



TUGAS AKHIR - SM141501

# PENERAPAN ALGORITMA PENINGKATAN PORTER STEMMER DAN LIKELIHOOD UNTUK IDENTIFIKASI TOPIK PADA ARTIKEL BERITA BERBAHASA INDONESIA

DEVI ANDRIYANI  
NRP 1212 100 088

Dosen Pembimbing  
Dr. Imam Mukhlash, S.Si, MT  
Alvida Mustika Rukmi, S.Si, M.Si

JURUSAN MATEMATIKA  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016





FINAL PROJECT - SM141501

# APPLICATION OF ENHANCEMENT TO PORTER STEMMER ALGORITHM AND LIKELIHOOD TO IDENTIFY THE NEWS ARTICLE TOPIC

DEVI ANDRIYANI  
NRP 1212 100 088

Supervisors  
Dr. Imam Mukhlash, S.Si, MT  
Alvida Mustika Rukmi, S.Si, M.Si

DEPARTMENT OF MATHEMATICS  
Faculty of Mathematics and Natural Sciences  
Sepuluh Nopember Institute of Technology  
Surabaya 2016



## LEMBAR PENGESAHAN

**PENERAPAN ALGORITMA PENINGKATAN PORTER  
STEMMER DAN LIKELIHOOD UNTUK IDENTIFIKASI  
TOPIK PADA ARTIKEL BERITA BERBAHASA INDONESIA**

**APPLICATION OF ENHANCEMENT TO PORTER STEMMER  
ALGORITHM AND LIKELIHOOD TO IDENTIFY THE NEWS  
ARTICLE TOPIC**

### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
- Untuk Memperoleh Gelar Sarjana Sains  
Pada Bidang Studi Ilmu Komputer  
Program Studi S-1 Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

**DEVI ANDRIYANI**  
NRP. 1212 100 088

Menyetujui,

Dosen Pembimbing II,

Dosen Pembimbing I,

  
Alvida Mustika Rukmi, S.Si, M.Si  
NIP. 19720715 199802 2 001

  
Dr. Imam Mukhlash, S.Si, M.T  
NIP. 19700831 199403 1 003



Dr. Imam Mukhlash, S.Si, MT  
NIP. 19700831 199403 1 003  
Surabaya, Juli 2016



# **PENERAPAN ALGORITMA PENINGKATAN PORTER STEMMER DAN LIKELIHOOD UNTUK IDENTIFIKASI TOPIK PADA ARTIKEL BERITA BERBAHASA INDONESIA**

Nama Mahasiswa : Devi Andriyani  
NRP : 1212 100 088  
Jurusan : Matematika - ITS  
Dosen Pembimbing : Dr. Imam Mukhlash, S.Si, MT  
Alvida Mustika Rukmi, S.Si, M.Si

## **ABSTRAK**

*Setiap informasi yang disajikan dalam suatu berita memiliki tema pembicaraan yang beragam sehingga tidak mungkin semua informasi tersebut bisa dicerna secara bersamaan, melainkan harus dikelompokkan berdasarkan relevansi topik dari berita tersebut. Pengelompokan tersebut dapat mempermudah pembaca untuk memilih informasi yang penting sesuai dengan topik yang ingin dibaca. Berkaitan dengan pengelompokan berita, berita memiliki karakteristik yang berbeda dengan informasi yang lain sehingga diperlukan suatu algoritma khusus yang mampu menangani penemuan topik dan klasifikasi menggunakan data training pada suatu berita. Pada penelitian ini akan diterapkan algoritma peningkatan Porter Stemmer pada proses stemming (pembentukan kata dasar) dan metode Likelihood untuk klasifikasi berita berdasarkan kategori serta identifikasi topik.*

*Berdasarkan hasil pengujian menggunakan 900 data training dan 90 data uji didapatkan akurasi yang cukup tinggi, yaitu 95,56 % untuk klasifikasi kategori dan 97,78 % untuk identifikasi topik.*

**Kata kunci :** berita, algoritma Peningkatan Porter Stemmer, likelihood, klasifikasi kategori, identifikasi topik.





# APPLICATION OF ENHANCEMENT TO PORTER STEMMER ALGORITHM AND LIKELIHOOD TO IDENTIFY THE NEWS ARTICLE TOPIC

Student Name : Devi Andriyani  
NRP : 1212 100 088  
Major : Mathematics - ITS  
Supervisors : Dr. Iman Mukhlash, S.Si, MT  
Alvida Mustika Rukmi, S.Si, M.Si

## ABSTRACT

*Any information presented in the news discuss about diverse theme and that all of information can not be digested simultaneously, so that it must be grouped by relevance to the topic of the news. The grouping may be easier for the reader to choose the information that is important according to the topic you want to read. Relating to classification of the news, the news has different characteristics with other information so we need a special algorithm that is capable of handling the topic discovery and classification using training data. In this research will be applied Porter Stemmer algorithm improvement in the process of stemming (basic word formation) and Likelihood method for classifying news by category and topic identification.*

*Based on testing results using 900 training data and 90 test data obtained high accuracy, which is 95.56% for the classification category and 97.78% for the identification of topics.*

**Keywords:** *news, Improved Porter Stemmer algorithm, likelihood, classification categories, identification of topics*



## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan limpahan rahmat, petunjuk serta hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul

### **“PENERAPAN ALGORITMA PENINGKATAN PORTER STEMMER DAN LIKELIHOOD UNTUK IDENTIFIKASI TOPIK PADA ARTIKEL BERITA BERBAHASA INDONESIA”**

sebagai salah satu syarat kelulusan Program Sarjana Jurusan Matematika FMIPA Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Sholawat serta salam senantiasa penulis curahkan kepada junjungan Nabi besar Muhammad SAW, beserta para keluarga dan sahabatnya. Tugas akhir ini dapat terselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih dan penghargaan kepada:

1. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Ketua Jurusan Matematika ITS serta selaku dosen pembimbing atas segala bimbingan dan sarannya kepada penulis dalam mengerjakan tugas akhir ini sehingga dapat terselesaikan dengan baik.
2. Ibu Alvida Mustika Rukmi, S.Si, M.Si selaku dosen pembimbing II atas segala bimbingan dan motivasinya kepada penulis dalam mengerjakan tugas akhir ini sehingga dapat terselesaikan dengan baik.
3. Bapak Drs. Nurul Hidayat, M.Kom, Bapak Muhammad Syifa'ul Mufid, S.Si, M.Si dan Bapak Drs. Soetrisno, M.Komp selaku dosen penguji atas semua saran yang telah diberikan demi perbaikan tugas akhir ini.

4. Bapak Dr. Didik Khusnul Arif, M.Si. selaku Ketua Program Studi Sarjana Matematika ITS.
5. Bapak Iis Dr. Iis Herisman, M.Sc selaku Sekretaris Ketua Program Studi Sarjana Matematika ITS dan Mas Ali yang selalu memberikan informasi mengenai tugas akhir.
6. Ibu Dian Winda Setyawati, S.Si, M.Si selaku dosen wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Jurusan Matematika FMIPA ITS.
7. Bapak dan Ibu dosen serta para staf Jurusan Matematika ITS yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap semoga tugas akhir ini dapat bermanfaat bagi banyak pihak.

Surabaya, Juli 2016

Penulis

## DAFTAR ISI

Abstraksi.....	i
Abstract .....	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	xi
<b>BAB I PENDAHULUAN</b> .....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah.....	2
1.3    Batasan Masalah .....	3
1.4    Tujuan.....	3
1.5    Manfaat .....	3
1.6    Sistematika Pembahasan.....	4
<b>BAB II TINJAUAN PUSTAKA</b> .....	7
2.1 <i>Corpus</i> .....	7
2.2    Metode Peningkatan Porter Stemmer .....	8
2.3 <i>Vector Space Model</i> .....	10
2.4    Metode TD-IDF .....	10
2.5 <i>Likelihood</i> .....	11
2.6    Algoritma Identifikasi Topik .....	12
2.7    Metode Evaluasi Uji Coba .....	14
<b>BAB III METODE PENELITIAN</b> .....	13
3.1    Studi Literatur .....	15
3.2    Perancangan Perangkat Lunak.....	16
3.3    Pengumpulan Corpus .....	17
3.4 <i>Training</i> Teks Dokumen .....	17
3.5    Klasifikasi Kategori .....	18
3.6    Identifikasi Topik.....	18
3.7    Uji Coba dan Evaluasi .....	18
3.8    Penarikan Kesimpulan dan Penulisan Buku Tugas Akhir.....	18

<b>BAB IV PERANCANGAN SISTEM DAN IMPLEMENTASI</b>	19
4.1 Diagram Alir Sistem	19
4.1.1 <i>Case Folding</i>	20
4.1.2 <i>Filtering</i>	20
4.1.3 <i>Stoplist Removal</i>	20
4.1.4 Penerapan Algoritma Peningkatan Porter Stemmer pada Proses <i>Stemming</i>	20
4.1.5 <i>Weighting</i>	23
4.1.6 Ekstraksi Kata Kunci	23
4.1.7 Perhitungan Nilai <i>Likelihood</i> dan Nilai Ambang	26
4.1.8 Perhitungan CosSim Topik	28
4.1.9 Perhitungan Nilai Ambang CosSim	31
4.2 <i>Desain Physical Data Model</i>	32
4.3 <i>Use Case Diagram</i>	33
4.4 Diagram Rancangan <i>Interface</i>	33
4.5 Implementasi Proses <i>Case Folding</i>	34
4.6 Implementasi Proses <i>Filtering</i>	34
4.7 Implementasi Proses <i>Stoplist Removal</i>	34
4.8 Implementasi Proses <i>Stemming</i>	35
4.9 Implementasi Proses Ekstraksi <i>Keywords</i>	40
4.10 Implementasi Proses Klasifikasi Kategori	42
4.11 Implementasi Proses Identifikasi Topik	44
4.12 Implementasi <i>Interface</i>	46
<b>BAB V UJI COBA DAN PEMBAHASAN</b>	51
5.1 Data Uji Coba	51
5.2 Hasil Uji Coba	52
<b>BAB VI PENUTUP</b>	59
6.1 Kesimpulan	59
6.2 Saran	59
DAFTAR PUSTAKA	61
LAMPIRAN	63
BIODATA PENULIS	71

## DAFTAR GAMBAR

Gambar 3.1 Alur Metodologi Penelitian.....	15
Gambar 3.2 Format <i>Corpus</i> .....	16
Gambar 4.1 Diagram Alir Sistem.....	19
Gambar 4.2 Proses <i>Stemming</i> .....	22
Gambar 4.3 Ekstraksi Kata Kunci.....	25
Gambar 4.4 <i>Desain Physical Data Model</i> .....	32
Gambar 4.5 <i>Use Case Diagram</i> .....	33
Gambar 4.6 Diagram <i>Desain Interface</i> .....	34
Gambar 4.7 Segmen Program Proses <i>Stemming</i> .....	36
Gambar 4.8 Segmen Program Reduksi Awalan.....	38
Gambar 4.9 Source Code Proses Reduksi Akhiran.....	39
Gambar 4.10 Source Code Metode <i>getDS2()</i> .....	39
Gambar 4.11 Source Code Reduksi Partikel.....	39
Gambar 4.12 Source Code Kata Ganti Kepemilikan.....	40
Gambar 4.13 Segmen Program Ekstraksi Keyword.....	41
Gambar 4.14 Segmen Program Proses Parsing Data.....	42
Gambar 4.15 Segmen Program Perhitungan Likelihood.....	43
Gambar 4.16 Segmen Program Perhitungan Standar Deviasi dan Threshold.....	43
Gambar 4.17 Segmen Program Metode <i>pCocokan()</i> .....	44
Gambar 4.18 Segmen Program Perhitungan <i>CosSim</i> .....	45
Gambar 4.19 Segmen Program Metode <i>dotProduk()</i> .....	45
Gambar 4.20 Segmen Program <i>formatToVektor()</i> .....	46
Gambar 4.21 Tab Preprocessing.....	47
Gambar 4.22 Tab Ekstraksi Keyword.....	48
Gambar 4.23 Tab Klasifikasi Kategori.....	49
Gambar 4.24 Tab Identifikasi Topik.....	50

Gambar 4.25 Rata-Rata Nilai Akurasi Klasifikasi Kategori.....	55
Gambar 4.26 Rata-Rata Nilai Akurasi Identifikasi Topik.....	56



## DAFTAR TABEL

Tabel 4.1 Contoh Perhitungan $P(k_i c_j)$ .....	26
Tabel 4.2 Contoh Perhitungan Likelihood .....	27
Tabel 4.3 Contoh Perhitungan Threshold.....	28
Tabel 4.4 Contoh Perhitungan CosSim .....	30
Tabel 5.1 Spesifikasi Data Uji.....	51
Tabel 5.2 Evaluasi Kalsifikasi Kategori Menggunakan 5 Kata Kunci.....	52
Tabel 5.3 Evaluasi Kalsifikasi Kategori Menggunakan 10 Kata Kunci.....	52
Tabel 5.4 Evaluasi Kalsifikasi Kategori Menggunakan 15 Kata Kunci.....	53
Tabel 5.5 Evaluasi Kalsifikasi Kategori Menggunakan 20 Kata Kunci.....	53
Tabel 5.6 Evaluasi Kalsifikasi Kategori Menggunakan 25 Kata Kunci.....	54
Tabel 5.7 Rata-Rata Nilai <i>Accuracy</i> Kalsifikasi Kategori .....	54



## **BAB I**

### **PENDAHULUAN**

Pada bab ini, dijelaskan mengenai latar belakang penelitian Tugas Akhir serta rumusan masalah dan batasan masalah berdasarkan latar belakang tersebut. Selain itu, juga dijelaskan tujuan dan manfaat penelitian Tugas Akhir serta sistematika penulisan tugas akhir. Dari uraian tersebut, diharapkan gambaran umum permasalahan dan pemecahan yang diambil dalam tugas akhir ini dapat dipahami dengan baik.

#### **1.1 Latar Belakang**

Seiring dengan perkembangan teknologi yang pesat semakin meningkat pula penyebaran informasi secara online seperti halnya berita atau artikel yang mudah sekali kita jumpai pada berbagai situs. Sekumpulan informasi tersebut tentunya memiliki tema pembicaraan yang beragam sehingga tidak mungkin semua informasi yang disajikan bisa dicerna secara bersamaan, melainkan harus dikelompokkan berdasarkan relevansi topik dari berita tersebut. Pengelompokan tersebut dapat mempermudah pembaca untuk memilih informasi yang paling penting sesuai dengan topik yang ingin dibaca.

Informasi dalam berita mempunyai karakteristik yang berbeda dengan koleksi dokumen lainnya yaitu aliran dinamis berupa dokumen – dokumen baru yang mungkin saja memiliki informasi yang tidak pernah ada pada dokumen sebelumnya. Maka untuk melakukan klasifikasi topik dibutuhkan sebuah algoritma khusus yang mampu menangani penemuan topik, dan klasifikasi menggunakan data training[1]. Proses identifikasi topik berita nantinya akan dilakukan pra-proses yaitu terdiri dari proses *filtering*, *stopword removal*, *stemming* dan *weighting*.

Pada [1] telah dilakukan penelitian untuk identifikasi topik dan kategori berita berbahasa Inggris menggunakan perhitungan likelihood. Sedangkan pada [2] dilakukan penelitian sejenis yaitu identifikasi topik dan kategori terhadap berita Bahasa Indonesia

menggunakan perhitungan likelihood serta penggunaan algoritma Confix Stemmer untuk pembentukan kata dasar (stemming). Meskipun running time yang diperlukan untuk identifikasi topik cukup lama tapi nilai precision yang dihasilkan cukup tinggi yaitu 97,26 %.

