

Pemanfaatan Algoritma Porter Stemmer Untuk Bahasa Indonesia Dalam Proses Klasifikasi Jenis Buku

Bonifacius Vicky Indriyono¹, Ema Utami², Andi Sunyoto³

Program Pascasarjana Magister Teknik Informatika STMIK AMIKOM Yogyakarta

Jl. Ring Road Utara Condong Catur, Depok, Sleman, Yogyakarta

E-mail: ¹bonifaciusvicky@gmail.com, ²emma@nrar.net, ³andi@amikom.ac.id

Masuk: 3 Mei 2015; Direvisi: 19 Mei 2015; Diterima: 22 Mei 2015

Abstract. *Stemming is the process of mapping and decomposition of various forms (variants) of a word to essentially find the root word. This process is also referred to as the conflation. Stemming process has been widely used in the activities of the information retrieval (search information) to improve the quality of the information obtained. Stemming works by employing words taken from a dictionary and the usage of the basic rules of affixes. Porter stemmer for Indonesian or commonly referred as Tala stemmer uses the rules of basic analysis to find the root of a word. Tala Stemmer does not use a dictionary in the process. Instead, it uses a rule-based algorithm. In this study, the principal issue raised is how to make the process of classification/determination of the book/library materials in a library with a fast and effective manner in order to minimize error in determining the type of books. The solution is to utilize the method used by the porter stemmer for stemming Indonesian.*

Keywords: *Stemming, Information Retrieval, Porter Stemmer, Classification*

Abstrak. *Stemming adalah proses pemetaan dan penguraian berbagai bentuk (variants) dari suatu kata menjadi bentuk kata dasarnya. Proses ini juga disebut sebagai conflation. Proses stemming secara luas sudah digunakan di dalam kegiatan Information retrieval (pencarian informasi) untuk meningkatkan kualitas informasi yang didapatkan. Cara kerja stemming dapat dilakukan dengan menggunakan kamus kata dasar maupun menggunakan aturan-aturan imbuhan. Porter stemmer untuk Bahasa Indonesia atau yang biasa disebut dengan stemmer Tala menggunakan rule base analisis untuk mencari root sebuah kata. Stemmer Tala tidak menggunakan kamus dalam proses, melainkan menggunakan algoritma berbasis aturan. Dalam penelitian ini, pokok permasalahan yang diangkat adalah bagaimana melakukan proses klasifikasi/penentuan jenis buku/bahan pustaka dalam sebuah perpustakaan dengan cara yang cepat dan efektif sehingga dapat meminimalisir kesalahan penentuan jenis buku. Solusi yang dipergunakan adalah dengan memanfaatkan metode stemming dengan porter stemmer untuk bahasa Indonesia.*

Kata Kunci: *Stemming, Information Retrieval, Porter Stemmer, Klasifikasi*

1. Pendahuluan

Dewasa ini, buku merupakan salah satu sarana pemberi informasi bagi masyarakat luas. Ibarat kata, buku adalah jendela informasi dunia, maka segala kebutuhan akan informasi apapun tersaji dalam pustaka buku. Untuk mempermudah pencarian informasi yang sesuai dengan yang diinginkan oleh pencari informasi dan sesuai dengan waktunya, maka pengklasifikasiasian dokumen akan membantu bagaimana mendapatkan informasi, sehingga mempermudah pengolahan dan penggunaannya sesuai kebutuhan dan tujuan yang ingin dicapai. Klasifikasi adalah proses menemukan model atau fungsi yang menjelaskan atau membedakan konsep/kelas data, dengan tujuan dapat memperkirakan kelas dari suatu objek dan merupakan proses awal pengelompokan data. Proses klasifikasi dokumen merupakan proses yang sangat penting dalam bidang sistem informasi, khususnya dalam proses penambangan data (*data mining*) untuk memperoleh pengetahuan bisnis (Mulyanto, 2009). Menurut (Walpole dan Myers, 1995), klasifikasi adalah salah satu tugas yang penting dalam *data mining*. Permasalahan yang muncul

sekarang adalah bagaimana melakukan proses klasifikasi buku dengan cepat dan efektif agar buku dapat benar-benar berada pada jenis yang sebenarnya.

Permasalahan yang diuraikan di atas dapat diatasi dengan menggunakan metode dalam *text preprocessing* yang dilakukan sebelum tahap klasifikasi. Metode ini dikenal dengan istilah *stemming*. *Stemming* adalah proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya. Proses ini juga disebut sebagai *conflation* (Tala, 2003). Proses *stemming* secara luas sudah digunakan di dalam kegiatan *information retrieval* (pencarian informasi) untuk meningkatkan kualitas informasi yang didapatkan. Pada analisa temu kembali informasi imbuhan merupakan bagian dari informasi yang tidak bermakna, seperti halnya *stop word*. Sehingga imbuhan harus dihilangkan untuk mempercepat proses *index*. Cara kerja *stemming* dapat dilakukan dengan menggunakan kamus kata dasar maupun menggunakan aturan-aturan imbuhan. Dalam penelitian ini, dipergunakan metode *porter stemmer* atau yang dikenal dengan nama *stemmer* Tala. Metode ini tidak menggunakan kamus kata dasar tetapi menggunakan algoritma berbasis aturan dalam proses kerjanya.

