Algoritme Pemotong Akhiran Baku untuk Kata dalam Bahasa Indonesia Berbasis Algoritme Porter

Julio Adisantoso*, Aji Hamim Wigena*, Choirul Hafidz Akhmadi**

*) *Staf Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam*
**) *Mahasiswa Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam*

Abstrak

Dalam suatu sistem temu kembali informasi (Information Retrieval / IR) *berbasis teks, terdapat kumpulan dokumen yang dideskripsikan dengan istilah-istilah. Proses penghilangan akhiran kata akan mengurangi jumlah total istilah dalam sistem IR sehingga mengurangi ukuran dan kompleksitas data dalam sistem, dan seringkali dapat meningkatkan kinerja sistem IR. Dari algoritme pemotong akhiran yang dibuat oleh Porter, dapat ditelaah kemungkinan pengembangan algoritme pemotong akhiran untuk kata dalam Bahasa Indonesia, dengan fokus pada akhiran baku* (-wan, -wati, -man, -nya, -nda, -anda, -wi, -iah, -wiah, -ni, -i, -kan, dan -an).

Dalam penelitian ini, selain memodifikasi aturan pemotongan akhiran agar sesuai dengan Bahasa Indonesia, akan diuji juga modifikasi penghitungan ukuran kata yang disesuaikan dengan penghitungan suku kata dalam Bahasa Indonesia. Penghitungan ukuran kata dilakukan oleh Deterministic Finite Automata. Sistem diimplementasikan pada platform Microsoft Windows menggunakan Microsoft Visual Basic, dengan bahan pengujian berupa kata-kata dari Kamus Besar Bahasa Indonesia dan berbagai artikel.

Dari hasil uji, cara penghitungan ukuran kata yang sesuai dengan cara penghitungan suku kata dalam Bahasa Indonesia dapat meningkatkan keberhasilan dan ketepatan pemotongan dibandingkan dengan menggunakan cara penghitungan ukuran kata dari algoritme Porter

PENDAHULUAN

Latar Belakang

Dalam sebuah sistem temu kembali informasi (*Information Retrieval* / IR) terdapat kumpulan dokumen, yang setiap dokumennya dideskripsikan oleh kata-kata (istilah). Istilah yang memiliki akar kata (*stem*) yang sama umumnya memiliki arti yang sama, misalnya hubung, hubungan, hubungkan, hubungi. Jika keempat istilah ini dikelompokkan ke dalam satu kelompok dengan menghilangkan akhirannya, kinerja sistem IR sering kali meningkat. Proses penghilangan akhiran kata akan mengurangi jumlah total istilah dalam sistem IR sehingga mengurangi ukuran dan kompleksitas data dalam sistem *(Porter, 1980)*.

Selama ini telah banyak dikembangkan cara menghilangkan akhiran kata untuk memperoleh kata dasar/akar kata, salah satunya dengan algoritme pemotongan akhiran Algoritme pemotong akhiran yang banyak dikembangkan adalah *iterative longest match*, seperti algoritme Lovins, Salton, Dawson dan Porter.

Algoritme Porter adalah algoritme yang lebih kecil ukurannya dengan kinerja yang tidak kalah baiknya dengan algoritme yang lebih besar ukurannya *(Frakes, 1992)*. Karena berbasis Bahasa Inggris sehingga belum tentu cocok bila diterapkan dalam sistem temu kembali informasi berbasis Bahasa Indonesia. Penelitian ini mengembangkan algoritme pemotong akhiran baku untuk kata dalam Bahasa Indonesia berdasarkan algoritme Porter.

Tujuan

Penelitian ini bertujuan untuk menelaah, menerapkan dan memodifikasi algoritme Porter untuk memotong akhiran baku untuk kata dalam Bahasa Indonesia.

Ruang Lingkup Penelitian

Penelitian ini dibatasi pada pengembangan algoritme pemotong akhiran baku untuk kata berbasis Bahasa Indonesia berdasarkan algoritme Porter dan pengujian algoritme tersebut pada sejumlah kata berakhiran.

