dapat menangani kasus *overstemming* dan *understemming* pada data yang digunakan. Hal ini menyebabkan tidak terlalu mempengaruhi peningkatan recall.

Stemming dengan modifikasi pada atribut kata kunci juga tidak mempengaruhi peningkatan recall, karena kata kunci tidak terlalu banyak mengandung kata yang terdapat pada aturan modifikasi stemming. Selain itu, kata kunci lebih banyak mengandung frase dibandingkan term, sehingga banyak kata yang tidak terpengaruh oleh adanya aturan stemming dengan modifikasi.

Atribut kata kunci memiliki nilai recall yang lebih tinggi dibandingkan judul. Hal ini disebabkan oleh term pada kumpulan kata kunci memiliki jenis kata yang jarang (unik) dimiliki pada suatu kategori.

#### IV. KESIMPULAN

Kasus *overstemming* dan *understemming* yang ditangani pada data yang digunakan, dapat meningkatkan hasil recall pada penggunaan atribut judul dan kata kunci, namun hasilnya tidak terlalu signifikan. Hal ini disebabkan oleh banyaknya kata yang tidak tertangani oleh tabel aturan yang diusulkan oleh Karaa, W (2013). Atribut judul hanya memiliki sedikit variasi kata yang terkandung pada aturan stemming dengan modifikasi. Selain itu, kata yang mengalami stemming dengan modifikasi kemungkinan termasuk kata yang bersifat umum (memiliki nilai *IDF* kecil) pada judul sehingga tidak terlalu mempengaruhi peningkatan recall. Sedangkan, atribut kata kunci lebih banyak mengandung frase dibandingkan term, sehingga banyak kata yang tidak terpengaruh oleh adanya aturan stemming dengan modifikasi.

#### DAFTAR PUSTAKA

- C, Fox. (1992) : Lexical Analysis and Stoplist in Information Retrieval, Data Structures and Algorithms, pages 102-130, Prentice Hall.
- Dogra dkk. (2013) : An Effective Stemmer in Devanagari Script. Conference of Recent Trends in Computing and Communication Engineering. RTCCE.
- Frakes, W, B. (1992) : Stemming Algorithms, Data Structures and Algorithms, pages 131-160, Prentice Hall.
- Joshi dkk. (2016) : Modified Stemming Algorithm. IJCSIT International Journal of Computer Science and Information Technologies, Vol. 7 (1) , 266-269.
- Karaa, W. (2013) : A New Stemmer to Improve Information Retrieval. International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.4.
- Manning dkk. (2009) : An Introduction to Information retrieval. Cambridge Online Edition.
- Mandala dkk. (2004) : Sistem Stemming Otomatis untuk Kata dalam Bahasa Indonesia. Seminar Nasional Aplikasi Teknologi Informasi (SNATI).
- Tamilselvi dan Brilly (2011) : Handling Duplicate Data in Data Warehouse for Data Mining. International Journal of Computer Applications (0975 – 8887) Volume 15– No.4.
- Yates dan Neto. (1999). Model Information Retrieval. ACM New York.
- Sumber : Enhanced North American Benchmark Lexicon (ENABLE2K). Retrieved from : <https://www.morewords.com/>