Berdasarkan hal di atas, pada Tugas Akhir ini diajukan pembuatan aplikasi identifikasi topik berita Bahasa Indonesia yang pada prosesnya akan digunakan algoritma lain yang diketahui memiliki running time yang cukup baik [6]. Algoritma tersebut adalah algoritma peningkatan Porter Stemmer yang dimodifikasi oleh Putu Bagus Susastra Wiguna dan Bimo Sunarfri Hantono [5] sehingga algoritma ini memiliki performa yang lebih baik dalam hal akurasi untuk proses stemming. Dengan kemampuan performa yang baik maka metode ini akan diterapkan pada proses *stemming* identifikasi topik berita sebagai uji kinerja aplikasi yang nantinya akan dibandingkan dengan penelitian terdahulu. Dengan demikian, aplikasi ini diharapkan dapat menunjukkan kinerja yang lebih baik terutama dalam keakuratan pengidentifikasian topik berita sehingga keberhasilan kinerjanya dapat menjadi media penunjang dalam mempermudah pemilihan informasi berita berdasarkan topik bagi pengguna.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, masalah yang akan dibahas dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana merepresentasikan suatu berita / teks dokumen agar metode klasifikasi dokumen dapat diterapkan?
2. Bagaimana menerapkan algoritma peningkatan *Porter Stemmer* pada proses *stemming* dokumen?
3. Bagaimana menerapkan metode *likelihood* untuk membuat aplikasi yang dapat mengidentifikasi topik dokumen?

4. Bagaimana *performance* nilai akurasi hasil identifikasi topik (perbandingan dengan penelitian terdahulu)?

### 1.3 Batasan Masalah

Dalam penyusunan Tugas Akhir ini, terdapat beberapa acuan batasan permasalahan sebagai berikut:

1. Dokumen yang digunakan adalah berita Bahasa Indonesia
2. Dokumen berita untuk *training* dan *testing* menggunakan corpus yang diunduh dari [www.kompas.com](http://www.kompas.com) dan disimpan dalam bentuk *notepad* yang kemudian di-*convert* ke ekstensi *.news*
3. Jumlah dokumen berita dan jumlah kategori primitif yang digunakan sesuai dengan jumlah data yang ada di *database*.
4. Aplikasi dibuat menggunakan Bahasa pemrograman Java dan *database* MySQL

### 1.4 Tujuan

Tujuan dari Tugas Akhir yang ingin dicapai berdasarkan rumusan masalah yang telah diuraikan adalah sebagai berikut:

1. Membuat aplikasi yang dapat mengidentifikasi topik dari berita berbahasa Indonesia.
2. Mengetahui hasil penerapan algoritma peningkatan Porter Stemmer saat proses stemming dalam hal akurasi hasil identifikasi topik dokumen berita berbahasa Indonesia.

### 1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah :

1. Program ini diharapkan nantinya dapat menjadi media penunjang yang dapat mempermudah pengguna dalam memilih informasi dari dokumen berita sesuai topik yang

diinginkan, dengan kata lain sebagai alat yang secara otomatis dapat mengidentifikasi topik berita tanpa harus melalui proses manual seperti pada umumnya.

2. Mendapatkan efisiensi waktu dalam pencarian topik pada berita yang ingin dibaca.
3. Memperluas wawasan soal kinerja metode yang paling baik untuk diterapkan.

## **1.6 Sistematika Pembahasan**

Pembahasan buku Tugas Akhir ini dibagi menjadi beberapa bab sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini berisi tentang latar belakang, permasalahan, batasan permasalahan, tujuan, manfaat, serta sistematika pembahasan buku Tugas Akhir.

### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi kajian teoritis atas beberapa metode dan algoritma yang digunakan di dalam penyusunan Tugas Akhir ini. Secara garis besar, bab ini berisi tentang *Corpus*, Metode Peningkatan Porter Stemmer, *Vector Space Model*, Metode TF-IDF, *Likelihood*, serta Algoritma Identifikasi Topik.

### **BAB III METODE PENELITIAN**

Bab ini membahas tentang tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini.

### **BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM**

Bab ini berisi tentang perancangan sistem serta implementasinya yang terdiri dari perancangan perangkat lunak serta implementasi proses *case folding*,

*filtering, stoplist removal, stemming*, klasifikasi kategori, identifikasi topik dan metode lainnya.

## BAB V UJI COBA DAN PEMBAHASAN

Bab ini membahas tentang uji coba dan evaluasi hasil uji coba serta penggunaan jumlah kata kunci yang memberikan nilai tertinggi.

## BAB VI PENUTUP

Bab ini berisi kesimpulan dan saran berdasarkan pembahasan dari seluruh penelitian tugas akhir.





## BAB II

### TINJAUAN PUSTAKA

Pada bab ini, diuraikan beberapa hal yang mendukung penyelesaian Tugas Akhir. Beberapa hal tersebut meliputi *Corpus*, metode peningkatan *porter stemmer*, *vector space model*, metode TF-IDF, Likelihood, dan algoritma identifikasi topik.

#### 2.1 Corpus

Corpus merupakan sekumpulan teks terstruktur. Secara lebih spesifik merupakan teks berita hasil pengunduhan dari situs berita *online* yang disimpan dalam format file teks tertentu dan memiliki keterkaitan kategori antar *corpus*. *Corpus* yang akan digunakan pada tugas akhir ini adalah *corpus* dengan ekstensi *.news* supaya memudahkan program mengenali corpus saat akan diproses[2]. Bila *corpus* tetap disimpan dalam ekstensi *.txt*, maka terdapat kemungkinan besar dalam sebuah *folder* terdapat *file – file* lain yang tidak berhubungan yang menggunakan ekstensi yang sama, sehingga akan ikut terproses dan mengganggu jalannya proses aplikasi.

Secara sederhana, *corpus* secara keseluruhan adalah hasil pengunduhan berita pada situs dengan menghilangkan atribut – atribut html maupun php pada halaman tersebut yang sudah terstruktur mengikuti format yang telah dijelaskan pada Gambar 3.2.

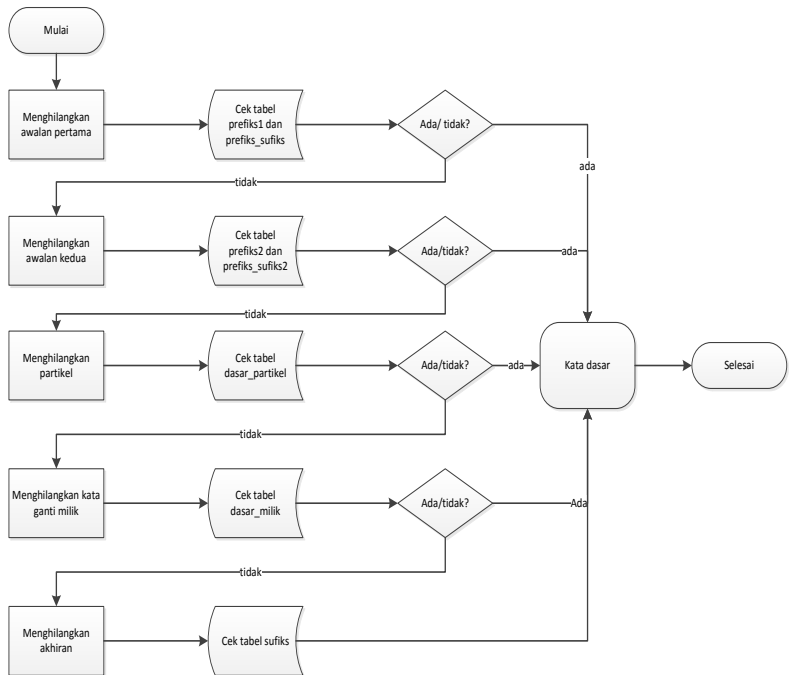
Pada *corpus* data training tidak dituliskan kategori yang telah ditetapkan oleh situs kompas karena terdapat beberapa kategori yang namanya diubah menjadi nama yang lebih umum sehingga berbeda dengan nama kategori yang dituliskan pada [www.kompas.com](http://www.kompas.com). Seluruh berita disimpan dalam folder menurut nama kategorinya. Perbedaan signifikan yang terdapat antara data training dengan data uji adalah pada atribut topik. Data uji yang mengalami pembentukan *corpus* juga, tidak memiliki atribut topik karena diasumsikan atribut tersebut akan menjadi hasil identifikasi pada proses identifikasi topik yang dilakukan program.

## 2.2 Metode Peningkatan *Porter Stemmer*

Tidak semua proses pembentukan kata dari kata dasar bisa diselesaikan dengan satu tingkat morfologi. Contoh pembentukan kata dengan penambahan imbuhan pada kata dasar dengan satu tingkat morfologi adalah “mem”+”baca” menjadi “membaca”, “men”+”cari” menjadi “mencari”.

Penambahan imbuhan pada kata dasar untuk membentuk kata baru dengan mengubah fonem dari kata dasar tidak bisa diselesaikan dengan satu tingkat morfologi. Contoh kata yang tidak bisa diselesaikan dengan satu tingkat morfologi adalah kata “memutar” berasal dari kata dasar “putar” yang mendapat imbuhan “men-”. Untuk menyelesaikan masalah ini maka diperlukan 2 tingkat morfologi untuk menyelesaikan masalah ini.

Secara umum proses stemming dibagi menjadi 5 bagian yaitu: menghilangkan awalan pertama (“meng-”, “peng-”, “mem-”, “pem-”, “meny-”, “peny-”, “men-”, “pen-” dan lainlain), menghilangkan awalan kedua ( “ber-”, “per-”, “ter-”, “se-”, “pel-”, dan lain-lain), menghilangkan partikel (“-kah”, “-lah”, “-tah”, “-tah”), menghilangkan kata ganti milik (“- ku”, “-mu”, “-nya”), menghilangkan akhiran (“-kan”, “-an”, “-i”, “-isme”, “-isasi”, “-onal”). Proses stemming secara umum yang dilakukan pada penelitian ini dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Langkah-langkah Algoritma Peningkatan Porter [5]

Untuk menunjang proses *stemming* dapat dilakukan dengan baik maka diperlukan *database* kata yang terdiri dari 7 tabel pada *database* yang menjadi kamus kata-kata pengecualian untuk tiap prosesnya. 7 tabel pada *database* yang digunakan adalah tabel *dsr\_milik*, tabel *dsr\_partikel*, tabel *dsr\_prefiks1*, tabel *dsr\_prefiks1\_sufiks1*, tabel *dsr\_prefiks2*, tabel *dsr\_prefiks21*, tabel *dsr\_sufiks*. Contoh Tabel *dsr\_prefiks1* digunakan untuk menyimpan kata dasar yang memiliki fonem awalan pertama. Tabel ini berisi fonem awalan pertama pada kata tersebut tidak dihilangkan karena

merupakan bagian dari kata dasar. Tabel kedua adalah tabel `dsr_prefiks1_sufiks1`. Tabel ini digunakan untuk menyimpan kata-kata yang harus diproses 2 tingkat morfologi untuk kata dasar yg berawalan huruf “k” dan “p”. Begitu seterusnya untuk tabel yang lain.

### 2.3 *Vector Space Model*

Salah satu model matematika yang digunakan pada sistem temu-kembali informasi untuk menentukan bahwa sebuah dokumen itu relevan terhadap sebuah informasi adalah Vector Space Model (VSM). Model ini menghitung derajat kesamaan antara setiap dokumen yang disimpan di dalam sistem dengan *query* yang diberikan oleh pengguna. Model ini pertama kali diperkenalkan oleh Salton (1989)[7].

*Vector space model* merupakan salah satu pendekatan yang paling umum digunakan untuk merepresentasikan model teks digital. Setiap dokumen  $d_j$  akan direpresentasikan menjadi vektor [4].

$$d_j = (w_{1j}, w_{2j}, \dots, w_{ij}, \dots, w_{nj}) \quad (1)$$

dimana  $w_{ij}$  adalah bobot *term* ke-  $i$  pada dokumen  $j$  yang bersangkutan dan  $n$  adalah banyaknya *term* pada dokumen  $d_j$ .

### 2.4 *Metode TF-IDF*

Baeza-Yates dan Ribeiro-Neto (1999), menyebutkan bahwa pembobotan (tf-idf) terdiri dari dua faktor, yaitu:

#### 1. *tf (term frequency)*

*tf* adalah frekuensi kemunculan suatu istilah  $k_i$  di dalam sebuah dokumen  $d_j$  dibandingkan dengan frekuensi istilah  $k_i$  yang sering muncul pada dokumen itu. Jika dimasukkan dalam rumus matematika didapatkan:

$$tf_{ij} = \frac{freq_{ij}}{max_l freq_{lj}} \quad (2)$$

## 2. *idf* (inverse document frequency)

*idf* adalah frekuensi kemunculan suatu istilah  $k_i$  di dalam seluruh dokumen. Penggunaan faktor *idf* didasarkan pada istilah yang muncul pada setiap dokumen tidak memberikan suatu ciri khusus untuk menentukan dokumen yang relevan dari yang tidak relevan. Jika jumlah seluruh dokumen didalam sistem dinyatakan dengan nilai  $N$  dan jumlah dokumen yang memiliki istilah  $k_i$  tersebut dinyatakan dengan  $df_i$ , maka nilai *idf*-nya dapat dinyatakan dengan:

$$idf_i = \log_2 \left( \frac{N}{df_i} \right) \quad (3)$$

Bobot setiap *term* dapat direpresentasikan dengan frekuensi *invers* dokumennya (TF-IDF) yang dinyatakan sebagai berikut:

$$w_{ij} = tf_{ij} \log_2 \left( \frac{N}{df_i} \right) \quad (4)$$

dimana  $w_{ij}$  adalah bobot *term* ke  $i$  pada dokumen ke  $j$  yang bersangkutan,  $tf_{ij}$  adalah frekuensi term ke  $i$  pada dokumen ke  $j$ .  $N$  adalah jumlah dokumen yang diproses dan  $df_i$  adalah jumlah dokumen yang memiliki *term* ke  $i$  di dalamnya [7].

## 2.5 Likelihood

Perhitungan *likelihood* untuk sebuah kategori dijelaskan pada persamaan (5).

$$\text{Likelihood}(c_j | A = \{k_1, k_2, \dots, k_n\}) = - \sum_{i=1}^n P(k_i | c_j) \log(P(k_i | c_j)) \quad (5)$$

dimana  $c_j$  adalah kategori ke  $j$ ,  $A$  adalah artikel dokumen uji,  $P(k_i | c_j)$  dihitung menggunakan “In-Document” dan perhitungan “jumlah total dokumen”.

Setelah semua kategori dihitung nilai *likelihood*-nya maka nilai ambang batas dapat diperoleh. Nilai ambang (*threshold*) digunakan untuk menentukan apakah sebuah kategori dapat ditetapkan untuk artikel uji atau tidak. Nilai *threshold* dihitung menggunakan persamaan (6).

$$Threshold = \frac{\sum_1^{|L|} l_i}{|L|} + \sqrt{\frac{\sum \left( l_i - \frac{\sum_1^{|L|} l_i}{|L|} \right)^2}{|L|}} \quad (6)$$

$L$  adalah jumlah banyaknya *likelihood*, sementara  $l_i$  adalah *likelihood* untuk kategori ke  $-i$ . Asumsinya adalah kategori  $-$  kategori yang tepat akan memiliki nilai yang besarnya jauh berbeda dibandingkan kategori  $-$  kategori lainnya [1].

## 2.6 Algoritma Identifikasi Topik

Algoritma identifikasi topik dapat dibagi menjadi dua proses, yaitu klasifikasi dan *dynamic thresholding*. Algoritma ini menghitung kemiripan antara kata kunci topik awal yang telah diketahui sebelumnya dan kata kunci artikel uji. Setelah itu, nilai yang memiliki *similarity* paling tinggi ditetapkan untuk artikel sebagai *conditionally assigned topic* [1].