TINJAUAN PUSTAKA

Temu Kembali Informasi

Suatu sistem temu kembali informasi tidak memberi informasi *(atau mengubah pengetahuan)* penggunanya tentang subyek *(fokus)* pencariannya. Sistem temu kembali informasi hanya menginformasikan tentang keberadaan *(atau ketidakberadaan)* dan letak dokumen yang berhubungan dengan pencariannya *(van Rijsbergen, 1979)*. Temu kembali informasi mencakup pengindeksan, penelusuran dan pemanggilan data, terutama data teks atau data dalam bentuk lain yang tidak terstruktur.

Deterministic Finite Automata

Sebuah *Finite Automata (FA)* terdiri dari himpuan terhingga (*finite*) dari state *(keadaan)* dan sekumpulan perpindahan *(transisi)* dari satu *state* ke *state* lain, yang mungkin kembali ke *state* sebelumnya *(Hopcroft dan Ullman, 1979)*.

Deterministic Finite Automata (DFA) adalah subset dari FA dimana untuk setiap input simbol terjadi transisi yang unik dari setiap *state (Hopcroft dan Ullman, 1979)*. Konvensi yang digunakan untuk melambangkan sebuah DFA adalah M = {Q, Σ, δ, q_0, F} dengan :

- Q adalah himpunan *state* yang ada pada sebuah sistem DFA. State awal (q_0) dan himpunan *state* akhir *(F)* adalah himpunan bagian dari Q.
- Σ adalah alfabet masukan, bisa mencakup angka, abjad, tanda baca. Alfabet masukan untuk setiap sistem DFA dapat ditentukan.
- δ adalah fungsi transisi *state* untuk setiap masukan simbol/alfabet yang diberikan pada sistem DFA.

Stemming

Stemming akan menghilangkan akhiran dan/atau awalan untuk memperoleh akar kata/kata dasar (*stem*). Algoritme *stemming* yang paling banyak dikembangkan adalah *affix removal (penghilangan awalan/akhiran)* menggunakan *iterative longest match stemmer*. Algoritme ini menggunakan daftar awalan/akhiran dan kondisi/aturan yang menyebabkan awalan/akhiran dipotong. Algoritme dijalankan secara berurutan dari kondisi/aturan pertama, dan akan memotong saat ditemukan kondisi/aturan paling panjang yang cocok, untuk kemudian diuji kembali di kondisi/aturan berikutnya. Proses ini berlanjut hingga kondisi/aturan terakhir *(Frakes, 1992; Porter, 1980)*.

Algoritme Porter

Menurut Porter *(1980)*, terdapat beberapa definisi untuk sebuah algoritme pemotong akhiran.

1. ***Penghitungan ukuran kata***

   Setiap kata / bagian dari kata terdiri dari kombinasi konsonan dan vokal, dimana konsonan adalah huruf-huruf selain huruf vokal A, I, U, E dan O.

   Dalam algoritme ini, konsonan dinotasikan dengan k dan vokal dengan v. Kelompok konsonan dengan jumlah lebih dari 0 dinotasikan dengan K, sedangkan kelompok vokal dengan V. Sehingga setiap kata, atau bagian dari sebuah kata dapat direpresentasikan oleh sebuah bentuk umum :

   `[K] VKVK... [V]` .... (1)

   Notasi [ ] menunjukkan bahwa unsur yang didalamnya dapat muncul secara acak sebanyak n kali, untuk n = 0,1,2,... .

   Bentuk umum (1) dapat disederhanakan menjadi

   $[K]\ (VK)^m [V]$ .... (2)

   Notasi m melambangkan ukuran (*measure*) kata. Contoh dalam Bahasa Indonesia :

   ```
   m = 0   : YA, IA, SI
   m = 1   : AS, TAS, JIKA
   m = 2   : MAKAN, BENTUK
   m = 3   : PRESENTASI, APAKAH
   ```

   Notasi ukuran kata tidak memiliki basis linguistik *(Porter, 1980)*.

