PENGINDEKSAN OTOMATIS DENGAN ISTILAH TUNGGAL UNTUK DOKUMEN BERBAHASA INDONESIA

Ahmad Ridha*, Ir. Julio Adisantoso, M.Komp.**, Ir. Fahren Bukhari, M.Sc.***
Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor
Jl. Raya Pajajaran, Bogor 16127, Indonesia
*ridha@ilkom.fmipa.ipb.ac.id
**julio@bima.ipb.ac.id
***fahren@ciam.or.id

ABSTRAK

Pengindeksan memiliki peranan penting dalam sistem temu-kembali informasi untuk menyediakan pemrosesan kueri yang lebih cepat dan daftar terurut dokumen hasil temu-kembali. Tujuan penelitian ini adalah mengembangkan metode pengindeksan dengan istilah tunggal untuk dokumen-dokumen dalam Bahasa Indonesia. Tercakup di dalamnya proses *parsing*, *stemming*, dan pembentukan *inverted index* dengan pembobotan istilah menggunakan fungsi *tf·idf*. Pengujian menggunakan 422 dokumen dan sepuluh kueri (yang masing-masing diungkapkan dalam tiga bentuk) dan diukur dengan 10-pt *average precision*. Ternyata penggunaan daftar kata buang dan *stemming* tidak berpengaruh signifikan terhadap kinerja temu-kembali. Akan tetapi daftar kata buang dan *stemming* bermanfaat dalam sistem temu-kembali untuk mengurangi ukuran indeks yang akan menjadikan operasi pencarian lebih efisien. Penggunaan daftar kata buang, *stemming*, dan keduanya menghasilkan penurunan lebih dari 25%, 10%, dan 30% masing-masingnya.

Kata kunci: sistem temu kembali informasi, indeks, *stemming*.

1. PENDAHULUAN

Penyimpanan dokumen secara digital berkembang dengan pesat seiring meningkatnya penggunaan komputer. Kondisi tersebut memunculkan masalah untuk mengakses informasi yang diinginkan secara akurat dan cepat sehingga pencarian terhadap seluruh isi dokumen yang tersimpan bukanlah solusi yang tepat mengingat pertumbuhan ukuran data yang tersimpan umumnya sangat tinggi. Dalam pencarian informasi di Internet para pengguna di Inggris menjadi frustrasi dalam dua belas menit jika tidak menemukan yang diinginkannya [1].

Untuk memenuhi tuntutan tersebut dibutuhkan *information retrieval system* (IRS) yang menggunakan suatu pengganti yang dapat merepresentasikan kumpulan dokumen dalam bentuk dan ukuran yang lebih mudah untuk pencarian. Struktur data yang populer dan telah lama digunakan untuk keperluan tersebut adalah sebuah indeks, yakni gugus kata atau konsep terpilih sebagai penunjuk ke informasi (atau dokumen) terkait. Indeks, dalam berbagai bentuk, merupakan inti setiap IRS modern karena menyediakan akses yang lebih cepat ke data dan juga mempercepat pemrosesan kueri [2].

Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan pengindeksan dengan istilah tunggal untuk digunakan dalam IRS untuk dokumen-dokumen teks berbahasa Indonesia.

Penelitian ini terbatas pada pemrosesan dokumen teks berbahasa Indonesia menjadi indeks istilah tunggal meliputi proses *parsing* serta *stemming*. Struktur data serta metode penyimpanan indeks tidak termasuk dalam penelitian ini.

2. PENGINDEKSAN OTOMATIS DENGAN ISTILAH TUNGGAL

Proses pengindeksan dilakukan dalam struktur *inverted index* dan menggunakan pembobotan *tf·idf*. Langkah-langkah dalam pengindeksan adalah sebagai berikut:

1. *inverted index* dikosongkan
2. dokumen diproses hingga menjadi *document terms*
3. untuk tiap *stem* pada *document terms*, tambahkan *posting list node* pada *posting list* yang bersesuaian dalam kamus istilah.
4. simpan informasi panjang dokumen (jumlah kata) pada kamus dokumen
5. proses dokumen berikutnya hingga seluruh koleksi telah ditambahkan pada indeks
6. lakukan pembobotan untuk seluruh isi kamus istilah dan hitung faktor normalisasi tiap dokumen untuk pembobotan *tf·idf*.