Untuk membandingkan antara vektor kata kunci dengan vektor topik, keduanya ditransformasikan ke dalam *vector-space* yang sama.

Topic:	war	iraq	US	UK		war	iraq	US	UK	violence
	2	5	4	1	⇒	2	5	4	1	0
Article:	war	iraq	violence			war	iraq	US	UK	violence
	1	3	1		⇒	1	3	0	0	1

**Gambar 2.2.** Contoh Transformasi Vektor [1]

Setelah itu dihitung nilai *similarity* menggunakan rumus berikut :

$$CosSim(t_i, A) = \frac{t_i A}{|t_i| |A|} \quad (7)$$

dengan  $t_i$  adalah vektor topik ke- $i$ ,  $A$  adalah artikel uji  $A$ ,  $|t_i|$  dan  $|A|$  masing-masing adalah panjang vektor topik ke  $-i$  dan panjang vektor artikel  $A$ .

Topik yang memiliki *similarity* terbesar nantinya akan diuji menggunakan nilai threshold dinamis (*dynamic threshold*). Nilai ambang ini akan membandingkan antara nilai topik awal yang ditentukan dengan nilai topik baru yang mungkin terbentuk *NewTSim* [1].

$$NewTSim(t_c, A) = \frac{(0.05 \times |t_c| \times (\text{mean}(A) - \text{StdDev}(A)) \times \text{mean}(t_c))}{(|A| \times (\text{mean}(A))^2) \times (|t_c| \times (\text{mean}(t_c))^2)} \quad (8)$$

dengan  $t_c$  merupakan topik awal yang telah ditentukan, yaitu hasil perhitungan *CosSim* terbesar,  $\text{Mean}(A)$  adalah rata-rata dokumen  $A$ ,  $\text{StdDev}(A)$  adalah standar deviasi vektor dokumen  $A$ , dan  $\text{mean}(t_c)$  adalah rata-rata topik awal yang telah ditentukan.

Langkah selanjutnya adalah *dynamic thresholding*, yaitu membandingkan nilai *NewSTim* dengan nilai topik awal yang telah ditentukan sebagai berikut (9) :

(i)  $\text{CosSim}(t_c, A) > 0.1 \wedge \text{CosSim}(t_c, A) > \text{NewTSim}(t_c, A)$

(ii)  $\text{NumTopics} > 10 \wedge \text{CosSim}(t_c, A) >$

$$(2 \times \text{StdDev}(\text{AllTopicSims}) + \text{Mean}(\text{AllTopicSims}))$$

dengan  $\text{CosSim}(t_c, A)$  adalah hasil perhitungan *Cosine Similarity* terbesar yang diperoleh dari persamaan (7) dan diasumsikan sebagai topik awal yang ditentukan.  $\text{NumTopic}$  adalah jumlah keseluruhan topik yang telah diketahui sebelumnya,  $\text{StdDev}(\text{AllTopicSims})$  dan  $\text{Mean}(\text{AllTopicSims})$  masing-masing adalah standar deviasi dan rata-rata seluruh *similarity* topik [1].

Pertidaksamaan 9(ii) berguna jika jumlah topik yang telah diketahui sebelumnya telah mencukupi. Berdasarkan hasil eksperimen pada [1], jumlah topik yang harus dipenuhi adalah sepuluh.

## 2.7 Metode Evaluasi Uji Coba

Pelaksanaan evaluasi uji coba seringkali menggunakan rumus *precision*, *recall*, *F-Measure* dan *accuration*. Adapun pengertian dari beberapa metode di atas adalah :

- *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem yang dirumuskan sebagai berikut :

$$Precision (P) = TP / (TP + FP)$$

- *Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi yang dirumuskan sebagai berikut :

$$Recall (R) = TP / (TP + FN)$$

- *F-Measure* adalah *harmonic mean* dari *precision* dan *recall* yang dirumuskan sebagai berikut :

$$F-Measure (F) = 2 * P * R / (P + R)$$

- *Accuracy* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual yang dirumuskan sebagai berikut :

$$Accuracy (A) = (TP + TN) / (TP + FP + FN + TN)$$

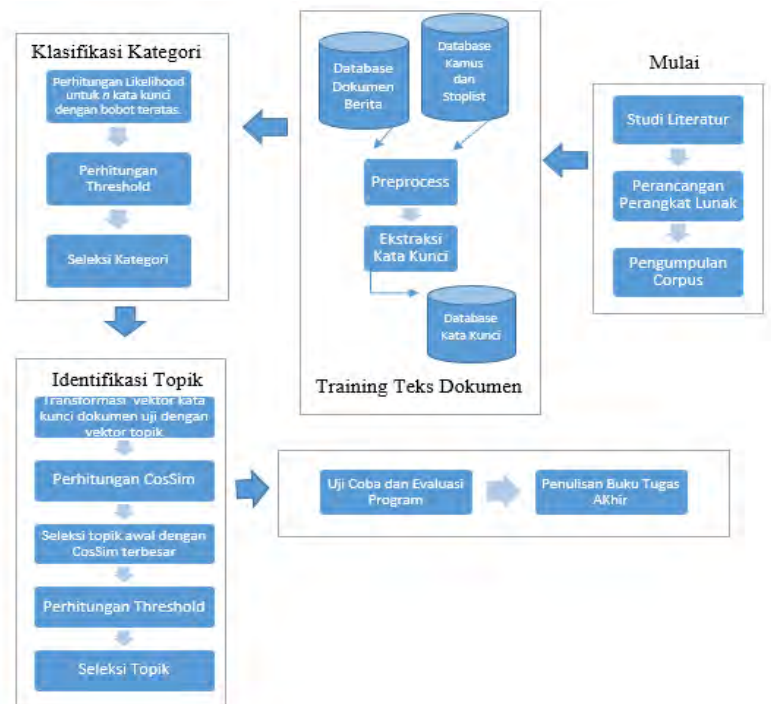
Tabel 4.1 Item Penyusun *Precision*, *Recall*, *F-measure*, *Accuracy* [8]

		Nilai Sebenarnya	
		<i>TRUE</i>	<i>FALSE</i>
Nilai prediksi	<i>TRUE</i>	TP ( <i>True Positive</i> ) <i>Corect result</i>	FP ( <i>False Positive</i> ) <i>Unexpected result</i>
	<i>FALSE</i>	FN ( <i>False Negative</i> ) <i>Missing result</i>	TN ( <i>True Negative</i> ) <i>Corect absence of result</i>



## BAB III METODE PENELITIAN

Bab ini membahas mengenai metodologi penelitian yang digunakan dalam menyelesaikan permasalahan pada Tugas Akhir ini, yang disusun secara sistematis sebagai berikut:



**Gambar 3.1** Alur Metodologi Penelitian

### 3.1 Studi Literatur

Pada tahap pertama ini dilakukan identifikasi masalah dan akan dilakukan pengkajian tentang *preprocess* identifikasi topik yang meliputi pencarian dan pemahaman informasi soal

representasi teks dokumen, *stopword elimination*, *stemming*, ekstraksi kata kunci, metode evaluasi, serta penerapan metode klasifikasi dan identifikasi topik pada dokumen berita.

### 3.2 Perancangan Perangkat Lunak (PL)

Perancangan PL ini terdiri dari diagram alir sistem, *use case diagram*, rancangan *database*, rancangan *interface* yang dapat dilihat pada Gambar 4.1 dan Gambar 4.4-4.6.

### 3.3 Pengumpulan Corpus

Data *input* aplikasi ini berupa *corpus* dokumen berita berbahasa Inggris dengan ekstensi *.news*. Ekstensi *.news* digunakan untuk mempermudah pengambilan file baik saat preproses ataupun proses pembelajaran aplikasi dilakukan. *Corpus* mempunyai format tanggal, kode sumber, judul dan isi dokumen berita. *Corpus* akan diambil melalui situs *www.kompas.com*. *Corpus* yang digunakan dalam Tugas Akhir ini terdiri dari 900 data training dan 10 data testing untuk masing-masing kategori.

Tanggal_berita	<Day, DD Month YYYY>
Topik_berita	<Topik Berita>
ID_Sumber	<ID Sumber Berita>
Judul_berita	<Judul Berita>
Isi_berita	<Isi berita>

**Gambar 3.2** Format *Corpus* [2]

### 3.4 Training Teks Dokumen

Dalam tahap ini akan dilakukan *Preprocess*, yaitu tahap perancangan fungsi-fungsi yang dapat diterapkan dalam aplikasi. Diantaranya adalah dokumen *training* harus

direpresentasikan dalam bentuk vektor yang meliputi *Case folding*, *Filtering*, *Stoplist Removal*, *Stemming*, dan *Weighting* [2].

### 3.5 Klasifikasi Kategori

Sebelum dilakukan klasifikasi kategori, akan digunakan kategori primitif yaitu kategori yang telah ditentukan sebelumnya berdasarkan pengamatan dari situs berita *online* [www.kompas.com](http://www.kompas.com). Terdapat sembilan kategori primitif yang digunakan yaitu Nasional, Regional, Internasional, Bisnis dan Ekonomi, Olahraga, Sains dan Teknologi, Metropolitan, Edukasi, Pariwisata. Pada proses klasifikasi kategori akan dilakukan pemilihan  $n$  kata kunci dengan bobot tertinggi yang kemudian dihitung nilai *likelihood* dari kata kunci tersebut. Selanjutnya dipilih kata kunci dengan nilai *likelihood* terbesar dan diuji menggunakan nilai *threshold*. Jika nilai *likelihood* lebih besar dari *threshold* maka kata kunci tersebut dikatakan memenuhi untuk dijadikan kategori dokumen.

### 3.6 Identifikasi Topik

Tahap selanjutnya adalah identifikasi topik dokumen. Pada tahap ini akan dilakukan transformasi vector kata kunci dokumen dengan kata kunci topik. Setelah itu dilanjutkan dengan perhitungan *similarity* dari kata kunci dokumen dengan kata kunci topik yang kemudian dipilih kata kunci dengan *similarity* tertinggi dan diuji menggunakan *dynamic thresholding*. Jika memenuhi nilai *dynamic thresholding* maka kata kunci tersebut ditetapkan sebagai topik dokumen.

### 3.7 Uji Coba dan Evaluasi

Pada tahap ini akan dilakukan uji coba pada program, yaitu menguji data *testing* Corpus yang sudah disimpan sebelumnya dengan jumlah 10 data untuk masing-masing kategori. Hasil dari

klasifikasi setiap kategori akan dihitung nilai *precision*, *recall*, *f-measure*, *accuracy* berdasarkan rumus pada bagian 2.7. Sedangkan untuk identifikasi topik menggunakan 90 data testing dan akan dihitung nilai *accuracy* berdasarkan jumlah teridentifikasi benar dibagi dengan total data uji.

### **3.8 Penarikan Kesimpulan dan Penulisan Buku Tugas Akhir**

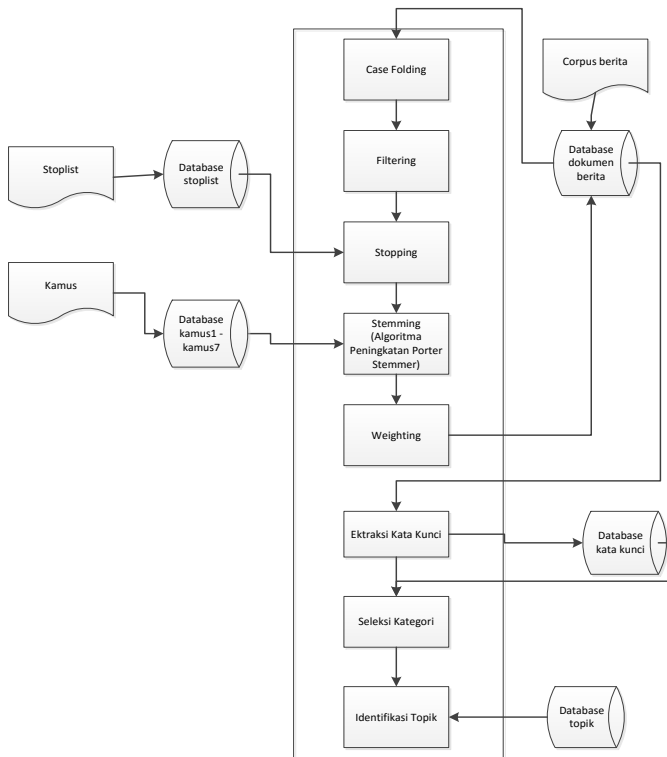
Tahap terakhir adalah membuat kesimpulan berdasarkan tujuan dan hasil uji coba, serta menyimpulkan jumlah kata kunci yang tepat dalam memberikan hasil yang maksimal untuk identifikasi topik. Kemudian dilanjutkan dengan penulisan buku Tugas Akhir sesuai format penulisan yang sudah diberikan sebelumnya.

## BAB IV

### PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini membahas tentang perancangan sistem dan implementasi dari hasil perancangan. Perancangan sistem Tugas Akhir ini terdiri diagram alir sistem, *use case diagram*, perancangan *database*, serta diagram rancangan *interface*. Kemudian dilanjutkan dengan pembahasan tentang implementasi yang terdiri dari implementasi program dan *interface*.

#### 4.1 Diagram Alir Sistem



**Gambar 4.1** Diagram Alir Sistem

#### 4.1.1 *Case Folding*

*Case folding* adalah proses seluruh huruf pada setiap kata dalam dokumen diubah menjadi huruf kecil

#### 4.1.2 *Filtering*

*Filtering* adalah eliminasi tanda baca

#### 4.1.3 *Stoplist Removal*

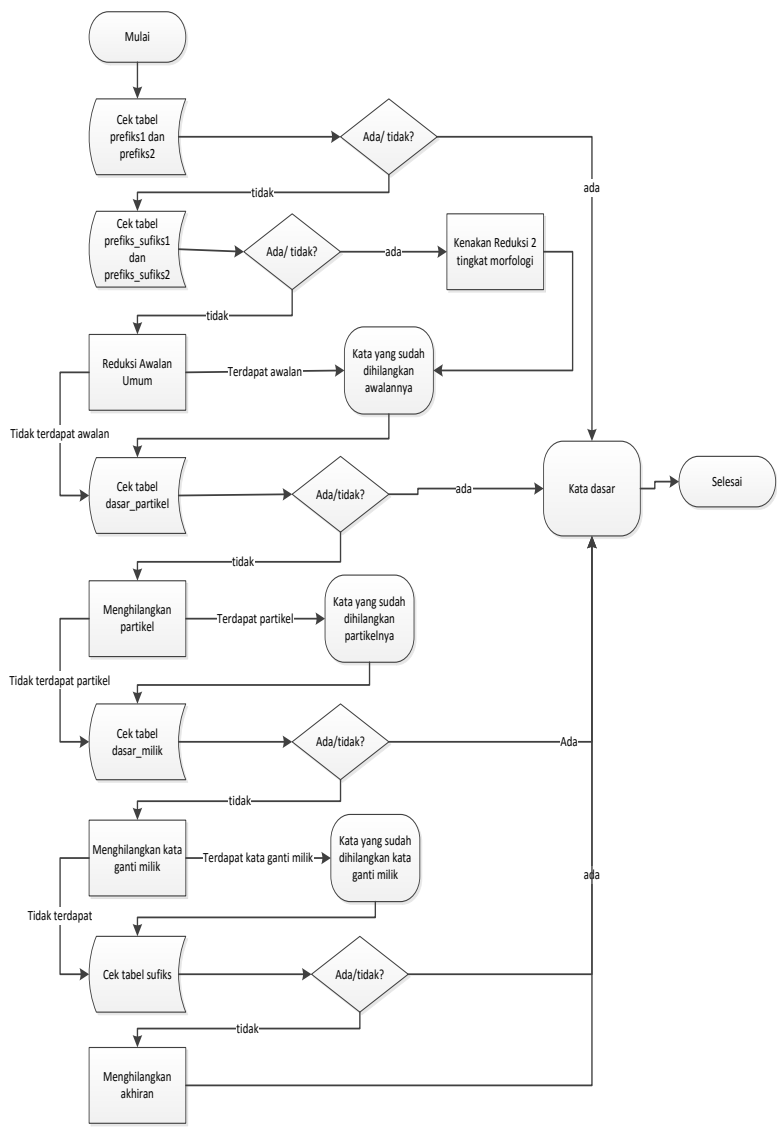
Pada tahap ini dilakukan penghilangan karakter yang memiliki frekuensi tinggi, karena dianggap bukan merupakan kata penting. Kata – kata tersebut antara lain: preposisi, konjungsi, dan lain – lain. Kata –kata yang termasuk dalam stoplist disebut *stopwords* dan telah disimpan dalam *database*.