2. ***Aturan Pemotongan Akhiran***

   Bentuk aturan yang digunakan untuk menghilangkan akhiran kata *(sufiks)* adalah :

   `(kondisi) S1 -> S2`

   Aturan di atas bermakna, jika sebuah kata berakhiran S1, dan kata sebelum S1 (*stem*) memenuhi kondisi yang diberikan, maka S1 akan diganti dengan S2.

   Kondisi dapat juga mengandung 'AND', 'OR' dan 'NOT' untuk mengakomodasi kondisi yang kompleks. Hanya aturan yang memiliki S1 terpanjang yang paling cocok (*longest match*) yang akan digunakan untuk memotong sebuah kata, misalkan :

   ```
   (m > 1) KAN ->
   (m > 1) AN ->
   ```

   Kata JANGANKAN akan dipotong menjadi JANGAN bukan JANGANK, karena KAN memiliki S1 terpanjang yang cocok dengan aturan pertama.

Pola Suku Kata dalam Bahasa Indonesia

Dalam Bahasa Indonesia setiap suku kata ditandai oleh sebuah vokal yang dapat didahului maupun diikuti oleh konsonan. Beberapa pola suku kata Bahasa Indonesia yaitu : v, vk, kv, kvk, kkv, kkvk, vkk, kvkk, kkvkk, kkkv, kkkvk, dengan k adalah konsonan dan v adalah vokal (Departemen Pendidikan dan Kebudayaan, 1994).

Sufiks / Akhiran dalam Bahasa Indonesia

Menurut Keraf (1989), sufiks / akhiran yang terdapat dalam Bahasa Indonesia antara lain sufiks baku seperti –an, –i, –kan, –nya, –man, –wan, –wati, –nda, –anda dan sufiks asing/serapan (–at, –er, –asi, –si, –al, –isme, –is, –er, –if, –ir, –al, –logi, –tas, dan lainnya). Sedangkan dalam Kamus Besar Bahasa Indonesia (Departemen Pendidikan dan Kebudayaan ,1994), akhiran dalam Bahasa Indonesia antara lain –an, –i, –wi, –iah, –wiah, –ni, –kan, –man, –wan, –wati, –nda dan –anda

Sufiks –man tidak produktif lagi karena pembentukan nomina baru sering menggunakan sufiks –wan. (Alwi, *et al.* 1998).

Penilaian Algoritme Stemming

Menurut Frakes (1992) terdapat beberapa kriteria untuk menilai *stemmer*:

1. *Correctness*, *overstemming* terjadi jika terlalu banyak bagian kata yang dihilangkan. Sedangkan *understemming* terjadi jika terlalu sedikit bagian kata yang dihilangkan.
2. *Retrieval effectiveness*, didapat dari keefektifan suatu sistem temu kembali informasi dalam menemukembalikan dokumen yang relevan setelah ditambahkan modul *stemmer*.
3. *Compression Performance*, dinilai dari ukuran indeks yang dibuat dengan menggunakan *stemmer*.

Suatu *stemmer* biasanya tidak dinilai berdasarkan kebenaran tata bahasa.

METODOLOGI PENELITIAN

Bahan

Bahan yang digunakan adalah sufiks dan kata-kata dalam Bahasa Indonesia. Jumlah sufiks yang digunakan adalah 13, yaitu : –wan, –wati, –nya, –nda, –anda, –man, –wi, –wiah, –iah, –ni, –i, –kan, –an. Kata-kata yang digunakan dibagi menjadi tiga kelompok yaitu yang berakhiran baku (kelompok pertama), yang tidak berakhiran (kelompok kedua) dan yang tidak berakhiran tetapi mengandung unsur akhiran *(kelompok ketiga)*.

1. *Uji Pertama*

Bahan untuk uji pertama terdiri dari bahan untuk algoritme dengan M_P dan untuk algoritme dengan M_I.

2. *Uji Kedua*

Bahan untuk uji kedua adalah gabungan dari bahan algoritme M_P dan algoritme M_I.