Tujuan yang ingin dicapai dalam penelitian ini diantaranya: (1) Memahami aturan-aturan dalam *Porter Stemmer* untuk bahasa Indonesia. (2) Dapat menerapkan aturan *Porter Stemmer* untuk Bahasa Indonesia untuk proses klasifikasi jenis buku berbahasa Indonesia dalam sebuah *prototype*. Beberapa manfaat yang dapat diambil dari penelitian ini diantaranya: (1) Penelitian ini diharapkan dapat menambah wawasan dan pengetahuan peneliti tentang aturan-aturan dalam *porter stemmer* untuk Bahasa Indonesia. (2) Dapat membantu mempermudah dalam proses klasifikasi jenis buku dengan menerapkan aturan-aturan dalam *porter stemmer* untuk Bahasa Indonesia. (3) Dapat membantu mempercepat proses klasifikasi dengan terbentuknya kata dasar dari implementasi *porter stemmer* untuk Bahasa Indonesia.

2. Tinjauan Pustaka

2.1. Penelitian Terdahulu

Beberapa penelitian tentang penerapan metode *Naïve Bayes Classifier* yang sudah pernah dilakukan peneliti sebelumnya antara lain penelitian yang dilakukan oleh: (1) Indranandita, dkk. (2008) menjelaskan penggunaan *porter stemmer* untuk proses transformasi teks dalam sistem klasifikasi dan pencarian jurnal menggunakan metode *Naïve Bayes Classifier* dan *Vector Space Model*. (2) Agusta (2009) memaparkan pemanfaatan algoritma *porter stemmer* Bahasa Indonesia dalam proses klasifikasi dokumen dan membandingkannya dengan algoritma Nazief & Adriani. (3) Baskoro, dkk. (2012) menjelaskan tentang langkah-langkah/algoritma, aturan dan pemanfaatan *porter stemmer* dalam Bahasa Indonesia. (4) Budhi, dkk. (2012) menjelaskan cara pembuatan kata kunci secara otomatis dengan menggunakan GenEx dan memanfaatkan *porter stemmer for bahasa Indonesia* untuk penyesuaian kata kunci. Dari hasil penelitian ini, disimpulkan bahwa *porter stemmer for bahasa Indonesia* dapat digunakan untuk membantu dalam proses penyesuaian kata kunci. (5) Utomo (2013) memaparkan pemanfaatan *stemmer* Tala untuk temu kembali informasi yang diterapkan dalam sebuah aplikasi berbasis *web*. Hasil dari penelitian ini, bahwa *stemmer* Tala yang merupakan algoritma berbasis aturan memiliki performa yang stabil dengan jumlah dokumen yang berkembang. (6) Pradnyana dan Sanjaya (2012) menguraikan penggunaan algoritma *porter stemmer* Bahasa Indonesia pada proses penemuan kalimat-kalimat penting dari tiap dokumen dalam sistem *Automated Document Integration*. (7) Karaa (2013) memaparkan tentang pemanfaatan algoritma *porter stemmer* untuk proses *information retrieval* (temu kembali informasi).

2.2. Pengertian Klasifikasi

Mulyanto (2009) mengatakan proses klasifikasi adalah proses menemukan model atau fungsi yang menjelaskan atau membedakan konsep/kelas data, dengan tujuan dapat memperkirakan kelas dari suatu objek dan merupakan proses awal pengelompokan data. Proses klasifikasi dokumen merupakan proses yang sangat penting dalam bidang sistem informasi, khususnya dalam proses penambangan data (*data mining*) untuk memperoleh

pengetahuan bisnis (*business knowledge*). Klasifikasi adalah salah satu tugas yang penting dalam *data mining*.

Dalam klasifikasi, sebuah pengklasifikasi dibuat dari sekumpulan data latih dengan kelas yang telah ditentukan sebelumnya (Walpole dan Myers, 1995), sedangkan menurut Han dan Kamber (2001), proses klasifikasi dibagi menjadi dua tahap, yaitu tahapan *learning* dan *testing*. Pada tahap *learning*/pembelajaran, sebagian data yang telah diketahui kelasnya (data latih) digunakan untuk membuat model klasifikasi. Tahap *testing*/pengujian menguji data uji dengan model klasifikasi untuk mengetahui akurasi model klasifikasi tersebut. Menurut Tenenboim, dkk (2008), klasifikasi dokumen adalah bidang penelitian dalam perolehan informasi yang mengembangkan metode untuk menentukan atau mengkategorikan suatu dokumen ke dalam satu atau lebih kelompok yang telah dikenal sebelumnya secara otomatis berdasarkan isi dokumen.