7. simpan faktor normalisasi untuk setiap dokumen dalam kamus dokumen.

Untuk penambahan dokumen tunggal dilakukan langkah 2-7 karena terjadi perubahan frekuensi istilah dan dengan sendirinya perubahan bobot istilah dan faktor normalisasi secara keseluruhan.

2.1. *Tokenizer*

Tokenizer menerima masukan berupa rangkaian karakter dan memilahnya menjadi *token* dengan aturan sebagai berikut:

- Suatu *token* dimulai oleh huruf atau angka
- *Token* dipisahkan oleh karakter *whitespace*
- Karakter-karakter khusus yang mengikuti huruf atau angka dianggap bagian dari *token* (misalnya tanda persen dalam 125%) namun dianggap sebagai pemisah *token* jika tidak.

2.2. *Stemming*

Stemming merupakan bagian yang sangat memerlukan pengetahuan bahasa karena penentuan *stem* suatu kata berbeda tergantung tata bahasa yang digunakan. Oleh karena itu perlu dikembangkan algoritme *stemming* tersendiri untuk Bahasa Indonesia.

Sistem pemotong sufiks untuk Bahasa Indonesia yang berdasarkan algoritme Porter [3] telah dikembangkan dalam [4] namun pengindeksan memerlukan sistem yang mampu memotong prefiks dan sufiks yang banyak digunakan. Infiks tidak dihilangkan karena prosesnya lebih kompleks dan tidak lagi produktif dalam Bahasa Indonesia.

Sebagaimana algoritme Porter, digunakan suatu fungsi penghitung ukuran kata untuk mencegah *stemming* menghasilkan *stem* yang terlalu pendek. Diasumsikan minimal *stem* hasil berukuran dua kecuali jika *token* berukuran kurang dari dua.

Akan tetapi fungsi ukuran kata pada algoritme Porter tidak dapat digunakan pada Bahasa Indonesia. Sebagaimana dalam [4], jumlah vokal dalam kata akan digunakan sebagai penentu ukuran kata kecuali kata-kata tanpa vokal yang terdiri dari tiga karakter atau lebih dianggap memiliki ukuran dua untuk mengakomodasi singkatan yang hanya terdiri dari konsonan.

Vokal didefinisikan sebagai huruf-huruf A, I, U, E, dan O. Huruf-huruf selain itu merupakan konsonan.

Aturan pemotongan diungkapkan sebagai:

$P1\ (kondisi)\ S1 \rightarrow P2\ S2$

yang artinya jika sebuah kata berprefiks P1 dan bersufiks S1, dan bagian kata setelah P1 dan sebelum S1 memenuhi kondisi yang diberikan, maka P1 dan S1 akan diganti dengan P2 dan S2.

Kondisi dapat menggunakan operator *AND*, *OR*, atau *NOT* untuk menyatakan aturan yang kompleks.

Beberapa notasi juga digunakan untuk membantu, yakni:

- *W*, seluruh kata termasuk P1 dan S1
- *M*, ukuran kata
- *L*, jumlah karakter dalam kata
- *V*, huruf vokal
- *C*, huruf konsonan
- *V**, kata diawali vokal
- *C**, kata diawali konsonan
- **CC*, kata diakhiri konsonan ganda
- *x**, kata diawali huruf atau kumpulan huruf *x*
- **x*, kata diakhiri huruf atau kumpulan huruf *x*
- *V*(*x*), huruf ke-*x* adalah vokal
- *C*(*x*), huruf ke-*x* adalah konsonan

Sebagai contoh, dalam aturan:

$(M > 1)\ wan \rightarrow$

S1 adalah *wan* dan S2 adalah *null* (kata kosong). Sehingga kata *dermawan* dipotong menjadi *derma* karena *derma* berukuran 2 ($M > 1$).