Kata yang akan diuji diambil dari berbagai tulisan ilmiah, makalah penelitian dan Kamus Besar Bahasa Indonesia *(Departemen Pendidikan dan Kebudayaan, 2001)*. Data mengenai jumlah kata yang digunakan dapat dilihat pada ***Tabel 1***

Tabel 1. *Ringkasan jumlah data yang digunakan untuk pengujian algoritme*

Data	Kelompok			Total
	1	2	3	
Uji 1 algoritme M_P	136	283	155	574
Uji 1 algoritme M_I	134	260	152	547
Uji 2	153	326	165	646

Metode

Penelitian ini dibagi menjadi beberapa tahapan, yaitu :

1. Pembuatan algoritme pemotongan akhiran kata berbasis Bahasa Indonesia
2. Implementasi algoritme pada bahasa pemrograman
3. Pengujian algoritme *(program)* dan menganalisis keluarannya

Pembuatan Algoritme pemotongan akhiran kata berbasis Bahasa Indonesia

Algoritme pemotongan akhiran kata yang digunakan sebagai dasar adalah algoritme Porter yang kemudian dikembangkan menggunakan kaidah-kaidah Tata Bahasa Indonesia untuk akhiran baku. Dalam pembuatan algoritme ini, akan diuji dua jenis ukuran kata, yaitu :

- Ukuran kata yang berasal dari algoritme Porter, dinotasikan dengan M_P
- Ukuran kata yang disesuaikan dengan aturan penghitungan suku kata dalam Bahasa Indonesia, dinotasikan dengan M_I.

Kedua ukuran kata ini akan menghasilkan dua algoritme pemotong akhiran baku untuk kata dalam Bahasa Indonesia, yaitu algoritme pemotong akhiran dengan M_P dan algoritme pemotong akhiran dengan M_I.

Implementasi Algoritme pada Bahasa Pemrograman

Implementasi dilakukan dengan bahasa pemrograman Microsoft Visual Basic 6.0. Untuk membantu pemasukan data digunakan DBMS Microsoft Access Xp. Seluruh sistem dijalankan pada *platform* Windows.

Pengujian Program dan Menganalisis Hasil Keluarannya

Algoritme *stemmer* yang dibuat diuji berdasarkan kriteria pertama dari Frakes *(1992)*, yaitu *correctness (kebenaran)*.

Kedua algoritme *(algoritme M_P dan algoritme M_I)* yang telah diimplementasikan ke bahasa pemrograman diuji terhadap bahan penelitian :

- Uji pertama, terhadap datanya sendiri
- Uji kedua, terhadap data gabungan *(data $M_P \cup$ data M_I)*.

Dari keluaran program dihitung jumlah kata terpotong yang valid, jumlah kata terpotong yang tidak valid dan jumlah kata yang tidak terpotong akhirannya. Suatu kata dianggap valid jika terdapat dalam kamus. Selain itu, dari hasil uji pertama, dianalisis jenis kata yang *understemmed* dan *overstemmed* Pada uji kedua, analisis *correctness* tidak dilakukan karena hasil uji pertama sudah mencukupi untuk menjelaskan karakteristik algoritme.

HASIL DAN PEMBAHASAN

Ukuran Kata Berdasarkan Jumlah Suku Kata dalam Bahasa Indonesia

Gugus vokal pada persamaan *(2)* tidak dapat digantikan dengan notasi V karena setiap suku kata ditandai oleh vokal sehingga setiap vokal harus dianggap atomik(unsur tunggal). Sehingga jika mengikuti pola suku kata dalam Bahasa Indonesia, bentuk umum *(2)* di atas diubah menjadi

$$([K]v[K])^n \quad \text{.....................................} \quad (3)$$

Pada bentuk umum *(3)* diatas, notasi n adalah ukuran kata yang mewakili jumlah suku kata yang terdapat dalam suatu kata. Contohnya antara lain :

```
n = 1 : AS,TAS,BUS,TIK
n = 2 : JIKA,APA,MAKAN,LAMBANG
n = 3 : ULANGI,APAKAH,KONSONAN
```