#### 4.1.4 Penerapan Algoritma Peningkatan Porter pada proses *Stemming*

Algoritma *stemming* yang digunakan dalam Tugas Akhir ini adalah *Peningkatan Porter Stemmer*. Seperti yang telah dijelaskan sebelumnya, algoritma ini telah diimplementasikan pada [5] dan merupakan algoritma *stemming* untuk Bahasa Indonesia. Alur kerja *stemmer* dapat dilihat pada Gambar 4.2. Input, output dan langkah – langkah *stemmer* dijelaskan sebagai berikut:

- a. *Input* : Dokumen yang belum di-*stem*
- b. *Output* : Dokumen yang telah di-*stem*
- c. Langkah – langkah :
  1. Aplikasi memeriksa isi dokumen mulai kata pertama (dengan asumsi dokumen telah melalui proses *case folding*, *filtering* dan *stopping*)
  2. Aplikasi memeriksa apakah kata terdapat dalam tabel *dsr\_prefiks1* atau *dsr\_prefiks2*. Bila iya, maka kata tersebut merupakan kata dasar yang tidak perlu melalui proses *stemming* dan langsung melakukan langkah 8. Bila tidak, maka lakukan langkah 3.
  3. Aplikasi memeriksa apakah kata terdapat pada table *dsr\_prefiks\_sufiks1* atau *dsr\_prefiks\_sufiks2*. Bila iya,

- maka hilangkan awalan pada kata dan lakukan langkah 4. Bila tidak, lakukan langkah 5.
4. Lakukan *recoding* bila perlu. Bila tidak, maka langsung ke langkah selanjutnya.
  5. Aplikasi memeriksa apakah kata terdapat dalam tabel *dsr\_partikel*. Bila ya, maka kata tersebut merupakan kata dasar yang tidak perlu melalui proses *stemming* dan langsung melakukan langkah 8. Bila tidak, maka hilangkan partikel pada kata.
  6. Aplikasi memeriksa apakah kata terdapat dalam tabel *dsr\_milik*. Bila ya, maka kata tersebut merupakan kata dasar yang tidak perlu melalui proses *stemming* dan langsung melakukan langkah 8. Bila tidak, maka hilangkan kata ganti milik pada kata.
  7. Aplikasi memeriksa apakah kata terdapat dalam tabel *dsr\_sufiks*. Bila ya, maka kata tersebut merupakan kata dasar yang tidak perlu melalui proses *stemming* dan langsung melakukan langkah 8. Bila tidak, maka hilangkan akhiran pada kata.
  8. Seluruh kata disimpan dalam *string*
  9. Aplikasi memeriksa apakah kata – kata dalam dokumen sudah habis. Bila ya, maka lanjutkan ke langkah berikutnya. Bila tidak, maka ulangi dari langkah 2.
  10. Simpan kata – kata hasil *stemming*.



**Gambar 4.2** Proses *Stemming*



#### 4.1.5 Weighting

*Weighting* adalah pembobotan setiap terms yang telah di-stem melalui metode TF-IDF [1].

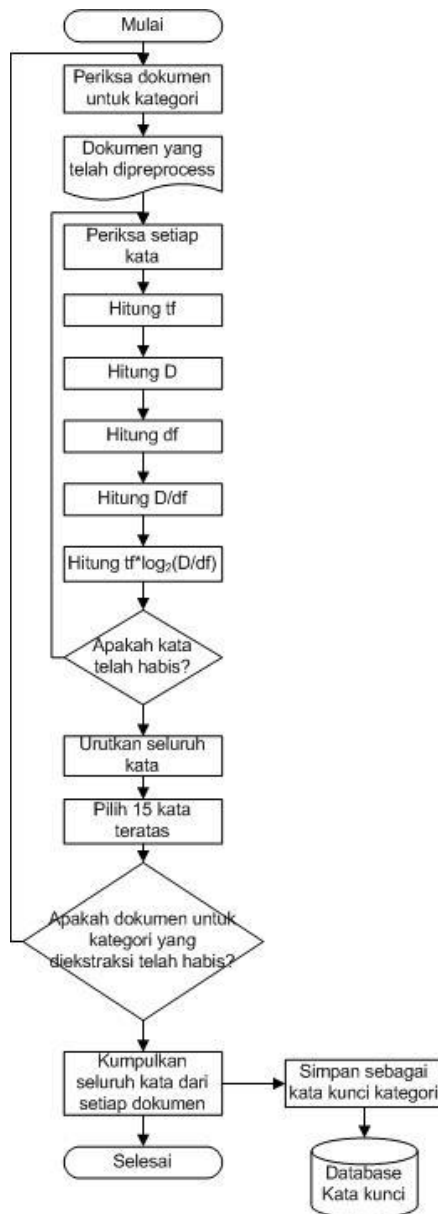
#### 4.1.6 Ekstraksi Kata Kunci

Kata kunci diekstraksi menggunakan metode TFIDF. Kata kunci yang dimaksud dalam Tugas Akhir ini adalah kata – kata yang dianggap sebagai kata – kata yang penting untuk membedakan antara sebuah dokumen dengan dokumen lainnya. Secara lebih spesifik, sebuah kata dapat dikatakan sebagai sebuah kata kunci adalah bila kata tersebut termasuk dalam 5 - 20 kata yang memiliki bobot TFIDF paling tinggi dari sebuah dokumen. Jumlah ini dipilih karena setelah melalui eksperimen, jumlah tersebut mampu memperlihatkan isi dokumen secara padat. *Input*, *output* dan langkah – langkah ekstraksi kata kunci adalah sebagai berikut:

- a. *Input*: Dokumen hasil *preprocessing*
- b. *Output*: Daftar kata kunci
- c. Langkah – langkah :
  1. Mulai dengan mengambil dokumen untuk sebuah kategori
  2. Periksa setiap kata pada dokumen tersebut
  3. Hitung frekuensi kemunculan kata pada sebuah dokumen (*tf*)
  4. Hitung jumlah koleksi dokumen yang dipunyai (tidak hanya pada kategori yang diperiksa) (*D*)
  5. Hitung berapa dokumen yang memuat kata tersebut (*df*). Sebagai contoh, apabila sebuah kata muncul di 3 dokumen (berapapun frekuensi kemunculan kata tersebut pada tiap dokumen), maka nilai *df* adalah 3
  6. Hitung  $D/df$  untuk setiap kata
  7. Untuk setiap kata, hitung  $tf \cdot \log_2 \left( \frac{N}{df} \right)$

8. Periksa apakah kata – kata telah habis. Bila iya, maka lanjutkan ke langkah berikutnya. Bila tidak. Maka ulangi dari langkah 2
9. Setelah didapatkan nilai TFIDF untuk sebuah dokumen, maka urutkan kata – kata menurut bobotnya
10. Ambil 5 - 20 kata dengan bobot tertinggi.
11. Kumpulkan seluruh kata tersebut dan simpan dalam *database* kata kunci sebagai kata kunci kategori.
12. Lakukan untuk seluruh dokumen pada kategori yang diperiksa. Apabila dokumen *training* pada kategori tersebut telah habis, maka ulangi mulai langkah 2 untuk kategori lainnya.

Proses ekstraksi kata kunci secara garis besar dapat dilihat pada Gambar 4.3.



**Gambar 4.3** Ekstraksi Kata Kunci

#### 4.1.7 Perhitungan *Likelihood* dan Nilai Ambang

Setelah *classifier* *ditraining* melalui *preprocessing* dan ekstraksi kata kunci, subproses selanjutnya adalah perhitungan nilai *likelihood* antara dokumen uji yang akan diklasifikasikan. Nilai *likelihood* dihitung mengikuti persamaan 5. Dalam rumus tersebut, nilai yang dibutuhkan adalah nilai  $P(k_i|c_j)$ . Nilai ini merupakan nilai proposi sebuah kata kunci terdapat dalam kategori tertentu. Contoh perhitungan  $P(k_i|c_j)$  dijelaskan pada Tabel 4.1 *likelihood* pada Tabel 4.2.

Tabel 4.1 Contoh Perhitungan  $P(k_i|c_j)$

[illegible]

Tabel 4.2 Contoh Perhitungan *Likelihood* untuk Kategori  $c_5$ 

Kata Kunci	$P(\text{kata kunci} \mid \text{kategori})$	$\log_2(P)$	$P * \log_2(P)$
<b>kerja</b>	0.0200000000	-5.6438561898	-0.1128771238
<b>uang</b>	0.0100000000	-6.6438561898	-0.0664385619
<b>harga</b>	0.0000000000	0.0000000000	0.0000000000
<b>jual</b>	0.0000000000	0.0000000000	0.0000000000
<b>usaha</b>	0.0100000000	-6.6438561898	-0.0664385619
<b>bisnis</b>	0.0300000000	-5.0588936891	-0.1517668107
<b>neraca</b>	0.0300000000	-5.0588936891	-0.1517668107
<b>stabil</b>	0.0500000000	-4.3219280949	-0.2160964047
<b>tingkat</b>	0.0400000000	-4.6438561898	-0.1857542476
<b>level</b>	0.0000000000	0.0000000000	0.0000000000
	<b>Nilai <i>Likelihood</i></b>		0.9511385213

$k_i$  merupakan kata kunci ke -  $i$  dari dokumen uji, dalam contoh ini jumlah kata kunci yang diambil adalah sepuluh dan  $c_i$  adalah kategori ke -  $i$  yang merupakan kategori primitif yang telah ditetapkan sebelumnya. Tabel 4.2 mencontohkan perhitungan *likelihood* antara kata kunci dokumen uji dengan kategori  $c_5$  yaitu kategori Bisnis & Ekonomi.

Dengan mengalikan  $P(k_i|c_j)$  dengan  $\log_2(P)$ , menjumlahkannya dan mengalikannya dengan -1, maka didapatkan nilai *likelihood* antara dokumen uji dengan kategori Bisnis dan Ekonomi sebesar 0,9511385213. Perhitungan *likelihood* dilakukan untuk seluruh kategori. Setelah *likelihood* untuk seluruh kategori dihitung, maka tahap selanjutnya adalah perhitungan nilai ambang. Perhitungan nilai ambang atau *threshold* didapatkan melalui penambahan rata – rata *likelihood* ditambah simpangan baku. Perhitungan *threshold* sesuai dengan persamaan 6 dan dicontohkan pada Tabel 4.3.

Tabel 4.3 Contoh Perhitungan *Threshold*

Likelihood <sub>i</sub> - Mean		(Likelihood – Mean) <sup>2</sup>
<b>Likelihood<sub>1</sub> – Mean</b>	0.0700000110	0.004900002
<b>Likelihood<sub>2</sub> – Mean</b>	0.1328771238	0.017656330
<b>Likelihood<sub>3</sub> – Mean</b>	0.0588546100	0.003463865
<b>Likelihood<sub>4</sub> – Mean</b>	0.0700000110	0.004900002
<b>Likelihood<sub>5</sub> – Mean</b>	0.9848402344	0.969910287
<b>Likelihood<sub>6</sub> – Mean</b>	0.0021317011	4.54415E-06
<b>Likelihood<sub>7</sub> – Mean</b>	0.0021317011	4.54415E-06
<b>Likelihood<sub>8</sub> – Mean</b>	0.0021317011	4.54415E-06
<b>Likelihood<sub>9</sub> – Mean</b>	0.0021317011	4.54415E-06
Sum		1.000848662
Mean		0.070000011
L		9
Sum /  L		0.111205407
Standard Deviasi		1.112054069
Threshold		1.18205408

Setelah aplikasi mendapatkan *P* dan *likelihood* untuk seluruh kategori dan menghitung *threshold*nya, lalu setiap *likelihood* dibandingkan dengan nilai *threshold* dan kategori yang memiliki nilai *likelihood* lebih besar daripada *threshold* dianggap sebagai kategori yang sesuai untuk dokumen.

#### 4.1.8 Perhitungan *CosSim* Topik

Untuk mengidentifikasi topik yang dimiliki sebuah dokumen, aplikasi melakukan dua tahap besar, yaitu perhitungan *CosSim* antara dokumen uji dengan topik – topik

yang terdapat dalam *database* kemudian perhitungan nilai ambang untuk menentukan apakah topik yang telah tersimpan dalam *database* sudah sesuai untuk dokumen uji. Langkah – langkah perhitungan *CosSim* adalah sebagai berikut:

1. Ambil kata kunci dokumen uji dari *database* kata kunci dan frekuensi kemunculannya dalam dokumen dari *database* vektor
2. Ambil kata kunci topik dari *database* kata kunci beserta jumlah dokumen tempat kata tersebut muncul. Jumlah dokumen tempat kata kunci tersebut ditemukan didapatkan melalui *query*
3. Transformasikan vektor kata kunci dokumen uji dan vektor kata kunci topik. Vektor kata kunci dokumen berisi skor frekuensi kemunculannya dalam dokumen, sedangkan vektor kata kunci topik berisi skor jumlah dokumen tempat kata tersebut muncul. Transformasi dicontohkan pada Gambar 3.2
4. Hitung panjang masing – masing vektor yang telah ditransformasikan
5. Hitung *CosSim* antara topik dengan dokumen uji dengan persamaan 7.
6. Ulangi dari langkah 1 untuk semua topik di *database* topik
7. Pilih topik yang memiliki *CosSim* terbesar
8. Tetapkan topik tersebut sebagai topik awal dokumen

Contoh perhitungan *CosSim* topik dijelaskan pada Tabel 4.4. Perhitungan *CosSim* cukup sederhana, mengikuti konsep perhitungan *cosine similarity* yang sering digunakan pada *vector space model*. Vektor yang dicontohkan pada Tabel 4.4 diasumsikan telah diketahui masing – masing skor kata kuncinya. Pada masing – masing vektor terdapat skor sebesar 0. Skor tersebut menunjukkan bahwa kata kunci tidak terdapat pada dokumen uji maupun pada kumpulan kata kunci topik.

Tabel 4.4 Contoh Perhitungan *CosSim*

Vektor dokumen uji	Vektor topik	Dot product	Panjang vektor dokumen uji	Panjang vektor topik
<b>10</b>	8	80	100	64
<b>2</b>	0	0	4	0
<b>5</b>	9	45	25	81
<b>3</b>	6	18	9	36
<b>7</b>	10	70	49	100
<b>11</b>	7	77	121	49
<b>0</b>	5	0	0	25
<b>0</b>	5	0	0	25
<b>4</b>	2	8	16	4
	<b>Sum</b>	<b>298</b>	<b>324</b>	<b>19,59</b>
			<b>CosSim</b>	0,84

Perhitungan *CosSim* menggunakan persamaan 7 seperti yang dicontohkan pada Tabel 3.4 dilakukan untuk seluruh topik yang ada pada *database*. Seluruh nilai *CosSim* kemudian diurutkan dan diambil nilai terbesarnya. Topik dengan nilai terbesar ditetapkan sebagai topik awal. Topik awal ini belum merupakan topik final untuk dokumen uji, karena pada tahap berikutnya topik tersebut dapat saja tidak memenuhi nilai ambang sehingga harus dimasukkan topik baru untuk dokumen.