2.3. Manfaat Klasifikasi

Menurut Eryono (1999), manfaat yang dapat diambil dari proses klasifikasi dokumen/bahan pustaka adalah sebagai berikut: (1) Buku/bahan pustaka yang sama atau mirip isinya akan terletak pada tempat yang berdekatan. (2) Memudahkan dalam mengadakan perimbangan koleksi yg dimiliki. (3) Memudahkan penelusuran terhadap bahan pustaka menurut subyek. (4) Memudahkan dalam membuat bibliografi menurut pokok masalah. Manfaat klasifikasi yang lainnya menurut Tenenboim (2008) adalah untuk mengelompokkan dokumen yang tidak terstruktur ke dalam kelompok-kelompok yang menggambarkan isi dari dokumen.

2.4. Pengertian Stemming

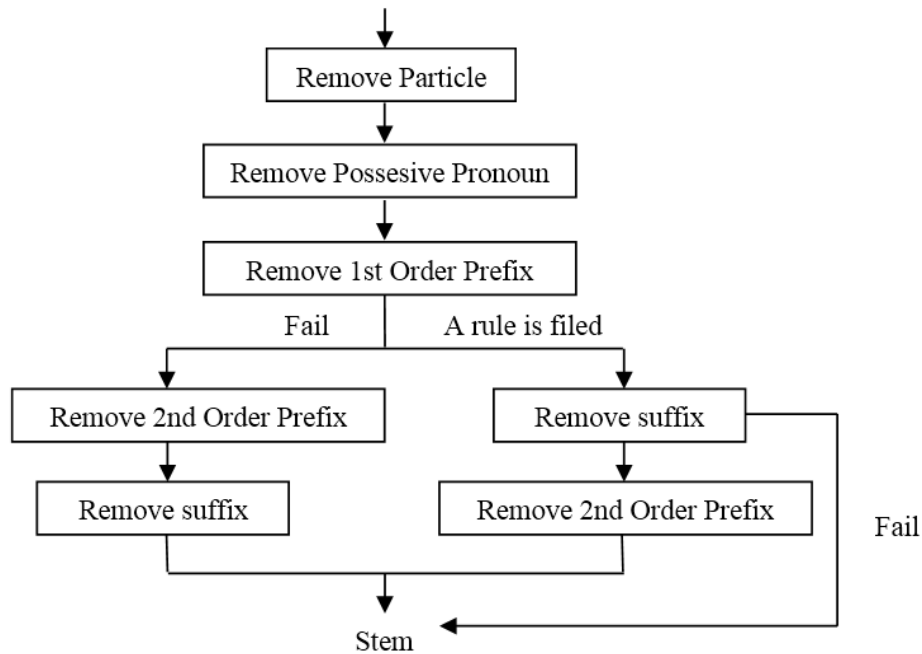
Stemming adalah proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya (Porter, 1980). Proses ini juga disebut sebagai *conflation*. Proses *stemming* secara luas sudah digunakan di dalam *information retrieval* (pencarian informasi) untuk meningkatkan kualitas informasi yang didapatkan (Frakes dan Baeza, 1992).

Stemming adalah suatu teknik pencarian bentuk dasar dari suatu *term*. Yang dimaksud dengan *term* itu sendiri adalah tiap kata yang berada pada suatu dokumen teks. *Stemming* dilakukan pada saat pembuatan indeks dari suatu dokumen. Pembuatan indeks dilakukan karena suatu dokumen tidak dapat dikenali langsung oleh suatu sistem temu kembali informasi atau *information retrieval (IR)* sistem. Oleh karena itu, dokumen tersebut terlebih dahulu perlu dipetakan ke dalam suatu representasi dengan menggunakan teks yang berada di dalamnya. Menurut Tala (2003), tujuan dari proses *stemming* adalah menghilangkan imbuhan-imbuhan baik itu berupa prefiks, sufiks, maupun konfiks yang ada pada setiap kata.

2.5. Porter Stemmer Untuk Bahasa Indonesia

Porter Stemmer untuk Bahasa Indonesia atau yang dikenal dengan nama *stemmer* Tala, dikembangkan oleh Tala (2003). Implementasi *Porter Stemmer* untuk Bahasa Indonesia berdasarkan *English Porter Stemmer* yang dikembangkan oleh Frakes dan Baeza (1992). Karena Bahasa Inggris datang dari kelas yang berbeda, beberapa modifikasi telah dilakukan untuk membuat Algoritma *Porter* dapat digunakan sesuai dengan Bahasa Indonesia (Tala, 2003). Desain dari *Porter Stemmer* untuk Bahasa Indonesia dapat dilihat pada Gambar 1.

Dari Gambar 1 dapat dijelaskan algoritma/langkah-langkah dalam *Porter Stemmer* untuk Bahasa Indonesia sebagai berikut: (1) Menghapus partikel seperti: -kah, -lah, -tah. (2) Menghapus kata ganti (*Possesive Pronoun*), seperti -ku, -mu, -nya. (3) Menghapus awalan pertama. Jika tidak ditemukan, maka lanjut ke langkah *remove 2nd order prefix*, dan jika ada maka lanjut ke langkah *remove suffix*. (4) (a) Menghapus awalan kedua, dan dilanjutkan pada langkah ke *remove suffix*. (b) Menghapus akhiran, jika tidak ditemukan maka kata tersebut diasumsikan sebagai kata dasar (*root word*). Jika ditemukan maka lanjut ke langkah *remove 2nd order prefix*. (5) (a) Menghapus akhiran dan kata akhir diasumsikan sebagai kata dasar (*root word*). (b) Menghapus awalan kedua dan kata akhir diasumsikan sebagai kata dasar (*root word*).