Stemming dilakukan terhadap elemen-elemen berikut:

- prefiks: *meng-*, *di-*, *per-*, *ber-*, *ter-*, *peng-*, *pe-*, *per-*, *se-*
- sufiks: *-an*, *-kan*, *-i*, *-nya*, *-ik*, *-is*, *-if*, *-al*, *-(is)asi*, *-at*, *-iah*, *-wi*, *-wiah*, *-isme*, *-sionis*
- konfiks: *ke-an*, *ke-i*
- partikel: *-kah*, *-lah*
- kata ganti: *ku-*, *kau-*, *-mu*, *-nya*

Walaupun partikel dan kata ganti tidak termasuk afiks namun diperlakukan sama sehingga partikel dianggap sebagai sufiks dan kata ganti dianggap sebagai prefiks atau sufiks sesuai posisinya.

Selanjutnya walaupun *stemming* umumnya hanya digunakan untuk memotong afiks suatu kata namun dalam sistem ini fungsi serupa juga diterapkan pada angka. *Token* berupa angka dikelompokkan ke dalam bentuk yang lebih umum misalnya 800.000 dan 796.352 menjadi bentuk yang sama yakni 800000.

2.4. Koleksi Pengujian

Koleksi pengujian menggunakan artikel-artikel utama, nasional (politik dan keamanan serta umum pada Media Indonesia) dan internasional harian Kompas, harian Media Indonesia, dan harian Koran Tempo (http://www.kompas.com, http://www.media-indonesia.com, dan http://www.tempo.co.id) terbitan 6 April 2002 dan 8-12 April 2002. Terbitan tanggal 7 April 2002 tidak disertakan karena merupakan harian Minggu yang memiliki susunan berita berbeda. Tiga media massa digunakan untuk menguji sistem pada dokumen-dokumen yang memiliki topik serupa namun dengan penyampaian yang berbeda-beda.

Tabel 1. Jumlah dokumen yang digunakan berdasarkan tanggal.

Sumber	Tanggal						Σ
	6*	8	9	10	11	12	
Kompas	31	23	23	24	28	27	156
Koran Tempo	21	28	18	24	18	16	125
Media Indonesia	25	25	23	21	24	23	141
Total	77	76	64	69	70	66	422

*April 2002

Untuk pengindeksan, dokumen-dokumen tersebut diubah susunannya menjadi terdiri dari:

1. judul (satu baris)
2. isi dokumen.

Sedangkan isi dokumen hanya mengalami perubahan dalam penggantian tanda *ampersand* (&) menjadi kata *dan* serta penggantian karakter dengan kode ASCII 173 menjadi tanda hubung (-). Kesalahan ejaan dan kesalahan tata bahasa tidak diperbaiki.

2.5. Evaluasi Sistem

Evaluasi pengindeksan otomatis dilakukan dengan menentukan kinerjanya dalam *recall* dan *precision*. Hal ini dilakukan dengan menggunakan koleksi pengujian beserta gugus kueri dan penilaian relevansinya (gugus jawaban) [5]. Dari hasil evaluasi tersebut dapat diperoleh nilai *average precision* (AVP).

Relevansi dokumen umumnya ditentukan oleh manusia, sebaiknya oleh orang yang sama yang memberikan kueri. Walaupun penilaian relevansi tersebut akan berbeda-beda bagi pemeriksa yang berbeda namun [6] menunjukkan bahwa kurva *recall* dan *precision* yang dihasilkan hampir identik. Metode serupa juga digunakan dalam TREC [7].

Sistem mengembalikan daftar dokumen terurut menurun berdasarkan besar hasil fungsi kesamaan kueri dan dokumen.

Selanjutnya dari hasil kueri dihitung banyak dokumen yang diperoleh untuk mencapai tingkat *recall* tertentu dan selanjutnya nilai *precision* dihitung. Tingkat *recall* yang digunakan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan 1.0 untuk menghitung AVP.

Untuk melihat pengaruh penggunaan *stemming* dan daftar kata buang dilakukan empat kali pengindeksan yakni:

1. IDX_A: menggunakan *stemming* dan daftar kata buang
2. IDX_B: hanya menggunakan *stemming*
3. IDX_C: hanya menggunakan daftar kata buang
4. IDX_D: tidak menggunakan *stemming* dan daftar kata buang.