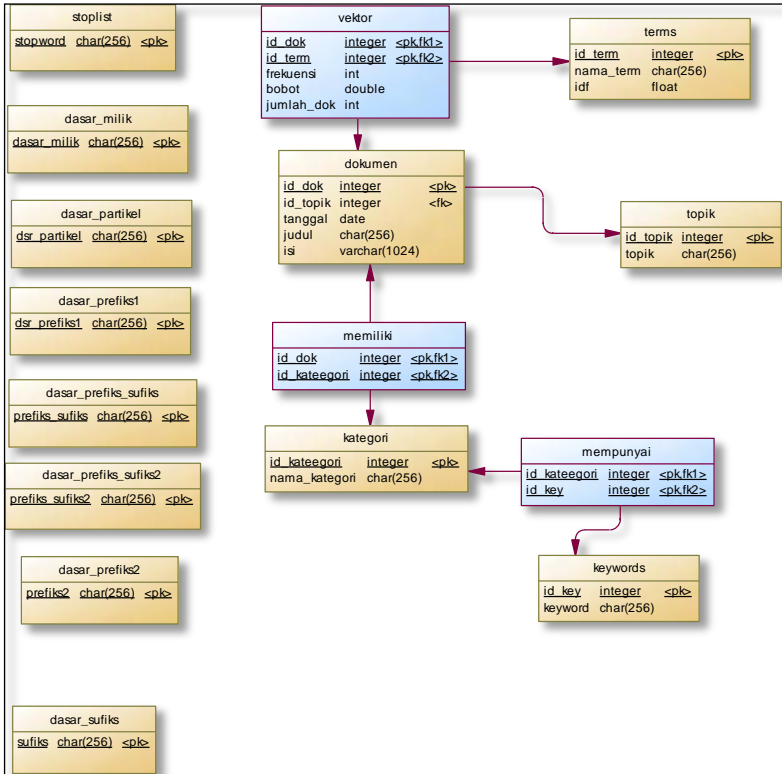
#### 4.1.9 Perhitungan Nilai Ambang *CosSim*

Bila sebuah topik awal telah ditetapkan untuk dokumen, maka topik tersebut perlu diuji kembali apakah telah sesuai untuk dokumen. Hal ini dilakukan karena topik baru muncul setiap hari, dan terdapat kemungkinan topik data *training* tidak dapat memenuhi seluruh topik baru yang mungkin muncul di kemudian hari.

Oleh karena itu, untuk dapat mengetahui apakah topik tersebut telah sesuai untuk dokumen, topik harus memenuhi dua nilai ambang, yaitu pertidaksamaan 9(i) dan 9(ii). Dalam nilai ambang tersebut, terdapat beberapa komponen baru yang belum didapatkan dari perhitungan *CosSim* sebelumnya, yaitu  $NewTSim(t_c, A)$ ,  $Mean(A)$ ,  $StdDev(A)$ ,  $|A|$ ,  $Mean(t_c)$  dan  $|t_c|$ .  $NewTSim((t_c, A)$  merupakan nilai topik hipotetis yang melambangkan topik baru yang mungkin lebih sesuai untuk dokumen, sedangkan  $t_c$  adalah topik awal (*topic conditional*) dan  $A$  adalah dokumen uji  $A$ . Nilai  $NewTSim((t_c, A)$  didapatkan sesuai rumus 8.  $Mean(A)$  merupakan rata – rata vektor dokumen uji  $A$ ,  $StdDev(A)$  adalah simpangan baku vektor dokumen uji  $A$ ,  $|A|$  adalah panjang vektor dokumen uji  $A$ , sedangkan  $Mean(t_c)$  dan  $|t_c|$  secara berturut – turut adalah rata – rata untuk topik awal dan panjang vektor topik awal.

Selain  $NewTSim(t_c, A)$ , komponen lain yang harus dihitung oleh aplikasi adalah  $StdDev(AllTopicSims)$  dan  $Mean(AllTopicSims)$  dua komponen tersebut secara berturut – turut adalah simpangan baku dari seluruh *CosSim* yang telah dihitung dan rata – ratanya. Bila topik awal memenuhi kedua nilai ambang, maka topik tersebut dinilai sudah sesuai untuk dokumen. Bila tidak, maka topik baru harus diberikan.

## 4.2 Desain Physical Data Model

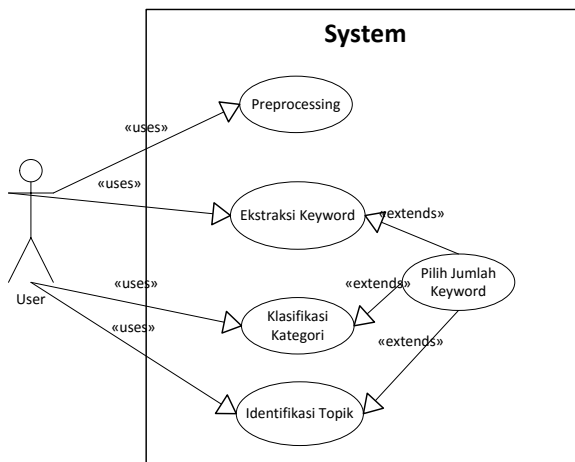


**Gambar 4.4** Desain Physical Data Model

*Desain Physical Data Model* di atas selanjutnya akan digunakan sebagai dasar untuk merancang database menggunakan MySQL.

### 4.3 Use Case Diagram

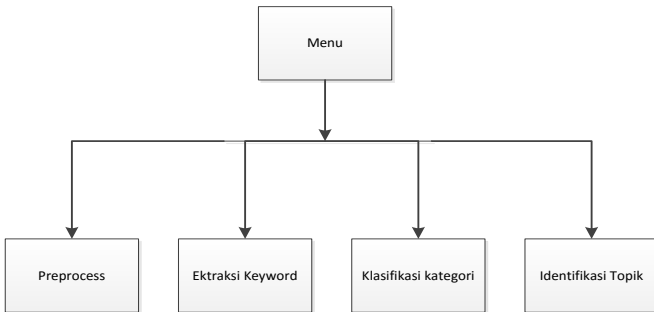
*Use case diagram* merupakan diagram yang menggambarkan interaksi antara *user* dengan sistem. Pada Gambar 4.5 dapat dilihat bahwa *user* dapat melakukan *preprocessing*, ekstraksi *keyword*, klasifikasi kategori dan identifikasi topik. *User* dapat melakukan pemilihan jumlah kata kunci yang kemudian sistem akan menampilkan hasil dari proses ekstraksi *keyword*, klasifikasi kategori dan identifikasi topik.



**Gambar 4.5** Use Case Diagram PL

### 4.4 Diagram Rancangan Interface

Pada program ini dibuat empat menu utama, yaitu menu *preprocess*, Ekstraksi *Keyword*, Klasifikasi Kategori dan Identifikasi Topik seperti pada Gambar 4.6.



**Gambar 4.6** Diagram Rancangan *Interface*

#### 4.5 Implementasi *Case Folding*

Proses ini mengubah seluruh huruf pada setiap kata menjadi huruf kecil.

```

public void doLowerCaseIsi()
{
    this.isi = this.isi.toLowerCase();
}
  
```

#### 4.6 Implementasi *Filtering* (Eliminasi Tanda Baca)

```

for(int i=0; i<this.isi.length(); i++)
{
    if( CarTerlarang(this.isi.charAt(i)) )
        this.isi = isi.substring(0,i) + " " +
            isi.substring(i+1,isi.length());
}
  
```

#### 4.7 Implementasi *Stoplist Removal* (Penghilangan Kata-Kata Tidak Penting)

```

String[] temp = isi.trim().split(" ");
for(int i=0; i<temp.length; i++)
{ if(temp[i].length()>=2)
    { if(adaStopword)
  
```

```

if(adaStopword)
{ if(!stopword.isContain(temp[i]))
    vektor.addTerm(temp[i],1);
  }
  else
    vektor.addTerm(temp[i],1);
  }
}

```

#### 4.8 Implementasi Proses *Stemming*

Setelah selesai melalui tahap *case folding*, *filtering* maka langkah selanjutnya adalah proses *stemming*. Pada proses ini akan diterapkan algoritma peningkatan Porter Stemmer. Berikut merupakan implementasi dari metode stemming menggunakan algoritma peningkatan Porter Stemmer :

```

//Tahap 1. Cek tabel dsr_prefiks1 dan dsr_prefiks2
if(dictionaryLookup1(kata)||dictionaryLookup3(kata)
{
    this.bentukDasar = kata;
}
// jika tidak ada maka cek table
drs_prefiks1_sufiks1 dan dsr_prefiks2_sufiks2
// lakukan reduksi awalan 1 dan awalan 2 sehingga
menjadi kata yang sudah dihilangkan awalannya
else if
(dictionaryLookup2(kata)||dictionaryLookup4(kata)
{
    kata = reduksiAwalan(kata, DS, true);

if(dictionaryLookup2(kata)||dictionaryLookup4(kata)
{
    this.bentukDasar=kata;
}
    else if(dictionaryLookup5(kata)){// cek

```

```

        else{
            partikel = FungsiStem.getPartikel(kata);
            kata=reduksiPartikel(kata,partikel);
            if(dictionaryLookup5(kata)){
                this.bentukDasar=kata;
            }
            else if(dictionaryLookup6(kata)){// cek tabel
                dsr_milik
                this.bentukDasar=kata;
            }
        }
        else{
            kataGantiKepunyaan =
            FungsiStem.getKataGantiKepunyaan(kata.substring(0,k
            ata.length()-partikel.length()) );
            kata=reduksiMilik(kata,kataGantiKepunyaan);

            if(dictionaryLookup7(kata)){// cek tabel dsr_sufiks
                this.bentukDasar=kata;
            }
        }
        else {
            DS = FungsiStem.getDS
            (kata.substring(0,kata.length()-partikel.length()-
            kataGantiKepunyaan.length()))
            if(!"".equals(DS)){
                kata=redAkhiran(kata,DS);
            }
            else if(FungsiStem.cekKombinasiTerlarang(kata,DS)){
                kata=redAkhiran(kata,DS);
                this.bentukDasar=kata;
            }
            DS=FungsiStem.getDS2(kata.substring(0,kata.length()
            -partikel.length()-kataGantiKepunyaan.length()));
            if (!"".equals(DS)){
                kata=redAkhiran(kata,DS);
            }
            this.bentukDasar=kata;
        }
    }
}
}
}

```

**Gambar 4.7** Segmen Program Proses *Stemming*

Pada kode di atas, terdapat beberapa *method* yang digunakan, yaitu:

1. dictionaryLookup1() s.d dictionaryLookup7()

Metode ini berfungsi untuk melakukan pemeriksaan pada kata yang akan *distemming* terdapat pada kamus apa tidak. Karena pada algoritma Porter Stemmer ini menggunakan tujuh tabel maka dibuat tujuh metode yang melakukan fungsi di atas.

2. reduksiAwalan()

Metode ini berfungsi untuk melakukan reduksi awalan pertama dan awalan kedua pada kata yang akan *distemming*. Pada metode ini juga digunakan fungsi *recoding()* yang bertugas untuk melakukan penambahan karakter tertentu pada kata yang mengalami proses reduksi dengan dua tingkat morfologi. Contoh : kata “memesan” memuat awalan *me-* yang apabila direduksi awalan maka kata tersebut menjadi “mesan” dimana kata “mesan” masih perlu satu proses lagi yaitu penggantian karakter pertama dengan karakter “p” sehingga menjadi kata dasar “pesan”. Proses tersebut dilakukan pada fungsi *recoding()* seperti yang sudah diterapkan pada [2].

```

rule = FungsiStem.getRule(word); //cari Rule
awalan yg akan dihilangkan
prefix = FungsiStem.getPrefix(rule,word);
//dapatkan string prefix sesuai rule-nya
    if(rule==0){
        rule = logRules;
        break;
    }
    word =
FungsiStem.getReduksi(rule,word);

    logRules = rule;
    logPrefix = prefix; //dicatat ke
log
    count++;
}while(count<=3);
if(isRecoded){
    word = recoding(rule,word);
}
return word;

```

**Gambar 4.8** Segmen Program Proses Reduksi Awalan

### 3. redAkhiran()

Metode ini berfungsi melakukan reduksi akhiran pada kata yang akan *distemming*. Dengan mendapatkan terlebih dahulu suatu kata mengandung akhiran *-kan,-an,-isme, -isasi,-onal* apa tidak menggunakan fungsi *getDS()*. Jika terdapat akhiran di atas, dilakukan reduksi akhiran. Jika tidak ada maka diperiksa apakah kata tersebut termasuk kata berimbuhan yang memiliki kombinasi terlarang atau tidak. Jika benar maka dilakukan reduksi akhiran, sedangkan jika salah maka dapatkan akhiran *-i* pada kata tersebut menggunakan fungsi *getDS2()*. Jika pada kata terdapat akhiran *-i* maka lakukan reduksi akhiran, jika tidak ada maka kata dikembalikan menjadi kata dasar.



```
private String redAkhiran(String kata, String DS){
    if(!DS.equals("")){
        kata = kata.substring(0,kata.length()-
DS.length());
        if(dictionaryLookup7(kata))
            return kata;
    }
    return kata;
}
```

**Gambar 4.9** *Source Code* Proses Reduksi Akhiran

```
public static String getDS2(String kata)
{
    String DS2 = "";
    //reduksi "-i"
    if(kata.endsWith("i"))
        DS2 = "i";
    return DS2;
}
```

**Gambar 4.10** *Source Code* Metode getDS2()

#### 4. reduksiMilik() dan reduksiPartikel()

Sama halnya seperti reduksi awalan, metode ini berfungsi untuk melakukan reduksi pada partikel -lah,-kah,-tah,-pun dan kata ganti milik -ku, -mu, -nya pada kata yang akan distemming.

```
private String reduksiPartikel(String kata,String
partikel){
    if(!partikel.equals("")){
        kata = kata.substring(0,kata.length()-
partikel.length());
        if(dictionaryLookup5(kata))
            return kata;
    }
}
```

**Gambar 4.11** *Source Code* Reduksi Partikel

```

private String reduksiMilik(String kata,
String kataGantiKepunyaan){
    //lalu reduksi kata ganti kepunyaan
    if(!kataGantiKepunyaan.equals("")){
        kata =
kata.substring(0,kata.length()-
kataGantiKepunyaan.length());
        if(dictionaryLookup6(kata))
            return kata;
    }
    return kata;
}