Gambar 1. Desain *Porter Stemmer* untuk Bahasa Indonesia (Tala, 2003)

2.6. Aturan *Porter Stemmer* Untuk Bahasa Indonesia

Terdapat lima aturan pada Algoritma *Porter Stemmer* untuk Bahasa Indonesia (Tala, 2003). Aturan-aturan tersebut dapat dilihat pada Tabel 1, Tabel 2, Tabel 3, Tabel 4, dan Tabel 5.

Tabel 1. Kelompok Rule Pertama: *Inflectional Particles*

<i>Suffix</i>	<i>Replacement</i>	<i>Additional Condition</i>	<i>Examples</i>
kah	NULL	NULL	Bukukah → buku
lah	NULL	NULL	Adalah → ada
pun	NULL	NULL	Bukupun → buku

Tabel 2. Kelompok Rule Kedua: *Inflectional Possesive Pronouns*

<i>Suffix</i>	<i>Replacement</i>	<i>Additional Condition</i>	<i>Examples</i>
ku	NULL	NULL	Bukuku → buku
mu	NULL	NULL	Bukumu → buku
nya	NULL	NULL	Bukunya → buku

Tabel 3. Kelompok Rule Ketiga: *First Order of Derivational Prefixes*

<i>Suffix</i>	<i>Replacement</i>	<i>Additional Condition</i>	<i>Examples</i>
meng	NULL	NULL	Mengukur → ukur
meny	S	V...	Menyapu → sapu
men	NULL	NULL	Menduga → duga; Menuduh → uduh
mem	P	V...	Memilah → pilah
mem	NULL	NULL	Membaca → baca
me	NULL	NULL	Merusak → rusak
peng	NULL	NULL	Pengukur → ukur
peny	S	V...	Penyapu → sapu
pen	NULL	NULL	Penduga → duga; Penuduh → uduh
pem	P	V...	Pemilah → pilah
pem	NULL	NULL	Pembaca → baca
di	NULL	NULL	Diukur → ukur
ter	NULL	NULL	Tersapu → sapu
ke	NULL	NULL	Kekasih → kasih

Tabel 4. Kelompok Rule Keempat: Second Order of Derivational Prefixes

Suffix	Replacement	Additional Condition	Examples
ber	NULL	NULL	Berlari → lari
bel	NULL	ajar	Belajar → ajar
be	NULL	K* er...	Bekerja → kerja
per	NULL	NULL	Perjelas → jelas
pel	NULL	ajar	Pelajar → ajar
pe	NULL	NULL	Pekerja → kerja

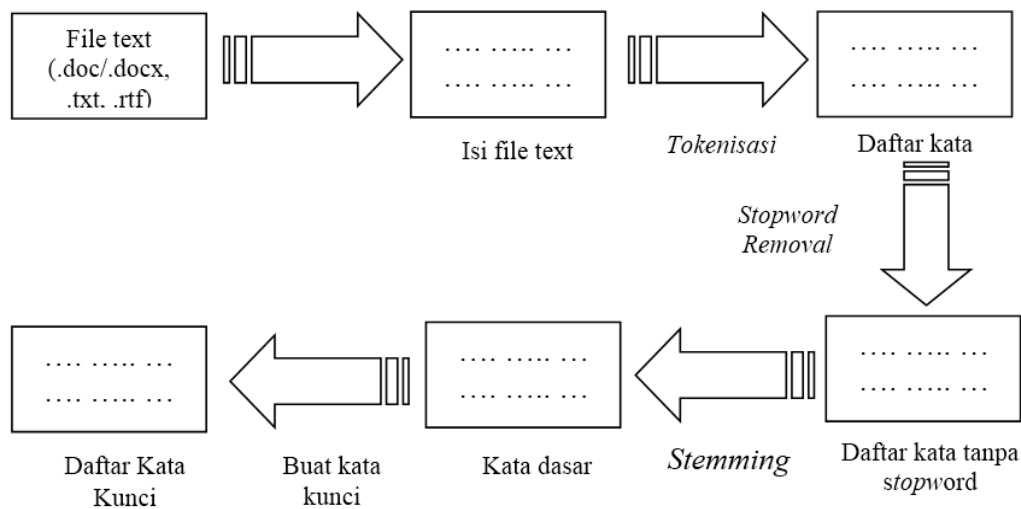
Tabel 5. Kelompok Rule Kelima: Derivational Suffixes

Suffix	Replacement	Additional Condition	Examples
kan	NULL	Prefix ∉ {ke, peng}	Tarikkan → tarik; (meng)ambilkan → ambil
an	NULL	Prefix ∉ {di, meng, ter}	Makanan → makan; (per)janjian → janji
i	NULL	V[K...c1c1, c1≠i, c2≠i and Prefix ∉ {ber, ke, peng}	Tandai → tanda; (men)dapati → dapat; Pantai → panta

3. Pembahasan dan Implementasi

3.1. Pembahasan

Sebelum dilakukan proses klasifikasi, data buku yang akan diproses untuk menentukan jenis bukunya, terlebih dahulu melalui tahap *text preprocessing*. Pada tahap *text preprocessing* dilakukan beberapa proses untuk menyiapkan judul dan sinopsis buku yang akan menjadi dokumen teks yang siap diolah pada tahap selanjutnya. Menurut Garcia (2005) pada tahap *text preprocessing* ini pada umumnya terdapat beberapa proses, antara lain *tokenizing*, *stopword removal*, *filtering*, *stemming* dan *term weighting*. Adapun tahap *text preprocessing* yang dilakukan dalam penelitian ini diperlihatkan pada Gambar 2.