Untuk membandingkan algoritme *stemming*, juga dilakukan pengindeksan $IDX_{A'}$ yang menggunakan *stemming* sufiks dalam [4] dan daftar kata buang.

Selanjutnya beberapa kueri beserta gugus jawaban dibuat berdasarkan koleksi pengujian. Kueri yang sama diungkapkan dalam bentuk-bentuk berikut:

1. Bentuk menengah, mengungkapkan topik yang diinginkan dalam kalimat
2. Bentuk pendek, hanya terdiri dari sebanyak-banyaknya lima kata yang merupakan inti dari bentuk menengah
3. Bentuk panjang, paragraf yang diambil dari salah satu dokumen yang relevan.

Pengujian dilakukan dengan uji *t* berpasangan dengan selang kepercayaan 95% terhadap:

1. IDX_A - IDX_C dan IDX_B - IDX_D: untuk pengaruh *stemming*
2. IDX_A - IDX_B dan IDX_C - IDX_D: untuk pengaruh daftar kata buang
3. IDX_A - IDX_D: untuk pengaruh keduanya
4. IDX_A - $IDX_{A'}$: untuk pengaruh perbedaan algoritme *stemming*

Selain kemampuan temu-kembali, juga dilihat ukuran indeks yang dihasilkan. Digunakan dua jenis ukuran yakni yang menyertakan keterangan posisi suatu istilah (S_P) dan yang tidak (S_{NP}).

2.6. Asumsi-asumsi

Asumsi-asumsi yang digunakan dalam penelitian ini adalah sebagai berikut:

1. indeks dapat termuat dalam memori utama
2. dokumen dan teks kueri menggunakan ASCII *character set*.

3. seluruh koleksi telah terindeks sebelum digunakan oleh untuk pemrosesan kueri.

3. HASIL EKSPERIMEN

3.1. Algoritme *Tokenizer*

Teks masukan diproses secara sekuensial per karakter dari awal dan menghasilkan sebuah *token* serta posisinya atau keterangan bahwa teks telah selesai diproses. Algoritme yang digunakan sebagai berikut:

1. Jika sudah mencapai akhir teks maka proses berakhir selainnya karakter yang diperoleh dibandingkan terhadap tabel jenis karakter. Sebuah karakter dapat memiliki salah satu di antara tiga jenis berikut:
   a. *whitespace*, berarti karakter ini merupakan karakter pemisah *token*
   b. *alphanumeric*, berarti karakter ini merupakan huruf atau angka
   c. *other*, berarti karakter ini tidak termasuk jenis-jenis di atas.
2. Jika karakter yang ditemukan merupakan huruf atau angka maka karakter berikut menjadi karakter pertama dari *token* sedangkan selainnya kembali ke langkah pertama.
3. Karakter-karakter selanjutnya menjadi bagian dari *token* hingga ditemukan karakter *whitespace* atau akhir dari teks.

3.2. Masalah-masalah Afiksasi

Proses afiksasi terutama prefiks dalam Bahasa Indonesia mengalami proses morfofonemik sesuai dengan fonem yang mengikutinya. Misalnya prefiks *meng-* dapat menjadi *meng-*, *me-*, *men-*, *mem-*, *meny-* atau *menge-*. Selain itu beberapa fonem juga mengalami peluluhan. Hal ini menyulitkan proses *stemming* karena satu bentuk perubahan dapat ditimbulkan oleh beberapa fonem, sehingga pemotongan afiks secara sempurna tanpa perbendaharaan kata yang lengkap sangat sulit dilakukan.

Untuk mengatasi masalah di atas, dilakukan perubahan-perubahan prefiks tertentu terhadap *stem* sebagai berikut:

1. $ny\,(M>0) \rightarrow s$
2. $V\,(M>0) \rightarrow ng$
3. $k\,(M>0) \rightarrow ng$
4. $p\,(M>0) \rightarrow m$
5. $t\,(M>0) \rightarrow n$

Selanjutnya masalah afiksasi pada sufiks *-an* dan *-kan*. Algoritme *stemming* yang dikembangkan berdasarkan algoritme Porter yang bersifat *iterative longest match* sehingga dapat terjadi kesalahan pemotongan seperti contoh berikut:

hentakan → *henta* + *-kan*

seharusnya

hentakan → *hentak* + *-an*

Sebagaimana masalah proses morfofonemik di atas, dilakukan perubahan pada akhir *stem* yakni:

$(M>1)\ k \rightarrow$

sehingga bentuk-bentuk *hentak*, *hentakan*, dan *hentakkan* menghasilkan *stem* yang sama yakni *henta*.