Algoritme Pemotong Akhiran

Algoritme pemotong akhiran baku untuk kata berbasis Bahasa Indonesia dengan M_P adalah sebagai berikut :

```
Langkah 1
  akhiran -wan
  (m > 1) WAN ->

  akhiran -wati
  (m > 0) WATI ->

  akhiran -nya
  (m > 0) NYA ->

  akhiran -anda
  (m > 1 AND *k) -anda ->
  (m > 0 AND *v) -nda ->

Langkah 2
  Akhiran -wi
  (m > 0) WI ->

  Akhiran -wiah
  (m > 0) WIAH ->

  Akhiran -iah
  (m > 1) IAH ->

  Akhiran -ni
  (gereja or biksu) NI ->
  (sulta or bada) NI -> N

Langkah 3
  akhiran -i
  (m > 0) SI -> SI
  (m > 0) NI -> NI
  (m > 0 AND me*) I -> I
  (m > 0 AND NOT *kk) I ->

Langkah 4
  akhiran -man
  (budi or seni) MAN ->

  Akhiran -kan
  (m > 1) kan ->

  Akhiran -an
  (m > 1 and *MAN) -> MAN
  (m > 1) AN ->
```

Algoritme pemotong akhiran baku untuk kata berbasis Bahasa Indonesia dengan M_I adalah sebagai berikut.

```
Langkah 1
  akhiran -wan
  (n > 1) WAN ->

  akhiran -wati
  (n > 1) WATI ->

  akhiran -nya
  (n > 0) NYA ->
```

```
akhiran -nda / -anda
(n > 1 AND *k) -anda ->
(n > 1 AND *v) -nda ->
```

Langkah 2

```
Akhiran -wi
(n > 1) WI ->

Akhiran -wiah
(n > 1) WIAH ->

Akhiran -iah
(n > 1) IAH ->

Akhiran -ni
(gereja or biksu) NI ->
(sulta or bada) NI -> N
```

Langkah 3

```
akhiran -i
(n > 1) SI -> SI
(n > 1) NI -> NI
(n > 1 AND me*) I -> I
(n > 1 AND NOT *kk) I ->
```

Langkah 4

```
akhiran -man
(seni or budi) MAN ->

Akhiran -kan
(n > 1) KAN ->

Akhiran -an
(n > 1 and *MAN) -> MAN
(n > 1) AN ->
```

Beberapa aturan pemotongan dikembangkan dengan melihat karakteristik akhiran dan jumlah katanya, misalnya pada aturan –ni atau –man.

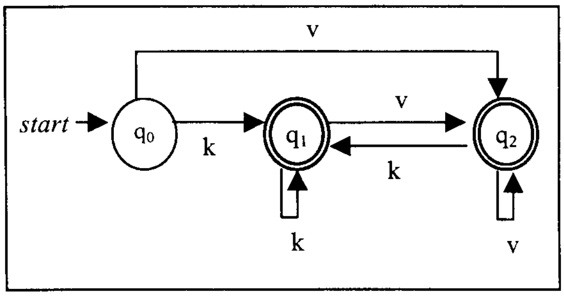
Setiap aturan dikelompokkan berdasarkan kedekatan ukuran kata. Akhiran yang mengandung unsur akhiran lain diletakkan pada langkah awal, seperti akhiran –ni, –wi, dan –man diletakkan sebelum akhiran –i dan –kan. Ukuran kata mencegah algoritme untuk memotong suatu kata jika akar katanya terlalu pendek.