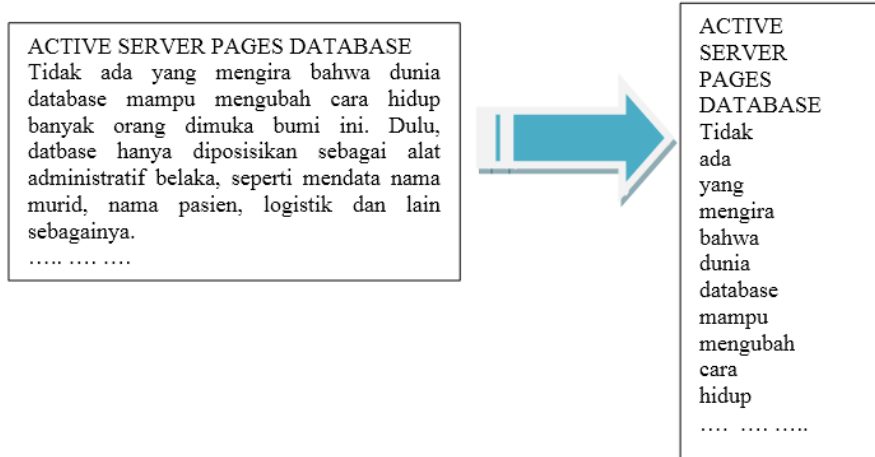
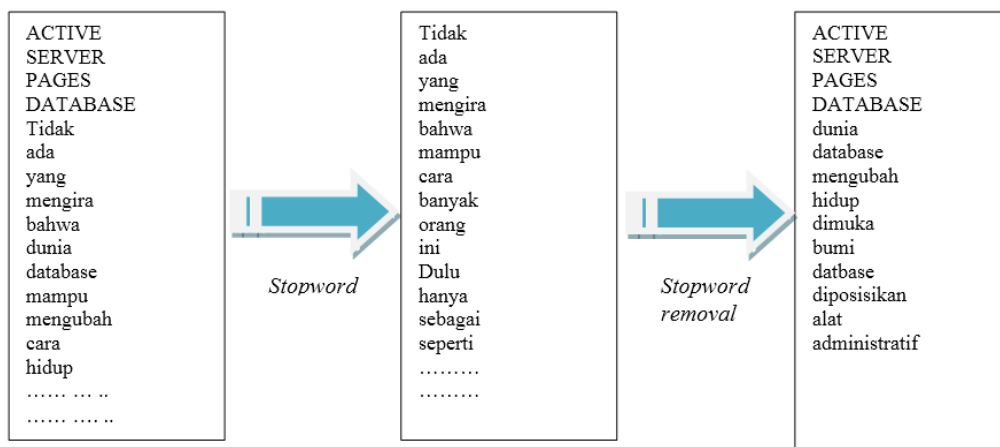
**Gambar 2. Bagan Tahap Text Preprocessing dalam Penelitian**

3.1.1. Tokenizing

Selama proses *tokenizing* berlangsung semua *string input* akan diuraikan sesuai dengan tiap kata yang menyusunnya. Setiap huruf *input* akan diubah menjadi huruf kecil. Semua tanda baca dan tanda hubung akan dihapuskan, termasuk semua karakter selain huruf *alphabet* (Garcia, 2005). Pada penelitian ini, tahap *tokenizing* akan melakukan pemisahan terhadap kata demi kata dari teks judul dan sinopsis buku yang dimasukkan. Tanda baca juga dihilangkan dalam proses ini, sehingga dokumen akan menghasilkan beberapa *token* yang akan diproses lebih lanjut. Contoh proses ini seperti diperlihatkan pada Gambar 3.

3.1.2. Stopword Removal

Pada bagian ini, sistem baru akan melakukan proses membuang kata-kata yang tidak bermakna dari daftar *token* yang dihasilkan pada proses *tokenizing*. Contoh proses *stopword removal* diperlihatkan pada Gambar 4.

Gambar 3. Contoh Tahap *Tokenizing*Gambar 4. Contoh Tahap *Stopword Removal*

3.1.3. Stemming

Pada bagian ini dilakukan proses menghilangkan imbuhan, awalan, akhiran dan kata depan pada *term* hasil dari proses *stopword removal*, sehingga *term* dapat menjadi bentuk kata dasarnya. Metode *stemming* yang digunakan adalah *Porter Stemmer* untuk Bahasa Indonesia atau yang lebih dikenal sebagai *stemming* Tala. Dengan ditemukannya kata dasar dari *term*, maka dapat ditemukan pula intensitas kemunculan *term* dalam tiap dokumen melalui proses *indexing*. Contoh *stemming* dapat dilihat pada Gambar 5.