Walaupun cara-cara di atas dapat dikatakan *merusak* bentuk *stem* namun karena tujuan *stemming* di sini adalah untuk sedapat mungkin menjadikan bentuk-bentuk kata yang diturunkan dari kata dasar yang sama ke dalam satu *stem* dan hasil *stemming* ditujukan untuk sistem (tidak untuk pengguna) maka bentuk yang *aneh* tersebut dapat digunakan. Asumsinya adalah tidak banyak kata-kata yang memiliki bentuk yang sama dengan hasil perubahan. Kesalahan terjadi dalam kasus seperti berikut:

pakan → *makan*
galak → *gala*

padahal terdapat kata *makan* dan *gala* yang memiliki makna yang berbeda dengan *pakan* dan *gala*.

Selain itu masih ada kasus yang belum tertangani yakni proses morfofonemik pada prefiks *ber-*, *ter-*, dan *per-* terhadap fonem vokal dan /r/ sehingga misalnya bentuk-bentuk *merebut*, dan *terebut*, menghasilkan *stem* yang berbeda yakni *rebut* dan *ebut* (atau *ngebut* dengan perubahan di atas).

3.3. Algoritme *Stemming*

Beberapa fungsi pendukung yang digunakan dalam *stemming* antara lain:

1. *Valid* (x), memeriksa kevalidan suatu *token* masukan untuk diproses lebih lanjut atau kevalidan suatu *stem* untuk digunakan. *Token* atau *stem* x valid jika memiliki:

- panjang lebih dari dua karakter dan memiliki huruf (misalnya *IBM* dan *M16*), atau
- panjang lebih dari tiga karakter dan tidak memiliki huruf namun memiliki angka (misalnya *2002* dan *1987-1992*).

2. *ReduceRep* (x), menangani kata x yang memiliki tanda ulang (-) dengan cara:
   - jika kata sebelum tanda ulang tersebut termasuk morfem terikat tertentu.
   - jika ukuran kata sebelum tanda ulang lebih dari satu atau kata sebelum tanda ulang sama dengan kata setelah tanda ulang maka yang digunakan hanyalah bagian sebelum tanda ulang (misalnya *lalu-lalang* → *lalu*, *hak-hak* → *hak*).
   - jika tidak termasuk dalam kedua bentuk di atas maka tanda ulang dipertahankan (misalnya *F-15* → *F-15*).
3. *ValidDblConsonant* (x), memeriksa kevalidan konsonan ganda yang mengawali kata x. Dalam Bahasa Indonesia konsonan ganda yang mengawali suatu suku kata terbatas pada *pl, bl, kl, gl, fl, sl, pr, br, tr, dr, kr, gr, fr, sr, ps, sw, kw, sp, sm, sn, sk, pt, ts, st, ng, ny, str, spr, skr,* dan *skl* (Alwi *et al*, 1998).
4. *AdjustHead* (x) dan *AdjustTail* (x), mengubah huruf awal dan akhir dari *stem* x untuk menangani masalah-masalah afiksasi.
5. *Right*(x, y), mengembalikan y karakter paling kanan dari kata x
6. *Left*(x, y), mengembalikan y karakter paling kiri dari kata x
7. *Mid*(x, y, z), mengembalikan z karakter dari kata x mulai posisi y. Jika z tidak diberikan maka semua karakter dari y hingga akhir dikembalikan.
8. *StripPrefix* (x, y), melakukan pemotongan prefiks terhadap kata x pada posisi y dengan aturan:
   - jika pada posisi y terdapat tanda ulang (-) maka dilakukan pemotongan y karakter, selainnya
   - jika ($V(y)$ *OR* ($C(y)$ *AND* $V(y+1)$) *OR* *ValidDblConsonant* (*Mid*(x, y))) maka dilakukan pemotongan (y-1) karakter, selainnya
   - tidak dilakukan pemotongan terhadap x