Deterministic Finite Automata

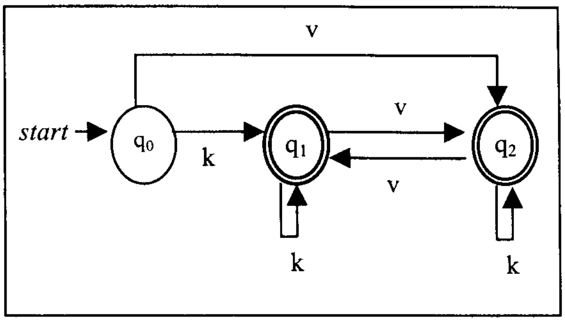
DFA yang digunakan untuk menghitung ukuran kata Porter (M_P) adalah P = {Q, Σ, δ, q_0, F}, dengan *state* Internal Q (q_0, q_1, q_2), himpunan masukan Σ {k, v} adalah semua alfabet dalam abjad yang dibagi kedalam dua kelompok yaitu konsonan (k) dan vokal (v), *state* awal $q_0 = [q_0]$, dan *state* akhir F= {$[q_1]$, $[q_2]$}. Fungsi pemetaan *state* δ adalah :

$$P([q_0], k) = [q_1], \quad P([q_0], v) = [q_2]$$
$$P([q_1], k) = [q_1], \quad P([q_1], v) = [q_2]$$
$$P([q_2], k) = [q_1], \quad P([q_2], v) = [q_2]$$

Diagram transisi untuk DFA P dapat dilihat pada ***gambar 1.***



Gambar 1. *Diagram transisi DFA penghitung ukuran kata M_P*



Gambar 2. *Diagram transisi DFA penghitung ukuran kata M_I*

Sedangkan DFA yang digunakan untuk menghitung ukuran kata Bahasa Indonesia (M_I) adalah M = {Q, Σ, δ, q_0, F}, dengan *state* Internal Q (q_0, q_1, q_2), himpunan masukan Σ {k, v} adalah semua alfabet dalam abjad yang dibagi kedalam dua kelompok yaitu konsonan (k) dan vokal (v), *state* awal $q_0 = [q_0]$, dan *state* akhir F= {$[q_1]$, $[q_2]$}. Fungsi pemetaan *state* δ adalah :

$$M([q_0], k) = [q_1] \quad M([q_0], v) = [q_2]$$
$$M([q_1], k) = [q_1] \quad M([q_1], v) = [q_2]$$
$$M([q_2], k) = [q_2] \quad M([q_2], v) = [q_1]$$

Diagram transisi untuk DFA M dapat dilihat pada ***gambar 2.***

Pada DFA P, penghitungan ukuran kata dilakukan saat terjadi perulangan vk *(perpindahan dari state 1 ke state 2),* sedangkan pada DFA M, penghitungan ukuran kata dilakukan pada saat terjadi perpindahan *state* karena masukan vokal.

Implementasi Algoritme

Kedua algoritme yang dibuat ditulis dalam modul terpisah dan setiap langkah ditulis dalam fungsi tersendiri. Selain fungsi untuk tiap langkah, terdapat fungsi lain untuk membantu algoritme pemotong akhiran

Pengujian Program dan Menganalisis Hasil Keluarannya

Program diuji dua kali, yaitu uji pertama *(terhadap datanya masing-masing)* dan uji kedua *(terhadap data gabungan).* Berikut ini akan disajikan hasil kedua uji tersebut.

Hasil Uji Pertama Algoritme Pemotong Akhiran dengan M_P

Untuk kelompok pertama *(kata berakhiran),* terlihat bahwa algoritme ini dapat mengenali hampir semua kata berakhiran yang dimasukkan. Dan dari kata-kata yang berhasil dipotong, hanya sedikit yang tidak memiliki arti *(tidak terdapat dalam kamus).* Untuk kelompok ketiga, algoritme mengalami kegagalan karena memotong 41,94% kata dasar yang mengandung unsur akhiran yang seharusnya tidak terpotong. Hasil pemotongan akhiran dengan M_P dapat dilihat pada ***Tabel 2.***

Tabel 2. *Hasil uji pertama untuk setiap kriteria pada data untuk algoritme dengan M_P (dalam persen).*

Kriteria	Kelompok		
	1	2	3
Terpotong-valid	86,03	0,00	0,00
Terpotong-tidak valid	2,94	0,00	41,94
Tidak terpotong	11,03	100,00	58,06

Analisis *correctness* dari algoritme pemotong akhiran ini adalah sebagai berikut.