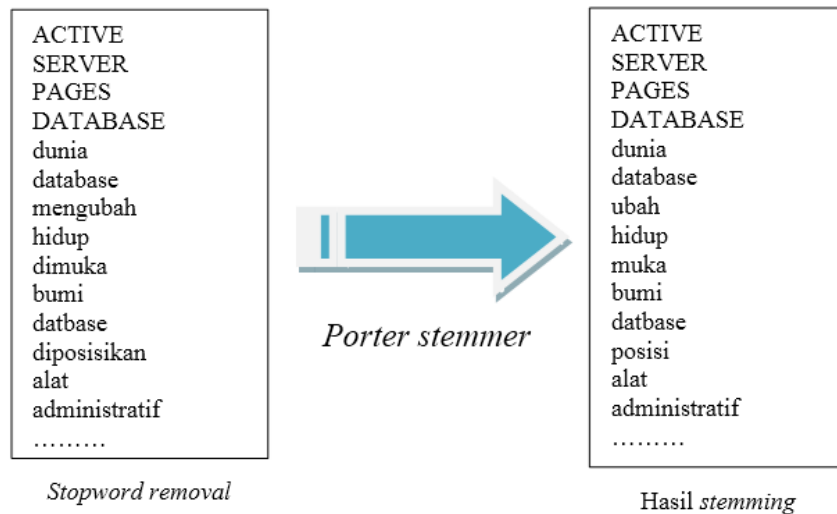
3.1.4. Indexing

Setelah tahap *stemming* dilakukan, maka data *term* yang dihasilkan dari *input* judul dan sinosis dari beberapa buku, diubah ke dalam *term frequency* (TF). TF adalah frekuensi kemunculan suatu *term* dalam tiap dokumen. Data TF tersebut kemudian disimpan untuk digunakan dalam proses klasifikasi. Pada proses klasifikasi ini, kata-kata yang telah dihasilkan dari proses *stemming* dan *indexing* akan digunakan untuk mencari probabilitas kesamaan kata dari dokumen buku yang akan diuji. Contoh dari TF dapat dilihat pada Gambar 6.

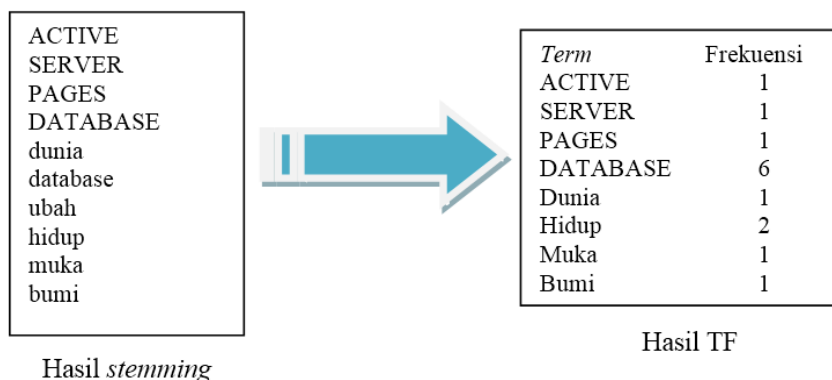
3.1.5 Proses Klasifikasi

Proses klasifikasi jenis buku mulai dilakukan setelah *term* yang dihasilkan dari data buku yang dijadikan sebagai dokumen *training* terbentuk. Pada bagian ini yang akan diuji adalah buku-buku yang dijadikan sebagai dokumen *testing*. Sebelum proses klasifikasi dilakukan, buku-buku *testing* harus melalui tahapan *text preprocessing* terlebih dahulu sampai pada tahap *stemming*. Dari tahap *stemming* ini dihasilkan kata-kata yang sudah tidak lagi memiliki imbuhan, awalan, akhiran dan kata depan

untuk kemudian dicocokkan dengan kata kunci yang sudah diberikan dan dihitung probabilitasnya dengan algoritma klasifikasi. Tabel 6 di bawah ini memperlihatkan contoh simulasi hasil proses klasifikasi dokumen *testing* yang cocok dengan tabel kata kunci yang ada.