Aturan-aturan *stemming* yang digunakan adalah sebagai berikut:

1. PreS1
   se (M > 1) →

2. PreS2
   mem (M > 1 *AND* (*b* OR p* OR f**)) →
   mem (M > 1) → *m*
   meng (M > 1 *AND* (*g* OR h* OR kh**)) →
   meny (M > 1 *AND V**) → *s*
   men (M > 1 *AND V**) → *n*
   men (M > 1) →
   me (M > 1) → *StripPrefix*(*W*, 3)

   di (M > 1) → *StripPrefix*(*W*, 3)

3. PreS3
   ber (M > 1) →
   be (M > 1 *AND Cer**) →

4. PreS4
   pem (M > 1 *AND b**) →
   peng (M > 1 *AND* (*g* OR h* OR kh**)) →
   peny (M > 1 *AND V**) → *s*
   pen (M > 1 *AND V**) → *n*
   pen (M > 1) →
   per (M > 1 *AND C**) → *StripPrefix*(*W*, 3)
   pe (M > 1) → *StripPrefix*(*W*, 3)

   ter (M > 1) →
   te (M > 1 *AND Cer**) →

5. SufS1
   (*seni OR budi*) *man* →
   (M > 1) *wan* →
   (M > 1) *wati* →

6. SufS2
   (L > 4) *-kan* →
   (M > 1) *kan* →

7. SufS3
   (L > 3) *-an* →
   (M > 1) *an* →

8. SufS4
   (M > 1 *AND NOT*(**i*) *AND*
   (**ng OR NOT*(**CC*))) *i* →

9. FSufS1
   (M > 1) *sionis* → *si*
   (L > 5) *-isme* → *is*
   (M > 0) *isme* → *is*
   (M > 1) *itas* →
   (M > 1) *asi* →
   (M > 1 *AND *C*) *si* → *t*
   (M > 1) *or* →
   (M > 1) *er* →

10. FSufS2
    (M > 1) *if* →
    (M > 1) *ik* →
    (M > 1) *is* →

11. FSuf3
$(M > 1)\ at \rightarrow$
$(M > 1)\ wi \rightarrow$
$(M > 1)\ wiah \rightarrow$
$(M > 1)\ iah \rightarrow$
$(M > 1)\ al \rightarrow$

12. FSuf4
$(M > 0\ AND\ {*V})\ v \rightarrow f$
$(M > 0\ AND\ {*V})\ pt \rightarrow p$
$(M > 0\ AND\ {*V})\ kt \rightarrow k$
$(M > 0\ AND\ {*V})\ nt \rightarrow n$

13. ConS
$ke\ (M > 1)\ an \rightarrow$
$ke\ (tahu)\ i \rightarrow$

14. ParS
$(M > 1)\ -kah \rightarrow$
$(M > 1)\ -lah \rightarrow$

15. ProS
$(M > 1)\ -ku \rightarrow$
$(M > 1)\ -mu \rightarrow$
$(M > 1)\ ku \rightarrow$
$(M > 1)\ mu \rightarrow$
$(M > 1)\ -nya \rightarrow$
$(M > 1)\ nya \rightarrow$
$ku\ (M > 1) \rightarrow$
$kau\ (M > 1) \rightarrow$

Dengan menggunakan fungsi-fungsi dan aturan-aturan tersebut *stemming* terhadap *token* t dilakukan dengan algoritme berikut:

1. t diubah menjadi *lower case*
2. *StripPunct*(t)
3. jika t diawali oleh tanda rupiah (*rp*) dan bersifat numerik maka *rp* dihilangkan
4. jika *Valid*(t) tidak benar maka proses berakhir
5. *ReduceRep*(t)
6. jika ukuran t kurang dari dua dan tidak mengandung tanda ulang maka proses berakhir
7. aturan-aturan *stemming* digunakan dengan urutan:
   - PreS1
   - ParS
   - ProS
   - ConS
   - PreS2
   - ConS
   - SufS1 – SufS3
   - PreS3 – PreS4
   - FSuf1 - FSuf4
   - SufS4
8. *AdjustHead*(t)
9. *AdjustTail*(t)
10. jika ada, hilangkan tanda ulang di sisi kiri t
11. *StripPunct*(t)
12. panjang t dibatasi maksimal 15 karakter dan jika lebih diambil 15 karakter paling kiri
13. jika t bersifat numerik maka diubah dengan pembulatan kemudian selain dua digit paling kiri diganti dengan nol
14. jika *Valid*(t) benar maka hasil stemming dikembalikan

3.4. Pemrosesan Dokumen dan Teks Kueri

Selain *tokenizer* dan *stemming*, digunakan fungsi *StripPunct* untuk membuang karakter-karakter tanda baca berikut dari sisi kanan: . , ? ! - : ;)] } > serta menghilangkan semua kemunculan tanda kutip satu (') dan tanda kutip dua (") dari *token*. Misalnya:

"Serang!" → *Serang*
Ma'ruf → *Maruf*

Baik dokumen maupun teks kueri diproses dengan langkah-langkah berikut:

1. dengan menggunakan *tokenizer*, *token* dikenali dan dimasukkan ke fungsi *StripPunct*.
2. *token* diubah ke *lower case*, jika *token* termasuk dalam daftar kata buang maka *token* tersebut dilewati dan langsung ke langkah 4. Jika tidak maka *token tersebut* mengalami *stemming*
3. *stem* dimasukkan dalam kamus istilah dan informasinya disesuaikan
4. jika teks belum berakhir kembali ke langkah 1.

Perbedaan pemrosesan dokumen dan teks kueri berada pada informasi yang disertakan dalam kamus istilah. Daftar istilah teks kueri hanya menyertakan informasi frekuensi istilah namun daftar istilah dokumen menyertakan informasi berupa *posting list node* (tetapi informasi bobot tentunya belum tersedia). Daftar istilah dokumen selanjutnya dirujuk sebagai *document terms* dan daftar istilah kueri disebut *query terms*.

3.5. Pemrosesan kueri

Pembobotan kueri menggunakan VSM dengan vektor dokumen ternormalisasi. Bobot istilah kueri w_{qj} diperoleh dengan cara serupa dengan pembobotan istilah dokumen namun frekuensi yang digunakan adalah frekuensi istilah T_j dalam kueri.

$$w_{qj} = tf_{qj} \cdot \log \frac{N}{df_j}$$

sehingga istilah yang tidak terdapat dalam kueri ($tf_{qj} = 0$) memiliki bobot nol.

3.6. Ukuran Indeks dan Jumlah Istilah

Deskripsi koleksi yang diproses dapat dilihat pada Tabel 2.

Tabel 2. Deskripsi koleksi pengujian.

Deskripsi	Total	Rata-rata
Ukuran*	1.558.884	3.694,038
Total *token*	210.622	499,104

*dalam *byte*

Tokenizer menghasilkan 16.442 *token* unik dengan frekuensi total sebesar 210.622. Penggunaan daftar kata buang mengurangi 250 *token*, yakni hanya 1,522% dari jumlah *token* unik, namun dengan frekuensi total sebesar 69.106, mencapai 32,81% dari frekuensi total.

Hal ini berdampak langsung terhadap S_P yang mengalami penurunan sebesar 25,80% ($IDX_B \rightarrow IDX_A$) dan 26,20% ($IDX_D \rightarrow IDX_C$) serta S_{NP} yang mengalami penurunan sebesar 21,27% ($IDX_B \rightarrow IDX_A$) dan 21,84% ($IDX_D \rightarrow IDX_C$).

Stemming turut mengurangi jumlah istilah indeks sebesar 41,671% (IDX_A) dan sebesar 41,68% (IDX_B) yang selanjutnya menurunkan ukuran indeks S_P sebesar 10,40% ($IDX_C \rightarrow IDX_A$) dan 10,89% ($IDX_D \rightarrow IDX_B$) serta S_{NP} yang mengalami penurunan sebesar 16.29% ($IDX_C \rightarrow IDX_A$) dan 16,89% ($IDX_D \rightarrow IDX_B$).