```

**Gambar 4.12** *Source Code* Reduksi Kata Ganti Kepemilikan

#### 4.9 Implementasi Proses Ekstraksi *Keywords*

Setelah proses *preprocessing* selesai dilakukan dilanjutkan dengan proses ekstraksi *keyword*. Pada proses ini akan dilakukan pengambilan dokumen – dokumen *training* yang terdapat dalam setiap kategori dan mengekstraksi *keywords* dokumen tersebut lalu menyimpannya dalam *database* [2].

```

try{
    String sql="SELECT d.id_dokumen FROM
dokumen d, dok_kat dk, kategori k"+"WHERE
d.id_dokumen=dk.id_dokumen AND
dk.id_kategori=k.id_kategori"+
    "AND
k.nama_kategori='"+namaKategori+"'";
ResultSet rs=theKoneksi.executeSelect(sql);
insert ins=new insert();
LinkedList<String> idDokDgnKategori=new
LinkedList<String>();
while(rs.next()){
    idDokDgnKategori.add(rs.getString(1));}
for(String idDok:idDokDgnKategori)
{
    sql="SELECT t.nama_term,v.frekuensi,v.bobot"+"FROM
terms t, vektor v"+"WHERE
v.id_dokumen="+idDok+"AND
t.id_term=v.id_term"+"ORDER BY v.bobot DESC"+
    "limit 0,10";
    rs=theKoneksi.executeSelect(sql);
    while(rs.next())
    {
        String sql2="SELECT id_kategori FROM kategori
WHERE"+"nama_kategori='"+namaKategori+"'";
        ResultSet r=theKoneksi.executeSelect(sql);
        int idKat=0;
        while(r.next())
        {
            idKat=Integer.parseInt(r.getString(1));
            ins.insertKat_Key(rs.getString(1),idKat);
        }
    }
}
}
catch (SQLException ex)

```

**Gambar 4.13** Segmen Program Ekstraksi *Keyword*

#### 4.10 Implementasi Proses Klasifikasi Kategori

Proses klasifikasi kategori harus melakukan proses *parsing* dokumen uji agar *corpus* dokumen yang masih mengikuti format penulisan *corpus* dapat diambil isi dokumennya yang akan diklasifikasi. Karena *corpus* memiliki format seperti pada Gambar 3.2, maka aplikasi harus memecah setiap atribut – atribut yang dimilikinya untuk dimasukkan ke *database* dan terutama mengambil isinya untuk dihitung *likelihood*nya [2].

```

if (dis.available() != 0) {
    News news = new News();
    news.tgl_sumber = dis.readLine();
    news.topik = dis.readLine();
    news.id_sumber =
Integer.valueOf(dis.readLine());
    news.judul = dis.readLine();
    while(dis.available() != 0)
    {
        news.isi = news.isi + dis.readLine();
    }
    insertData(news);
    everything[0] = news.judul.toString();
    everything[1] = news.isi.toString();
    everything[2] = news.tgl_sumber.toString();
    everything[3] = news.topik.toString();
}

```

**Gambar 4.14** Segmen Program Proses *Parsing* Data

Tahap utama berikutnya adalah perhitungan nilai *likelihood*. Metode *likelihood* berfungsi untuk menghitung *likelihood* itu sendiri sekaligus menghitung *mean*nya. Metode yang selanjutnya yaitu *pCocokan* berfungsi untuk menyeleksi kategori mana yang melebihi nilai *threshold* sehingga layak dianggap sebagai kategori untuk dokumen uji.

```

for(int j = 0; j < index; j++)
{
    temppllogp = 0.0;
    for(int i = 0; i < batas; i++)
    {
        p = (double)indexKat[i][j] /
(double)totDok[j];

        if(p != 0.0)
            temppllogp =
temppllogp+p*Math.log10(p);
        else temppllogp = temppllogp + 0;
    }
    plogp[j] = temppllogp != 0.0 ?
temppllogp*-1: temppllogp;
}
// menghitung mean
for(int i = 0; i < index; i++)
{
    mean = mean + plogp[i];
}
mean = mean / index; // ini mean sebenarnya

```

**Gambar 4.15** Segmen Program Perhitungan *Likelihood*

```

for(int i = 0; i < lengthArray; i++ )
{
    liMean[i] = plogp[i] - mean;
    kuadratLiMean[i] = Math.pow(liMean[i],
(double)2);
    sumKuadrat = sumKuadrat +
kuadratLiMean[i];
}
double newSumKuadrat = sumKuadrat /
lengthArray;
standaDev = Math.sqrt(newSumKuadrat);
threshold = standaDev + mean;

```

**Gambar 4.16** Segmen Program Perhitungan Standar Deviasi dan *Threshold*

```

for(int i = 0 ; i < lengthArray; i++)
{
    if(plogp[i] >= threshold{
String dok = namaKategori.get(i);
System.out.println("Dokumen ini termasuk
kategori : "+dok);
tidakCocok = false;
namaKategoriLagi.add(dok);
    }
}
if(tidakCocok)
{
    namaKategoriLagi.add("tidak ada kategori yang
cocok");
}

```

**Gambar 4.17** Segmen Program Metode pCocokan()

#### 4.11 Implementasi Proses Identifikasi Topik

Proses identifikasi topik meliputi dua proses utama, yaitu perhitungan *CosSim* dan perhitungan *threshold*.

```

for (int i = 0; i < batas; i++) // mencari keyword
dok A di topik dan dokumen a
{
    vektorTopik vt =
listVektorTopikGabungan.get(i);
    vektorDokA[i] = vt.frekuensi;
    vektorTI[i] = getFreqTI(vt.keyword);
    //System.out.println(vektorDokA[i]+"
"+vektorTI[i]+" "+vt.keyword);
}
for (int i = batas; i < sizeVektorGab; i++)
// mencari frekuensi keyword topik pada dokumen dan
topik
{

```

```

vektorTopik vt = listVektorTopikGabungan.get(i);
    vektorDokA[i] =
getFreqTopikPadaDok(vt.keyword);

vektorTI[i] = getFreqTopikPadaTopik(vt.keyword);
    }
    this.dokumenTerbaik.add(vektorDokA);
    this.terbaik.add(vektorTI);
    dotProduk(vektorTI, vektorDokA);

```

**Gambar 4.18** Segmen Program Perhitungan *CosSim*

```

try {
kon.connectFirst();
String getFreq = "select v.frekuensi "+ "from
vektor v, terms t "+ "where v.id_term = t.id_term "
+ "and t.nama_term = '" + keyword + "'" + "and
v.id_dok = " + getIDok();
ResultSet rsl = kon.executeSelect(getFreq);
while (rsl.next()) {
    retFreq = rsl.getInt(1);
    }
    kon.destroyConnection();
    } catch (Exception e) {
        e.printStackTrace();
    }
    }
    return retFreq;
for (int i = 0; i < sizeArray; i++) {
    pembilang = pembilang + (vektorDokA[i] *
vektorTI[i]);
    penyebutA = penyebutA +
Math.pow(vektorTI[i], (double) 2);
    penyebutB = penyebutB +
Math.pow(vektorDokA[i], (double) 2);
    }
    double hasil = pembilang /
(Math.sqrt(penyebutA) * Math.sqrt(penyebutB));
    sethasilCosSim(hasil);

```

**Gambar 4.19** Segmen Program Metode dotProduk()

```

LinkedList<vektorTopik> listVektorTopikGabungan
= new LinkedList<vektorTopik>();
    for (vektorTopik vt :
listVektorTopikDokA) {
        listVektorTopikGabungan.add(vt);
    }
    for (int i = 0; i <
listVekTopik.size(); i++) {
        boolean tidakSama = true;
        String keyWordDokA = "";
        String keyWord = "";
        for (int j = 0; j <
listVektorTopikGabungan.size(); j++) {
            keyWordDokA =
listVektorTopikGabungan.get(j).keyword;
            keyWord =
listVekTopik.get(i).keyword;
            if
(keyWord.equals(keyWordDokA)) {
                tidakSama = false;
                j =
listVektorTopikGabungan.size();
            }
        }
        if (tidakSama) {
            vektorTopik vt = new
vektorTopik(keyWord, 0);
            listVektorTopikGabungan.add(vt);
        }
    }
}

```

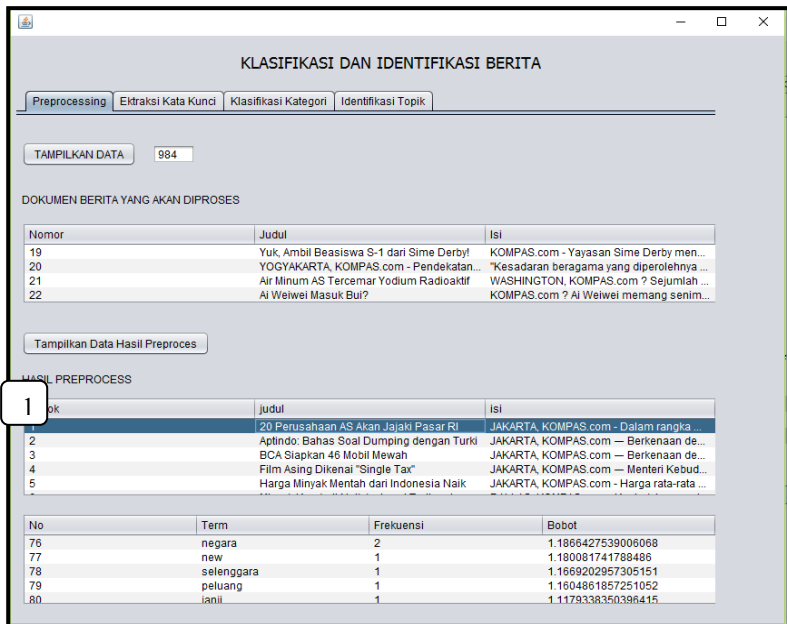
**Gambar 4.20** Segmen Program *formatToVektor*

#### 4.12 Implementasi *Interface*

*Interface* pada aplikasi ini memiliki empat *tab* utama, yaitu *tab* Preprocessing, *tab* Ekstraksi Kata Kunci, *tab* Klasifikasi Kategori dan *tab* Identifikasi Topik. *Tab* Preprocessing memiliki fitur utama yaitu “Preprocessing”. Ketika tombol Preprocess ditekan maka akan ditampilkan hasil *preprocessing* dari seluruh dokumen *training*. Bila kita klik salah satu dokumen seperti yang



saya tandai dengan 1 maka tabel kedua akan menampilkan seluruh terms dari dokumen yang dipilih beserta frekuensi dan bobot dari masing-masing terms.



**KLASIFIKASI DAN IDENTIFIKASI BERITA**

Preprocessing | Ekstraksi Kata Kunci | Klasifikasi Kategori | Identifikasi Topik

TAMPAILKAN DATA 984

DOKUMEN BERITA YANG AKAN DIPROSES

Nomor	Judul	Isi
19	Yuk, Ambil Beasiswa S-1 dari Sime Derby!	KOMPAS.com - Yayasan Sime Derby men...
20	YOGYAKARTA, KOMPAS.com - Pendekatan...	"Kesadaran beragama yang diperolehnya ...
21	Air Minum AS Tercemar Yodium Radioaktif	WASHINGTON, KOMPAS.com ? Sejumlah ...
22	Ai Weiwei Masuk Bul?	KOMPAS.com ? Ai Weiwei memang senim...

Tampilkan Data Hasil Preproses

**1** HASIL PREPROCESS

No	Term	Frekuensi	Bobot
76	negara	2	1.1866427539006068
77	new	1	1.180081741768486
78	selenggara	1	1.1669202957305151
79	peluang	1	1.1604861857251052
80	janil	1	1.1179338350396415

**Gambar 4.21** *Tab Preprocessing*

Tab selanjutnya adalah tab Ekstraksi Kata Kunci.



**Gambar 4.22** Tab Ekstraksi Keyword

Tab ketiga pada fitur klasifikasi kategori pengguna dapat memilih berita yang akan diklasifikasi melalui tombol “Browse”, dan pengguna dapat memilih jumlah kata kunci yang diekstraksi dalam proses ekstraksi kata kunci, setelah itu aplikasi akan memunculkan hasil proses ekstraksi kata kunci beserta bobotnya pada tabel dan nilai *likelihood* dengan setiap kategori pada tabel berikutnya. Bila seluruh proses perhitungan telah selesai dilakukan, maka aplikasi akan memunculkan hasil kategori yang memenuhi *standard threshold*.

**KLASIFIKASI DAN IDENTIFIKASI BERITA**

Preprocessing   Ekstraksi Kata Kunci   **Klasifikasi Kategori**   Identifikasi Topik

Dokumen Berita: D:\data testing\Perusahaan AS Akan Jajaki Pasar RI news   Browse   5   Klasifikasi

Hasil Preproses dan Ekstraksi

Nomor	Keywords	Bobot
1	meningkatkan	5.711775010478613
2	perusahaan	5.711775010478613
3	selasa	4.0526751068741005
4	scot	3.5563192949228544
5	deleksi	3.240653203529572

Hasil Perhitungan Likelihood Kategori

Nomor	Kategori	Likelihood
1	Nasional	0.03968953215411174
2	Regional	0.018558115319620226
3	Internasional	0.0
4	Metropolitan	0.05731997731881425
5	Bisnis dan Ekonomi	0.10729032506370176

KATEGORI BERITA: Sains dan Teknologi   Nilai Threshold: 0.1503475966890147

CLOSE   Identifikasi Topik

**Gambar 4.23** Tab Klasifikasi Kategori

*Tab* terakhir adalah identifikasi topik. Pengguna dapat memilih apakah dokumen yang ingin diidentifikasi sama dengan dokumen yang diklasifikasi pada *tab* sebelumnya atau dokumen berbeda. Apabila dokumen yang dimasukkan sama, maka pada *tab* “Klasifikasi Kategori” pengguna harus menekan tombol “Identifikasi Topik”. Apabila tidak, maka pengguna dapat memilih dokumen baru yang ingin diidentifikasi pada *tab* identifikasi topik. Seperti pada klasifikasi kategori, jumlah kata kunci yang diekstraksi juga dapat dipilih.

KLASIFIKASI DAN IDENTIFIKASI BERITA

Preprocessing

Ekstraksi Kata Kunci

Klasifikasi Kategori

Identifikasi Topik

Dokumen Be...

Perusahaan.AS.Akan.Jajaki.Pasar.RI.news

Browse

5

Judul Dokumen

Perusahaan AS Akan Jajaki Pasar RI

Identifikasi Topik

Kategori

Sains dan Teknologi

Keyword Dokumen

Nomor	Keywords	Frekuensi
1	indonesia	7
2	as	6
3	usaha	4
4	meningkatkan	4

CosSim Similarity

Topik	CoSim
Investasi	0.7248721135449595
FISIKA	0.5373283954075739
Saham	0.5211684385622208
Pameran Astindn	0.5211684385622208

TOPIK DOKUMEN :

Investasi

Gambar 4.24 Tab Identifikasi Topik

## **BAB V**

### **UJI COBA DAN PEMBAHASAN**

#### **5.1 Data Uji Coba**

**Karakteristik** : Data berupa corpus berita online berbahasa Indonesia yang didapatkan dari [www.kompas.com](http://www.kompas.com). Berita diunduh berdasar kategori yang telah ditetapkan. Kategori primitif dalam uji coba berguna untuk mengevaluasi hasil klasifikasi.

**Jumlah** : Antara sebuah kategori dengan kategori lainnya memiliki jumlah dokumen uji yang sama. Spesifikasi jumlah dokumen untuk setiap kategori dapat dilihat pada Tabel 5.1

**Tabel 5.1** Spesifikasi Jumlah Dokumen Setiap Kategori

<b>Kategori</b>	<b>Jumlah Dokumen</b>
Nasional	10
Regional	10
Internasional	10
Metropolitan	10
Bisnis dan Ekonomi	10
Olahraga	10
Sains dan Teknologi	10
Edukasi	10
Pariwisata	10
<b>Total</b>	<b>90</b>

## 5.2 Hasil Uji Coba

Berikut ini disajikan beberapa tabel hasil perhitungan evaluasi proses klasifikasi kategori dengan pemilihan jumlah kata kunci sebanyak 5, 10, 15, 20, 25 dengan menggunakan *precision*, *recall*, *f-measure*, dan *accuracy*.