1. *Data kata berakhiran (Kelompok Pertama M_P)*
   - *Overstemmed*
     Terdapat dua kata yang mengalami *overstemmed* pada kelompok ini, yaitu pada kata dengan unsur akhiran –nda yang mengandung unsur –i dengan ukuran kata 1, contohnya suaminda menjadi suam.
   - *Understemmed*
     Terdapat 16 kata yang mengalami *understemmed* pada kelompok ini. Ukuran kata merupakan penyebab utama *understemmed* pada kelompok ini, contohnya rugikan menjadi rugik.
2. *Data kata tidak berakhiran dengan unsur akhiran (Kelompok Ketiga M_P)*
   Penyebab utama terjadinya *overstemmed* 41,94% dari seluruh data adalah ukuran kata yang dijadikan kriteria pemotongan. Misalnya pada akhiran –wan, terdapat 10 kata yang overstemmed karena aturan pemotong akhiran –wan akan memotong kata-kata dengan ukuran lebih dari 0.

Hasil Uji Pertama Algoritme Pemotong Akhiran dengan M_I

Hasil pemotongan akhiran dengan M_P dapat dilihat pada *Tabel 3.*

Tabel 3. *Hasil uji pertama untuk setiap kriteria pada data untuk Algoritme dengan M_I (dalam persen).*

Kriteria	Kelompok		
	1	2	3
Terpotong-valid	88,06	0,00	0,00
Terpotong-tidak valid	2,24	0,00	38,82
Tidak terpotong	9,70	100,00	61,18

Pada ***Tabel*** *3*, pada kelompok pertama algoritme pemotong akhiran dengan M_I lebih baik dalam mengenali akhiran dan memotongnya. Begitu juga untuk kelompok ketiga, algoritme ini lebih baik karena tidak memotong 61,18% kata dasar yang mengandung unsur akhiran. Analisis *correctness* dari algoritme pemotong akhiran dengan M_I adalah sebagai berikut.

1. Data kata berakhiran *(Kelompok Pertama M_I)*
   - *Overstemmed*
     Overstemmed terjadi pada satu kata dengan unsur akhiran –nda yang mengandung unsur –i dengan ukuran kata lebih dari 2.
   - *Understemmed*
     Terdapat 13 kata yang mengalami *understemmed* pada kelompok ini. Penyebab terbesar *understemmed* pada data ini adalah aturan pemotongan akhiran –i, dan –kan.

2. Data kata tidak berakhiran dengan unsur akhiran *(Kelompok Ketiga M_I)*
   Jumlah kata yang *overstemmed* pada data ini mencapai 59 kata atau 38,82% dari seluruh data. Penyebab utama terjadinya *overstemmed* adalah ukuran kata yang dijadikan kriteria pemotongan.

Hasil uji pertama secara keseluruhan menunjukkan algoritme pemotong akhiran baku berbasis Bahasa Indonesia dengan menggunakan M_I memiliki kemampuan dan *correctness* yang lebih baik dibandingkan algoritme yang menggunakan M_P.

Hasil Uji Kedua
(Terhadap Data Gabungan)

Pada kelompok pertama *(kata berakhiran),* algoritme dengan M_I dapat memotong 137 kata *(89,54%)* dengan valid, memotong 3 kata *(1,96%)* dengan tidak valid, dan tidak memotong 13 kata *(8,50%).* Sedangkan pada kelompok ketiga *(kata dasar yang memiliki unsur akhiran),* algoritme ini hanya memotong 66 kata *(40%)* dari 165 kata. Hasil ini lebih baik dibandingkan algoritme dengan M_P, dengan perbedaan yang kecil *(tidak mencapai 5%).* Hal ini dapat dilihat pada ***Tabel 4*** dan ***Tabel 5.***

Tabel 4. *Persentase untuk setiap kriteria pada data gabungan untuk Algoritme dengan M_P.*

Kriteria	Kelompok		
	1	2	3
Terpotong-valid	85,62	0	0
Terpotong-tidak valid	3,27	0	42,42
Tidak terpotong	11,11	100	57,58