Gambar 5. Contoh Tahap Stemming



Gambar 6. Contoh Tahap Indexing

Tabel 6. Simulasi Hasil Proses Klasifikasi Dokumen *Testing*

Judul Dokumen	Hasil	Kategori Akhir	Fitur (Kemunculan)
Membuat Database dengan Ms. Access	Akurat. Sesuai dengan klasifikasi	Database	Database(5), Access(2), Aplikasi(1)
Tuntunan Praktis: Belajar Database Menggunakan MySql	Akurat. Sesuai dengan klasifikasi	Database	Database(3), DBMS(1)
Aplikasi Excel Untuk Guru Edisi Revisi	Akurat. Sesuai dengan klasifikasi	Ms. Office	Excel(5), Aplikasi(1)
Basis Data	Akurat. Sesuai dengan klasifikasi	Database	Basis(8), Data(7)
Adobe in Design Creative Suite 2	Kurang akurat. Tidak sesuai dengan klasifikasi	Ms. Office	Adobe(4), word(1)
Buku Pintar Pemrograman Delphi untuk Pemula (Promo)	Akurat. Sesuai dengan klasifikasi	Delphi	Delphi(2), Program(1)
Dasar Pengolahan Citra dengan Delphi + CD	Akurat. Sesuai dengan klasifikasi	Delphi	Delphi(2), citra(2), aplikasi(1)
Asyiknya Mengenal JAVA	Akurat. Sesuai dengan klasifikasi	Java	Java(7), Program(4)
Buku Sakti Webmaster (Promo)	Akurat. Sesuai dengan klasifikasi	Pemrograman Web	Web(7), Database(3)
Ide Bisnis Bermodal Blog	Akurat. Sesuai dengan klasifikasi	Internet	Blog(9), Aplikasi(2), Program(3)
Belajar Desain Grafis untuk Pemula	Akurat. Sesuai dengan klasifikasi	Desain Grafis	CorelDraw(3), Photoshop(3)
SPSS 22 Pengolah Data Terpraktis	Akurat. Sesuai dengan klasifikasi	SPSS	SPSS(4), Data(3)

3.2. Implementasi

Pada bagian ini dilakukan proses perancangan antar muka untuk mengimplementasikan seluruh tahapan *text preprocessing*. Perancangan antar muka ini menggunakan *compiler* Delphi 2010 seperti yang diperlihatkan pada Gambar 7, Gambar 8, dan Gambar 9.

The screenshot shows the 'Data Dokumen Training' application. It has a main window with a 'Kategori' dropdown and a search bar. Below the search bar, there are two lists of documents. The first list, 'Kategori : DATABASE', shows documents DOC01 to DOC10. The second list, 'Kategori : MS. OFFICE', shows documents DOC11 to DOC18. To the right of the document list, there is a detailed view for document DOC04, titled 'Pemrograman database'. This view contains a description of the book and a table of tokens extracted from the document. The tokens are listed in a table with columns 'Kode Dokumen' and 'Kata/Token'.

Kode Dokumen	Kata/Token
DOC04	Pemrograman
DOC04	database
DOC04	Buku
DOC04	ini
DOC04	ditujukan
DOC04	untuk
DOC04	para
DOC04	programmer
DOC04	pemula
DOC04	yang
DOC04	ingin
DOC04	mengembangkan
DOC04	pengetahuannya

At the bottom of the application window, there are buttons for 'Proses Parsing' and 'Keluar'.

Gambar 7. Form Proses Tokenizing

The screenshot shows the 'Data dokumen training' application. It has a main window with a 'Kategori' dropdown and a search bar. Below the search bar, there is a table of documents. The table has columns 'Kode Dokumen', 'Judul', 'Sinopsis', and 'Pengarang'. The documents listed are DOC01, DOC02, and DOC04. Below the table, there are two sections: 'Hasil parsing judul & sinopsis' and 'Hasil stopword removal'. The 'Hasil parsing judul & sinopsis' section shows a list of tokens extracted from the documents. The 'Hasil stopword removal' section shows a list of tokens with stopwords removed. The tokens are listed in a table with columns 'Kata/Token' and 'Jml'.

Kata/Token	Jml
Pemrograman	1
database	1
Buku	1
ini	1
ditujukan	1
untuk	1
para	1
yang	1
ingin	1
dan	1
di	1
dalam	1
Jml:31	

The 'Hasil stopword removal' section shows a list of tokens with stopwords removed. The tokens are listed in a table with columns 'Kata/Token' and 'Jml'.

Kata/Token	Jml
Pemrograman	1
database	1
Buku	1
programmer	1
pemula	1
mengembangkan	1
pengetahuan	1
kemahirannya	1
program	1
database	1
Jml:30	

At the bottom of the application window, there are buttons for 'Stop Word Removal' and 'Keluar'.

Gambar 8. Form Proses Stopword Removal

Data Dokumen Training

Kategori : DATABASE

Kode Dokumen	Judul Buku
DOC01	ACTIVE SERVER PAGES DATABASE
DOC02	Panduan lengkap menguasai perintah
DOC04	Pemrograman database

Kode Dokumen: DOC01
 Judul Buku: ACTIVE SERVER PAGES DATABASE
 Pengarang: GREGORIUS AGUNG
 Sinopsis: Tidak ada yang mengira bahwa dunia database mampu mengubah cara hidup banyak orang dimuka bumi ini. Dulu, database hanya

Stop Word Removal

Drag a column header here to group by that column

Kata/Token
ACTIVE
SERVER
PAGES
DATABASE
dunia
database
mengubah
hidup
dimuka
bumi
database
diposisikan
alat
administratif
belaka
mendata
nama