**Tabel 5.2** Evaluasi Klasifikasi Kategori dengan 5 Kata Kunci

No	Kategori	Precision	Recall	F-Measure	Accuracy
1	Internasional	1.0000000000	0.8000000000	0.8888888889	1.0000000000
2	Nasional	0.8888888889	0.8888888889	0.8888888889	0.9000000000
3	Regional	0.8750000000	0.7777777778	0.8235294118	0.8000000000
4	Metropolitan	0.8750000000	0.7777777778	0.8235294118	0.8000000000
5	Bisnis Ekonomi	0.8888888889	0.8888888889	0.8888888889	0.9000000000
6	Olahraga	0.8750000000	0.7777777778	0.8235294118	0.9000000000
7	Pariwisata	0.8750000000	0.7777777778	0.8235294118	0.9000000000
8	Sains Teknologi	0.8888888889	0.8888888889	0.8888888889	1.0000000000
9	Edukasi	0.8888888889	0.8888888889	0.8888888889	0.9000000000
Rata-Rata		0.8950617284	0.8296296296	0.8598402324	0.9000000000

**Tabel 5.3** Evaluasi Klasifikasi Kategori dengan 10 Kata Kunci

No	Kategori	Precision	Recall	F-Measure	Accuracy
1	Internasional	1.0000000000	0.8000000000	0.8888888889	1.0000000000
2	Nasional	0.8888888889	0.8888888889	0.8888888889	1.0000000000
3	Regional	0.7777777778	0.7777777778	0.7777777778	0.7000000000
4	Metropolitan	0.8750000000	0.7777777778	0.8235294118	0.8000000000
5	Bisnis Ekonomi	0.8888888889	0.8888888889	0.8888888889	1.0000000000
6	Olahraga	0.8888888889	0.8888888889	0.8888888889	0.9000000000
7	Pariwisata	0.8888888889	0.8888888889	0.8888888889	1.0000000000
8	Sains Teknologi	0.8888888889	0.8888888889	0.8888888889	1.0000000000
9	Edukasi	0.8888888889	0.8888888889	0.8888888889	0.9000000000
Rata-Rata		0.8873456790	0.8543209877	0.8692810458	0.9222222222

**Tabel 5.4** Evaluasi Klasifikasi Kategori dengan 15 Kata Kunci

No	Kategori	Precision	Recall	F-Measure	Accuracy
1	Internasional	0.8888888889	0.8000000000	0.8421052632	0.9000000000
2	Nasional	0.8888888889	0.8888888889	0.8888888889	1.0000000000
3	Regional	0.8888888889	0.8888888889	0.8888888889	0.9000000000
4	Metropolitan	0.8750000000	0.7777777778	0.8235294118	0.8000000000
5	Bisnis Ekonomi	0.8888888889	0.8888888889	0.8888888889	1.0000000000
6	Olahraga	0.8888888889	0.8888888889	0.8888888889	0.9000000000
7	Pariwisata	0.8888888889	0.8888888889	0.8888888889	1.0000000000
8	Sains Teknologi	0.8888888889	0.8888888889	0.8888888889	1.0000000000
9	Edukasi	0.8888888889	0.8888888889	0.8888888889	0.9000000000
	Rata-rata	0.8873456790	0.8666666667	0.8764285441	0.9333333333

**Tabel 5.5** Evaluasi Klasifikasi Kategori dengan 20 Kata Kunci

No	Kategori	Precision	Recall	F-Measure	Accuracy
1	Internasional	0.8000000000	0.8000000000	0.8000000000	0.8000000000
2	Nasional	0.8888888889	0.8888888889	0.8888888889	1.0000000000
3	Regional	0.8888888889	0.8888888889	0.8888888889	0.9000000000
4	Metropolitan	1.0000000000	0.8888888889	0.9411764706	1.0000000000
5	Bisnis Ekonomi	0.8888888889	0.8888888889	0.8888888889	1.0000000000
6	Olahraga	0.8888888889	0.8888888889	0.8888888889	0.9000000000
7	Pariwisata	0.8888888889	0.8888888889	0.8888888889	1.0000000000
8	Sains Teknologi	0.8888888889	0.8888888889	0.8888888889	1.0000000000
9	Edukasi	0.8888888889	0.8888888889	0.8888888889	0.9000000000
	Rata-Rata	0.8913580247	0.8790123457	0.8848220770	0.9444444444

**Tabel 5.6** Evaluasi Klasifikasi Kategori dengan 25 Kata Kunci

No	Kategori	Precision	Recall	F-Measure	Accuracy
1	Internasional	0.888888889	0.800000000	0.8421052632	0.9000000000
2	Nasional	0.888888889	0.888888889	0.888888889	0.9000000000
3	Regional	0.888888889	0.888888889	0.888888889	0.9000000000
4	Metropolitan	1.000000000	0.888888889	0.9411764706	1.0000000000
5	Bisnis Ekonomi	1.000000000	0.888888889	0.9411764706	1.0000000000
6	Olahraga	0.888888889	0.888888889	0.888888889	0.9000000000
7	Pariwisata	1.000000000	0.888888889	0.9411764706	1.0000000000
8	Sains Teknologi	1.000000000	0.888888889	0.9411764706	1.0000000000
9	Edukasi	1.000000000	0.888888889	0.9411764706	1.0000000000
Rata-Rata		0.9506172840	0.8790123457	0.9127393648	0.9555555556

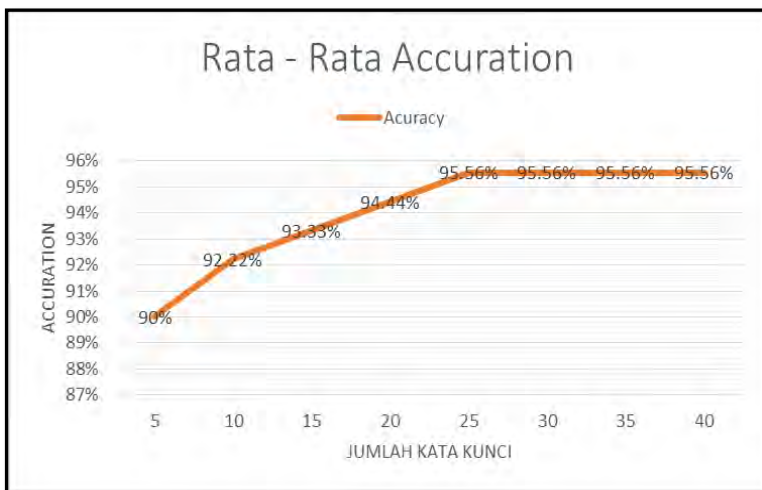
Pada tabel 5.2-5.6 dapat kita lihat bahwa hasil klasifikasi kategori menunjukkan hasil yang paling baik menggunakan evaluasi *accuracy*. Sehingga *accuracy* klasifikasi kategori dari tabel-tabel di atas dapat dirangkum sebagai berikut :

**Tabel 5.7** Rata – Rata Nilai *Accuration* Klasifikasi Kategori

	5	10	15	20	25
	Accuration	Accuration	Accuration	Accuration	Accuration
Kategori					
Internasional	100.00%	100.00%	90.00%	80.00%	100.00%
Nasional	90.00%	100.00%	100.00%	100.00%	100.00%
Regional	80.00%	70.00%	90.00%	90.00%	90.00%
Metropolitan	80.00%	80.00%	80.00%	100.00%	100.00%
Bisnis Ekonomi	90.00%	100.00%	100.00%	100.00%	100.00%
Olahraga	90.00%	90.00%	90.00%	90.00%	90.00%
Pariwisata	90.00%	100.00%	100.00%	100.00%	100.00%
Sains & Teknologi	100.00%	100.00%	100.00%	100.00%	100.00%
Edukasi	90.00%	90.00%	90.00%	90.00%	80.00%
Rata - rata	90.00%	92.22%	93,33%	94,44%	95,56%

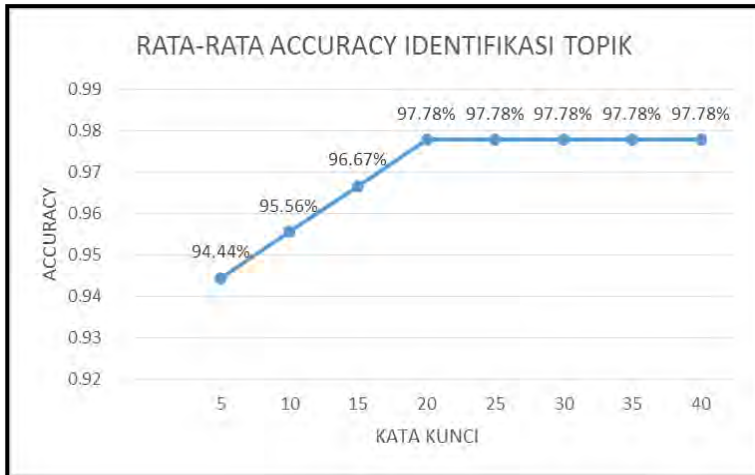


Data hasil perhitungan pada tabel 5.7 merupakan perhitungan *accuracy* untuk masing-masing kategori. Sebenarnya evaluasi dilakukan menggunakan pemilihan jumlah kata kunci sebesar 5, 10, 15, 20, 25, 30, 35, 40 dan didapatkan nilai paling maksimal adalah dengan menggunakan 25 kata kunci. Sebab pada jumlah kata kunci 30-40 menghasilkan rata-rata *accuracy* yang sama dengan 25. Hal tersebut dapat kita lihat pada Gambar 4.25.



**Gambar 4.25** Rata – Rata Nilai Akurasi Klasifikasi Kategori

Sedangkan untuk identifikasi topik diperoleh perhitungan sebagai berikut :



**Gambar 4.26** Rata-Rata Accuracy Identifikasi Topik

Nilai *accuracy* untuk identifikasi topik diperoleh sebesar 0.9778 atau 97,78%. Pada identifikasi topik diperoleh hasil yang maksimal saat pemilihan kata kunci sebesar 20.

Jika kita lihat pada [2], nilai akurasi yang dihasilkan untuk klasifikasi kategori dan identifikasi topik masing – masing adalah 93,84 % dan 97,26%. Sedangkan pada tugas akhir ini, dihasilkan nilai akurasi yang lebih tinggi yaitu 95,56% untuk klasifikasi kategori dan 97,78% untuk identifikasi topik.

Selain hasil diatas, pada Gambar 4.25 dapat kita lihat bahwa terjadi peningkatan rata-rata akurasi sebesar 2,22% pada jumlah kata kunci sebesar 10, hal ini dikarenakan terjadi perubahan jumlah dokumen yang diidentifikasi benar pada 4 kategori yaitu penurunan 10 % pada kategori Regional dan peningkatan 10% pada Nasional, Bisnis Ekonomi dan Pariwisata. Sedangkan pada jumlah kata kunci 10 ke jumlah kata kunci 15 hingga 25, terjadi peningkatan rata-rata akurasi yang konstan yaitu sebesar 1,11%, hal tersebut dikarenakan adanya perubahan jumlah dokumen yang diidentifikasi benar pada 2 kategori yaitu penurunan akurasi 10% pada kategori

Internasional dan peningkatan akurasi 10% pada kategori Metropolitan atau terjadi peningkatan 20% pada kategori Internasional namun terjadi penurunan 10% pada kategori Edukasi.

Selain itu pada Gambar 4.27 terjadi peningkatan yang konstan sebesar 1,11% dari pemilihan kata kunci sejumlah 5-20, sedangkan dari pemilihan jumlah kata kunci 20-40 tidak terjadi peningkatan, melainkan hasil maksimal diperoleh untuk pemilihan jumlah kata kunci sebesar 20.



## LAMPIRAN

### Evaluasi Klasifikasi Kategori dengan 30 Kata Kunci

No	Kategori	Precision	Recall	F-Measure	Accuracy
1	Internasional	1.0000000000	0.8000000000	0.8888888889	1.0000000000
2	Nasional	0.8888888889	0.8888888889	0.8888888889	0.9000000000
3	Regional	0.8888888889	0.8888888889	0.8888888889	0.9000000000
4	Metropolitan	0.8888888889	0.8888888889	0.8888888889	0.9000000000
5	Bisnis Ekonomi	1.0000000000	0.8888888889	0.9411764706	1.0000000000
6	Olahraga	0.8888888889	0.8888888889	0.8888888889	0.9000000000
7	Pariwisata	1.0000000000	0.8888888889	0.9411764706	1.0000000000
8	Sains Teknologi	1.0000000000	0.8888888889	0.9411764706	1.0000000000
9	Edukasi	1.0000000000	0.8888888889	0.9411764706	1.0000000000
	Rata-Rata	0.9506172840	0.8790123457	0.9121278141	0.9555555556

### Evaluasi Klasifikasi Kategori dengan 35 Kata Kunci

No	Kategori	Precision	Recall	F-Measure	Accuracy
1	Internasional	1.0000000000	0.8000000000	0.8888888889	1.0000000000
2	Nasional	0.8888888889	0.8888888889	0.8888888889	0.9000000000
3	Regional	1.0000000000	0.8888888889	0.9411764706	1.0000000000
4	Metropolitan	0.8888888889	0.8888888889	0.8888888889	0.9000000000
5	Bisnis Ekonomi	1.0000000000	0.8888888889	0.9411764706	1.0000000000
6	Olahraga	0.8888888889	0.8888888889	0.8888888889	0.9000000000
7	Pariwisata	1.0000000000	0.8888888889	0.9411764706	1.0000000000
8	Sains Teknologi	0.8888888889	0.8888888889	0.8888888889	0.9000000000
9	Edukasi	1.0000000000	0.8888888889	0.9411764706	1.0000000000
	Rata-Rata	0.9506172840	0.8790123457	0.9121278141	0.9555555556

Evaluasi Klasifikasi Kategori dengan 40 Kata Kunci

No	Kategori	Precision	Recall	F-Measure	Accuracy
1	Internasional	0.888888889	0.800000000	0.8421052632	0.9000000000
2	Nasional	0.888888889	0.888888889	0.888888889	0.9000000000
3	Regional	0.888888889	0.888888889	0.888888889	0.9000000000
4	Metropolitan	1.000000000	0.888888889	0.9411764706	1.0000000000
5	Bisnis Ekonomi	1.000000000	0.888888889	0.9411764706	1.0000000000
6	Olahraga	0.888888889	0.888888889	0.888888889	0.9000000000
7	Pariwisata	1.000000000	0.888888889	0.9411764706	1.0000000000
8	Sains Teknologi	1.000000000	0.888888889	0.9411764706	1.0000000000
9	Edukasi	1.000000000	0.888888889	0.9411764706	1.0000000000
	Rata-Rata	0.9506172840	0.8790123457	0.9127393648	0.9555555556

Akurasi Identifikasi Topik dengan 5 Kata Kunci

No	Kategori	Dokumen										JUMLAH BENAR	JUMLAH SALAH
		1	2	3	4	5	6	7	8	9	10		
1	Internasional	B	B	B	S	B	B	B	B	B	B	9	1
2	Nasional	B	B	B	B	B	B	B	B	B	B	10	0
3	Regional	B	B	B	B	B	B	B	B	S	B	9	1
4	Metropolitan	B	B	B	B	B	B	B	B	B	B	10	0
5	Bisnis Ekonomi	B	B	B	B	S	B	B	B	B	B	9	1
6	Olahraga	B	B	B	B	B	B	B	B	B	S	9	1
7	Pariwisata	B	B	B	B	B	B	B	B	B	B	10	0
8	Sains Teknologi	B	B	B	B	B	B	B	B	B	S	9	1
9	Edukasi	B	B	B	B	B	B	B	B	B	B	10	0
		JUMLAH BENAR										85	5
		JUMLAH SALAH											
		Accuracy										0.9444	