Tabel 5. *Persentase untuk setiap kriteria pada data gabungan untuk Algoritme dengan M_I.*

Kriteria	Kelompok		
	1	2	3
Terpotong-valid	89,54	0	0
Terpotong-tidak valid	1,96	0	40
Tidak terpotong	8,50	100	60

PENUTUP

Stemming diperlukan dalam suatu sistem temu kembali informasi berbasis teks untuk memperkecil indeks dan meningkatkan efektivitas sistem temu kembali informasi. Dari algoritme pemotong akhiran yang dibuat oleh Porter, dapat ditelaah kemungkinan pengembangan algoritme pemotong akhiran baku untuk kata dalam Bahasa Indonesia. Untuk membuat aturan pemotongan akhiran, yang harus dipertimbangkan adalah ukuran kata, karakteristik akhiran dan jumlah kata yang menggunakan akhiran tersebut.

Dari kedua hasil uji, cara penghitungan ukuran kata yang sesuai dengan cara penghitungan suku kata dalam Bahasa Indonesia dapat meningkatkan keberhasilan dan *correctness* dibandingkan dengan menggunakan cara penghitungan ukuran kata dari algoritme Porter.

Algoritme pemotong akhiran ini masih memotong sejumlah kata dasar yang seharusnya tidak terpotong. Untuk itu dapat ditambahkan basisdata kata-kata dasar untuk mencegah kata dasar yang seharusnya tidak terpotong. Jika kata-kata yang akan dipotong tidak ditemukan dalam basisdata ini, maka kata-kata tersebut akan dipotong melalui algoritme pemotong akhiran ini.

Algoritme pemotong akhiran ini belum mengakomodasi akhiran asing dan awalan, sehingga akan lebih baik jika dilengkapi dengan akhiran serapan/asing dan awalan, sehingga dapat menjadi sebuah *stemmer* yang lengkap.

Aturan-aturan pemotongan yang dibuat masih dapat disempurnakan agar lebih banyak memotong kata berakhiran dan tidak memotong kata tidak berakhiran dengan unsur akhiran. Misalnya untuk aturan akhiran –kan ditambahkan aturan :

```
(m = 1 and *K) KAN →
```

Aturan ini akan memotong semua akhiran –kan dengan ukuran 1 yang kata dasarnya berakhir dengan konsonan.

Walaupun tidak diuji berdasarkan kebenaran linguistik, tetapi sebuah *stemmer* akan lebih baik jika dapat mempelajari data kata di lingkungan pemakainya sehingga dapat menghasilkan lebih banyak kata-kata yang valid dan sesuai dengan lingkungan penggunanya.

DAFTAR PUSTAKA

Alwi, Hasan., S. Darwowidjojo, H. Lapoliwa, & A.M. Moeliono. 1998. *Tata Bahasa Baku*

Bahasa Indonesia Edisi Ketiga. Balai Pustaka, Jakarta.

Hopcroft, John E. & Jeffrey D. Ullman. 1979. *Introduction to Automata Theory, Languages, And Computation*. Addison-Wesley Publishing Company Inc.

Departemen Pendidikan dan Kebudayaan. 1994. *Kamus Besar Bahasa Indonesia, Edisi Kedua*. Balai Pustaka, Jakarta.

Departemen Pendidikan dan Kebudayaan. 2001. *Kamus Besar Bahasa Indonesia, Edisi Ketiga*. Balai Pustaka, Jakarta.

Frakes, William B. 1992. *Stemming Algorithm : Information Retrieval, Data Structures and Algorithm*. Prentice-Hall Inc, New York.

Keraf, Gorys. 1989. *Tata Bahasa Indonesia*. Penerbit Nusa Indah, Flores – NTT.

Porter, M.F. 1980. *An Algorithm For Suffix Stripping*. Program, Juli 1980, 14(3), 130-137.

van Rijsbergen, C.J. 1979. *Information Retrieval, Second Edition*. Butterworths. London.