Sebagai perbandingan, $IDX_{A'}$ mengurangi jumlah istilah indeks sebesar 14,995% serta menurunkan S_P sebesar 4,625% ($IDX_C \rightarrow IDX_{A'}$) dan S_{NP} sebesar 6,249% ($IDX_C \rightarrow IDX_{A'}$). Hal ini menunjukkan tingginya penggunaan prefiks dalam afiksasi.

Secara keseluruhan, penggunaan daftar kata buang dan *stemming* ($IDX_D \rightarrow IDX_A$) menurunkan ukuran indeks S_P sebesar 33,88% dan S_{NP} sebesar 34,57%.

3.7. Kinerja Temu-Kembali

Kinerja temu-kembali masing-masing indeks dapat dilihat pada Tabel 3.

Tabel 3. Ringkasan kinerja temu-kembali.

Indeks	AVP
IDX_A	0,775
IDX_B	0,776
IDX_C	0,771
IDX_D	0,771
$IDX_{A'}$	0,773

Hasil uji menunjukkan bahwa tidak ditemukan perbedaan yang signifikan dalam kinerja temu-kembali. Perbedaan algoritme *stemming* ($IDX_A - IDX_{A'}$) juga tidak menghasilkan perbedaan kinerja yang signifikan.

Kinerjanya secara umum dapat dikatakan baik karena dengan AVP sekitar 0,77 berarti secara rata-rata pada tiap *recall point*, 77% hasil temu-kembali relevan dengan kueri. Akan tetapi hal ini dapat juga ditimbulkan oleh kecilnya ukuran koleksi sehingga memiliki *noise* yang rendah.

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa:

1. Daftar kata buang dan *stemming* berperan untuk memperkecil ukuran indeks sehingga meningkatkan efisiensi operasi temu-kembali.
2. Penggunaan *stemming* prefiks dan sufiks penting bagi IRS Bahasa Indonesia karena tingginya penggunaan prefiks walaupun dari segi kinerja temu-kembali tidak signifikan.

Sistem ini dapat dikembangkan lebih lanjut untuk menjadikannya sebuah IRS yang lengkap untuk Bahasa Indonesia. Beberapa alternatif pengembangan antara lain:

1. penggunaan kompresi untuk memperkecil ruang penyimpanan serta mempercepat proses pencarian teks.
2. penggunaan *thesaurus* untuk mengelompokkan istilah-istilah yang berhubungan.
3. penggunaan frase-frase yang diturunkan dari istilah-istilah yang muncul bersamaan dalam koleksi contoh.
4. penggunaan teknik *relevance feedback* untuk penyesuaian bobot istilah.
5. penggunaan koleksi yang lebih besar untuk lebih mendekati penggunaan sesungguhnya.

Penggunaan teknik kedua hingga keempat diperkirakan meningkatkan *precision* sebesar 10% - 20%, 5% - 10% dan 30% - 60% masing-masingnya [8].

REFERENSI

[1] Nua Internet Surveys, *Net users' patience only lasts 12 minutes,* http://www.nua.ie/surveys/index.cgi?f=VS&art_id=905356650&rel=true [7 Maret 2002]

[2] Baeza-Yates, R. and B. Ribeiro-Neto, *Modern Information Retrieval,* Addison-Wesley, 1999.

[3] Porter, M.F., An Algorithm for Suffix Stripping. *Program,* 14(3), 1980, 130-137.

[4] Akhmadi, C.H., *Algoritme Pemotong Sufiks Baku untuk Kata dalam Bahasa Indonesia Berbasis Algoritme Porter.* Bogor; Jurusan Ilmu Komputer IPB, 2002.

[5] Lancaster, F. and A. Warner, *Information Retrieval Today.* Arlington; Information Resources Press, 1993.

[6] Salton, G., *Automatic Text Analysis.* Technical Report No. 69-36, Department of Computer Science. Ithaca; Cornell University, 1969.

[7] Vorhees, E.M. and D. Harman, Overview of TREC 2001. *Proc. of the 10th Text REtrieval Conference,* 2001.

[8] Salton, G, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, 1989.