### Akurasi Identifikasi Topik dengan 10 Kata Kunci

No	Kategori	Dokumen										JUMLAH BENAR	JUMLAH SALAH
		1	2	3	4	5	6	7	8	9	10		
1	Internasional	B	B	B	B	B	B	B	B	B	S	9	1
2	Nasional	B	B	B	B	B	B	B	B	B	B	10	0
3	Regional	B	B	B	B	B	B	B	B	S	B	9	1
4	Metropolitan	B	B	B	B	B	B	B	B	B	B	10	0
5	Bisnis Ekonomi	B	B	B	B	B	B	S	B	B	B	9	1
6	Olahraga	B	B	B	B	B	B	B	B	B	B	10	0
7	Pariwisata	B	B	B	B	B	B	B	B	B	B	10	0
8	Sains Teknologi	B	B	B	S	B	B	B	B	B	B	9	1
9	Edukasi	B	B	B	B	B	B	B	B	B	B	10	0
		JUMLAH BENAR										86	4
		JUMLAH SALAH											
		Accuracy										0.9556	

### Akurasi Identifikasi Topik dengan 15 Kata Kunci

No	Kategori	Dokumen										JUMLAH BENAR	JUMLAH SALAH
		1	2	3	4	5	6	7	8	9	10		
1	Internasional	B	B	B	B	B	B	B	B	B	S	9	1
2	Nasional	B	B	B	B	B	B	B	B	B	B	10	0
3	Regional	B	B	B	B	B	B	B	B	B	B	10	0
4	Metropolitan	B	B	B	B	B	B	B	B	B	B	10	0
5	Bisnis Ekonomi	B	B	B	B	B	B	B	B	B	S	9	1
6	Olahraga	B	B	B	B	B	B	B	B	B	B	10	0
7	Pariwisata	B	B	B	B	B	B	B	B	B	B	10	0
8	Sains Teknologi	B	B	B	B	B	B	B	B	B	S	9	1
9	Edukasi	B	B	B	B	B	B	B	B	B	B	10	0
		JUMLAH BENAR										87	3
		JUMLAH SALAH											
		Accuracy										0.9667	

Akurasi Identifikasi Topik dengan 20 Kata Kunci

No	Kategori	Dokumen										JUMLAH BENAR	JUMLAH SALAH
		1	2	3	4	5	6	7	8	9	10		
1	Internasional	B	B	B	B	B	B	B	B	B	B	10	0
2	Nasional	B	B	B	B	B	B	B	B	B	B	10	0
3	Regional	B	B	B	B	B	B	B	B	S	B	9	1
4	Metropolitan	B	B	B	B	B	B	B	B	B	B	10	0
5	Bisnis Ekonomi	B	B	B	B	B	B	B	B	B	B	10	0
6	Olahraga	B	B	B	B	B	B	B	B	B	S	9	1
7	Pariwisata	B	B	B	B	B	B	B	B	B	B	10	0
8	Sains Teknologi	B	B	B	B	B	B	B	B	B	B	10	0
9	Edukasi	B	B	B	B	B	B	B	B	B	B	10	0
		JUMLAH BENAR										88	2
		JUMLAH SALAH											
		Accuracy										0.9778	

Akurasi Identifikasi Topik dengan 25 Kata Kunci

No	Kategori	Dokumen										JUMLAH BENAR	JUMLAH SALAH
		1	2	3	4	5	6	7	8	9	10		
1	Internasional	B	B	B	B	B	B	B	B	B	B	10	0
2	Nasional	B	B	B	B	B	B	B	B	B	B	10	0
3	Regional	B	B	B	B	B	B	B	B	B	B	10	0
4	Metropolitan	B	B	S	B	B	B	B	B	B	B	9	1
5	Bisnis Ekonomi	B	B	B	B	B	B	B	B	B	B	10	0
6	Olahraga	B	B	B	B	B	B	B	B	B	S	9	1
7	Pariwisata	B	B	B	B	B	B	B	B	B	B	10	0
8	Sains Teknologi	B	B	B	B	B	B	B	B	B	B	10	0
9	Edukasi	B	B	B	B	B	B	B	B	B	B	10	0
		JUMLAH BENAR										88	2
		JUMLAH SALAH											
		Accuracy										0.9778	



### Akurasi Identifikasi Topik dengan 30 Kata Kunci

No	Kategori	Dokumen										JUMLAH BENAR	JUMLAH SALAH
		1	2	3	4	5	6	7	8	9	10		
1	Internasional	B	B	B	B	B	B	B	B	B	B	10	0
2	Nasional	B	B	B	B	B	B	B	B	B	B	10	0
3	Regional	B	B	B	B	B	B	B	B	S	B	9	1
4	Metropolitan	B	B	B	B	B	B	B	B	B	B	10	0
5	Bisnis Ekonomi	B	B	B	B	B	B	B	B	B	B	10	0
6	Olahraga	B	B	B	B	S	B	B	B	B	B	9	1
7	Pariwisata	B	B	B	B	B	B	B	B	B	B	10	0
8	Sains Teknologi	B	B	B	B	B	B	B	B	B	B	10	0
9	Edukasi	B	B	B	B	B	B	B	B	B	B	10	0
		JUMLAH BENAR										88	2
		JUMLAH SALAH											
		Accuracy										0.9778	

### Akurasi Identifikasi Topik dengan 35 Kata Kunci

No	Kategori	Dokumen										JUMLAH BENAR	JUMLAH SALAH
		1	2	3	4	5	6	7	8	9	10		
1	Internasional	B	B	B	B	B	B	B	B	B	B	10	0
2	Nasional	B	B	B	B	B	B	B	B	B	B	10	0
3	Regional	B	B	B	B	B	B	B	B	S	B	9	1
4	Metropolitan	B	B	B	B	B	B	B	B	B	B	10	0
5	Bisnis Ekonomi	B	B	B	B	B	B	B	B	B	B	10	0
6	Olahraga	B	B	B	B	B	B	B	B	B	S	9	1
7	Pariwisata	B	B	B	B	B	B	B	B	B	B	10	0
8	Sains Teknologi	B	B	B	B	B	B	B	B	B	B	10	0
9	Edukasi	B	B	B	B	B	B	B	B	B	B	10	0
		JUMLAH BENAR										88	2
		JUMLAH SALAH											
		Accuracy										0.9778	

Akurasi Identifikasi Topik dengan 40 Kata Kunci

No	Kategori	Dokumen										JUMLAH BENAR	JUMLAH SALAH
		1	2	3	4	5	6	7	8	9	10		
1	Internasional	B	B	B	B	B	B	B	B	B	B	10	0
2	Nasional	B	B	B	B	B	B	B	B	B	B	10	0
3	Regional	B	B	B	B	B	B	B	B	B	B	10	0
4	Metropolitan	B	B	B	B	B	B	B	B	B	S	9	1
5	Bisnis Ekonomi	B	B	B	B	B	B	B	B	B	B	10	0
6	Olahraga	B	B	B	B	B	B	B	B	B	S	9	1
7	Pariwisata	B	B	B	B	B	B	B	B	B	B	10	0
8	Sains Teknologi	B	B	B	B	B	B	B	B	B	B	10	0
9	Edukasi	B	B	B	B	B	B	B	B	B	B	10	0
		JUMLAH BENAR										88	2
		JUMLAH SALAH											
		Accuracy										0.9778	

### **Contoh Corpus Bereksistensi .news (Kategori Sains dan Teknologi)**

Selasa, 6 April 2011

Investasi

1

Perusahaan AS Akan Jajaki Pasar RI

JAKARTA, KOMPAS.com - Dalam rangka mewujudkan Comprehensive Partnership antara AS dan Indonesia, yang telah ditandatangani oleh Presiden Barack Obama dan Yudhoyono, pada kunjungan ke Indonesia bulan November 2010 silam, kedua negara telah setuju untuk meningkatkan volume investasi dan bisnis. Duta Besar Amerika Serikat (AS) untuk Indonesia Scot A Marciel, di Jakarta, Selasa (29/3/2011), menyebutkan AS akan meningkatkan volume investasi di Indonesia, dengan tujuan mendongkrak posisi lebih dari posisi saat ini yang menjadi investor terbesar ke-3 di Indonesia. "Presiden (AS dan RI) bersama meluncurkan sebuah partnership (perjanjian kerjasama) yang baru November lalu di Jakarta, untuk meningkatkan kooperasi dan kerja bersama di seluruh area. Dan, satu area yang ingin kita tingkatkan yaitu bisnis," jelas Scot. Salah satu wujud nyata dari kerjasama yang baru ini, 20 delegasi perusahaan AS akan selenggarakan diskusi mengenai peluang investasi di Indonesia, hari ini, Selasa (5/4/2011). Perusahaan yang bergerak di bidang pertanian dan pangan olahan mendominasi delegasi tersebut, diantaranya Monsanto, Case New Holland, Mirasco, SunOpta, Mirasco Inc, Datepac Inc, dan Commercial Creamery Company. Tidak luput pula perusahaan pangan dengan sertifikasi halal, seperti Midamar Corporation, Salwa Foods, dan Islamic Services of America, yang menyediakan layanan edukasi dan sertifikasi halal di AS. Scot menuturkan, dengan meningkatkan volume bisnis maka akan menciptakan tingkat kesejahteraan yang lebih tinggi dan lebih banyak pekerjaan di kedua negara. Delegasi 20 perusahaan ini dipimpin oleh Under Secretary of Commerce for International Trade Francisco J Sanchez dan Under Secretary of Agriculture Michael Scuse, yang akan berada di Indonesia hingga 6 April, esok hari.

**Contoh Corpus Bereksistensi .news (Kategori Nasional)**

Jumat, 8 Januari 2016

Politik

1

Jika Ketua DPR Dilantik 11 Januari, PDI-P Tidak Akan Hadir

JAKARTA, KOMPAS.com- Sekjen PDI Perjuangan Hasto Kristiyanto memastikan bahwa seluruh anggota Fraksi PDI-P tidak akan menghadiri pelantikan Ketua DPR RI jika dilakukan pada 11 Januari 2016 mendatang. Sebab, kata Hasto, pada tanggal tersebut PDI-P akan menggelar rapat kerja nasional di JIExpo Kemayoran, Jakarta. "Pada tanggal tersebut, seluruh Fraksi PDI-P menjadi peserta aktif di dalam Rakernas I," kata Hasto di Jakarta, Jumat (8/1/2016). Hasto mengaku sudah meminta Fraksi PDI-P untuk mengirimkan surat kepada pimpinan DPR agar pelantikan tersebut bisa ditunda. Apalagi, lanjut Hasto, sampai saat ini legalitas Partai Golkar yang kadernya akan menempati posisi Ketua DPR juga belum jelas. Baik kubu Aburizal dan kubu Agung Laksono sama-sama mengajukan calonnya masing-masing. Aburizal mengusulkan Ade Komarudin sementara Agung mengusulkan Agus Gumiwang Kartasasmita. "Agenda penting dan strategis seperti itu, seharusnya dapat dilakukan setelah ketentuan legalitas dan tatib DPR RI terpenuhi," ucap Hasto. Hasto membantah agenda rakernas sengaja dijadwalkan bertabrakan dengan pelantikan karena PDI-P tidak setuju Ketua DPR berasal dari Golkar.

## **BAB VI PENUTUP**

### **6.1 Kesimpulan**

1. Program telah selesai dibuat menggunakan Algoritma Peningkatan Porter Stemmer dan Likelihood serta diuji mampu melakukan proses klasifikasi kategori serta identifikasi topik pada artikel berita berbahasa Indonesia
2. Berdasarkan hasil uji coba, proses klasifikasi kategori mendapatkan hasil yang optimal saat menggunakan jumlah kata kunci sebanyak 25, sedangkan untuk identifikasi topik diperoleh hasil yang maksimal dengan jumlah kata kunci sebanyak 20.
3. Nilai *accuracy* untuk klasifikasi kategori diperoleh sebesar 95,56 %, sedangkan untuk identifikasi topik sebesar 97,78 %. Kedua nilai tersebut tampak lebih baik daripada nilai *accuracy* yang dihasilkan pada penelitian sebelumnya.

### **6.2 Saran**

Sebagai evaluasi dan pengembangan selanjutnya diharapkan dapat dilakukan beberapa saran berikut :

1. Riset lebih lanjut dalam hal *running time*, karena membutuhkan waktu yang cukup lama saat identifikasi topik.
2. Program disediakan fungsi *download* dokumen agar secara otomatis disimpan mengikuti format Corpus.



## DAFTAR PUSTAKA

- [1] Bracewell D, Jiajun Yan, Fuji Ren dan Shingo Kuroiwa.2009. "Category Classification and Topic Discovery of Japanese and English News Article," **Electronic Notes in Theoretical Computer Science** 225(2009) 51-65.
- [2] Fuddoly, Aini Rachmania Kusumaagama, Agus Zainal Arifin.2011. "Klasifikasi Kategori dan Identifikasi Topik pada Artikel Berita Bahasa Indonesia," ITS.Surabaya
- [3] Karaa,Wahiba Ben Abdessalem, "A New Stemmer to Improve Information Retrieval," *International Journal of Network Security & Its Applications (IJNSA)*, Vol.5, No.4, July 2013
- [4] DR.E. Garcia,2006. **The Classic Vector Space Model**,  
<URL:<http://www.miislita.com/term-vector/term-vector-3.html>>
- [5] Wiguna ,Putu Bagus Susastra, Bimo Sunarfri Hantono."Peningkatan Algoritma Porter Stemmer Bahasa Indonesia berdasarkan Metode Morfologi dengan Mengaplikasikan 2 Tingkat Morfologi dan Aturan Kombinasi Awalan dan Akhiran," JNTETI, Vol.2, No.2,2013
- [6] Augusta Ledy, "Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia", Konferensi Nasional Sistem dan Informatika 2009; Bali, November 14, 2009

- [7] Nadirman Firas, 2006. **Sistem Temu-Kembali Informasi Dengan Metode Vector Space Model Pada Pencarian File Dokumen Berbasis Teks**,  
<URL:<http://kabulkurniawan.web.ugm.ac.id/wp-content/uploads/SKRIPSI.pdf>>
- [8]<URL:<https://dataq.wordpress.com/2013/06/16/perbedaan-precision-recall-accuracy/>>



## BIODATA PENULIS



Penulis memiliki nama lengkap Devi Andriyani atau biasa dipanggil Devi, lahir di Probolinggo, 19 Juli 1993. Terlahir sebagai anak pertama dari 2 bersaudara. Sejak usia enam tahun, penulis mulai bersekolah formal di SDN Sumber Kedawung II (2000-2006), SMP Negeri 1 Leces (2006-2009), SMA Taruna Dra. Zulaiha (2009-2012). Setelah lulus SMA, penulis melanjutkan studi ke jenjang S1 di jurusan Matematika

Institut Teknologi Sepuluh Nopember tahun 2012.

Di jurusan Matematika, penulis mengambil bidang minat Ilmu Komputer. Selama masa perkuliahan, penulis pernah mengikuti penelitian bidang Pengabdian kepada Masyarakat bersama Dosen Alvida Mustika Rukmi, S.Si, M.Si, pemenang kompetisi tingkat nasional *Business Plan* FEW 2015 di Jakarta, serta finalis dalam beberapa kompetisi nasional yang lain. Penulis memiliki *passion* yang baik di bidang Ilmu Komputer terutama *data mining*. Saat ini, penulis sedang menjabat sebagai *Finance Director* di CV. Indi Global yang merupakan perusahaan IT Consultant di Surabaya.

Adapun mengenai informasi lebih lanjut atau ingin berdiskusi mengenai tugas akhir ini dapat ditujukan ke email penulis [devi12@mhs.matematika.its.ac.id](mailto:devi12@mhs.matematika.its.ac.id) atau [deviand67@gmail.com](mailto:deviand67@gmail.com).