Hasil Stemming

Drag a column header here to group by that column

Kode Dokumen	Kata	stem	Cacah
DOC01	PAGES	PAGES	1
DOC01	mampu	mampu	1
DOC01	banyak	banyak	1
DOC01	ini	ini	1
DOC01	diposisikan	posisi	1
DOC01	belaka	belaka	1
DOC01	murid	murid	1
DOC01	lain	lain	1
DOC01	berpikir	pikir	1
DOC01	kepentingan	nting	1
DOC01	ekonomi	ekonomi	1
DOC01	Sejak	Sejak	1
DOC01	bertambahnya	tambah	1
DOC01	upaya	upaya	1
DOC01	sekedar	sekedar	1
DOC01	interaktif	interaktif	1

Stemming Keluar

Gambar 9. Form Proses Stemming

4. Kesimpulan dan Saran

Berdasarkan hasil penelitian yang telah diuraikan pada bab sebelumnya dan perancangan *prototype* yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut: (1) Tahap *text preprocessing* yang dilakukan terhadap dokumen buku yang diuji semakin mempermudah untuk proses *stemming*. (2) Algoritma *stemming* dengan *porter stemmer* untuk Bahasa Indonesia yang menggunakan algoritma berbasis aturan sangat membantu mempercepat proses klasifikasi karena *word* yang dihasilkan dari dokumen-dokumen uji merupakan kata dasar yang semestinya sehingga memudahkan dalam pencocokan dengan kata kunci dan perhitungan probabilitas kemunculannya. (3) *Prototype* yang dibangun dengan penerapan aturan *porter stemmer* untuk Bahasa Indonesia ini dapat digunakan untuk membantu mempercepat dan mempermudah dalam proses klasifikasi jenis buku.

Penelitian ini menggunakan dokumen uji yang tidak terlalu banyak (900 buku) dan berdasarkan pada kriteria judul serta sinopsis buku, sehingga dapat diberikan saran bagi peneliti yang lain untuk menggunakan dokumen uji yang lebih banyak lagi dengan kriteria uji tidak hanya judul dan sinopsis agar hasil *stemming* dengan *porter stemmer* untuk Bahasa Indonesia lebih akurat.

Referensi

- Agusta, L. 2009. *Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia*. Konferensi Nasional Sistem dan Informatika, Bali, 14 November.
- Baskoro, D.O., Malik, H., & Anshari, M.H. 2012. *Porter Stemmer Information Retrieval*. Mini Paper Project. Yogyakarta: Computer Science, Gadjah Mada University.
- Budhi, G.S., Noertjahyana, A., & Susilo, R.Y. 2012. *Ekstraksi Kata Kunci Otomatis untuk Dokumen Berbahasa Indonesia menggunakan metode Genitor-plus-Extractor (GenEx)*. Seminar Nasional Teknologi Informasi dan Komputasi (SNASTIK), November, ISSN: 2302-7088.
- Eryono, K. 1999. *Penentuan Tajuk dan Bentuk Tajuk Nama Orang*. Jakarta: Universitas Terbuka.
- Frakes, W. B. & Baeza, R. 1992. *Information Retrieval Data Structure and Algorithms*. New Jersey: Prentice-Hall.
- Garcia, E. 2005. *Document Indexing Tutorial*, (Online), <http://www.miislita.com/information-retrieval-tutorial/indexing.html>, diakses pada tanggal 31 Oktober 2012.

- Han, J. & Kamber, M. 2001. *Data mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1-55860-489-8.
- Indranandita, A., Susanto, B., & Rachmat, A. 2008. Sistem Klasifikasi Dan Pencarian Jurnal Dengan Menggunakan Metode Naive Bayes Dan Vector Space Model. *Jurnal Informatika*, Vol. 4 No. 2, November.
- Karaa, W.B.A. 2013. A New Stemmer to Improve Information Retrieval. *International Journal of Network Security & Its Applications (IJNSA)*, Vol.5, No.4, July 2013.
- Mulyanto, A. 2009. *Sistem Informasi Konsep dan Aplikasi* (Cetakan I). Yogyakarta: Pustaka Pelajar.
- Porter, M. F. 1980. An Algorithm For Suffix Stripping. *Program*, 14(3), 130-137.
- Pradnyana, G.A. & Sanjaya, N.A. 2012. Perancangan Dan Implementasi Automated Document Integration Dengan Menggunakan Algoritma Complete Linkage Agglomerative Hierarchical Clustering. *Jurnal Ilmu Komputer*, 5(2).
- Tala, F.Z. 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Thesis. Institute for Logic Language and Computation Universiteit van Amsterdam TheNetherlands.
- Tenenboim, L., Shapira, B., & Shoval, P. 2008. *Ontology-Based Classification of News in an Electronic News Paper*. Paper presented at Intelligent Information and Engineering Systems Conference, Varna, Bulgaria.
- Utomo, M.S. 2013. Implementasi Stemmer Tala pada Aplikasi Berbasis Web. *Jurnal Teknologi Informasi DINAMIK Volume 18*, No.1, Januari 2013, ISSN: 0854-9524.
- Walpole, E. R. & Myers, R. H. 1995. *Ilmu Peluang dan Statistika untuk Insinyur dan Ilmuan* (Edisi ke-4). Bandung: ITB.