

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

**KLASIFIKASI DOKUMEN BAHASA JAWA  
MENGGUNAKAN METODE NAÏVE BAYESIAN**

Skripsi

Diajukan Untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

Program Studi Teknik Informatika



Oleh

Y. Violya Yosnaningsih

085314098

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA**

**2015**

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

# JAVANESE DOCUMENT CLASSIFICATION USING NAÏVE BAYESIAN ALGORITMS

A Thesis

Presented as Partial Fulfillment of The Requirements

To Obtain *Sarjana Komputer* Degree

in Informatics Engineering Study Program



By

Y. Violya Yosnaningsih

085314098

**INFORMATICS ENGINEERING STUDY PROGRAM  
DEPARTMENT OF INFORMATICS ENGINEERING  
FACULTY OF SCIENCE AND TECHNOLOGY  
SANATA DHARMA UNIVERSITY  
YOGYAKARTA  
2015**

HALAMAN PERSETUJUAN

SKRIPSI

KLASIFIKASI DOKUMEN BAHASA JAWA  
MENGGUNAKAN METODE NAÏVE BAYESIAN



HALAMAN PENGESAHAN

SKRIPSI

KLASIFIKASI DOKUMEN BAHASA JAWA  
MENGGUNAKAN METODE NAÏVE BAYESIAN

Dipersiapkan dan ditulis oleh :

Y. Violya Yosnaningsih

NIM : 085314098

Telah dipertahankan di depan Panitia Penguji

Pada tanggal 16 Februari 2015

dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

Nama Lengkap

Ketua Ridowati Gunawan, S.Kom., M.T.

Sekretaris

Alb. Agung Hadhiatma, S.T., M.T.

Anggota

Sri Hartati Wijono, S.Si., M.Kom.

Tanda Tangan

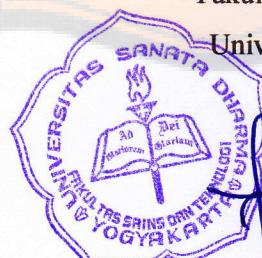


Yogyakarta, 16 Februari 2015

Fakultas Sains dan Teknologi

Universitas Sanata Dharma

Dekan





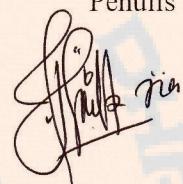
( Paulina Heruningsih Prima Rosa, S.Si., M.Sc.)

**PERNYATAAN KEASLIAN KARYA**

Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka sebagaimana layaknya karya ilmiah.

Yogyakarta, Februari 2015

Penulis



Y. Violya Yosnaningsih

**HALAMAN MOTO**

*Apa yang kau alami kini mungkin tak dapat engkau mengerti  
Satu hal tanamkan dihati, indah semua yang Tuhan beri  
Tangan Tuhan sedang merenda, suatu karya yang agung mulia  
Saatnya kan tiba nanti kau lihat pelangi kasih-Nya*

*“semua indah pada waktuNya.*

*Nya besar, bukan nya kecil”*

*-MoNdhan-*

**HALAMAN PERSEMBAHAN**

Tugas akhir ini aku persembahkan untuk :

Tuhan Yesus

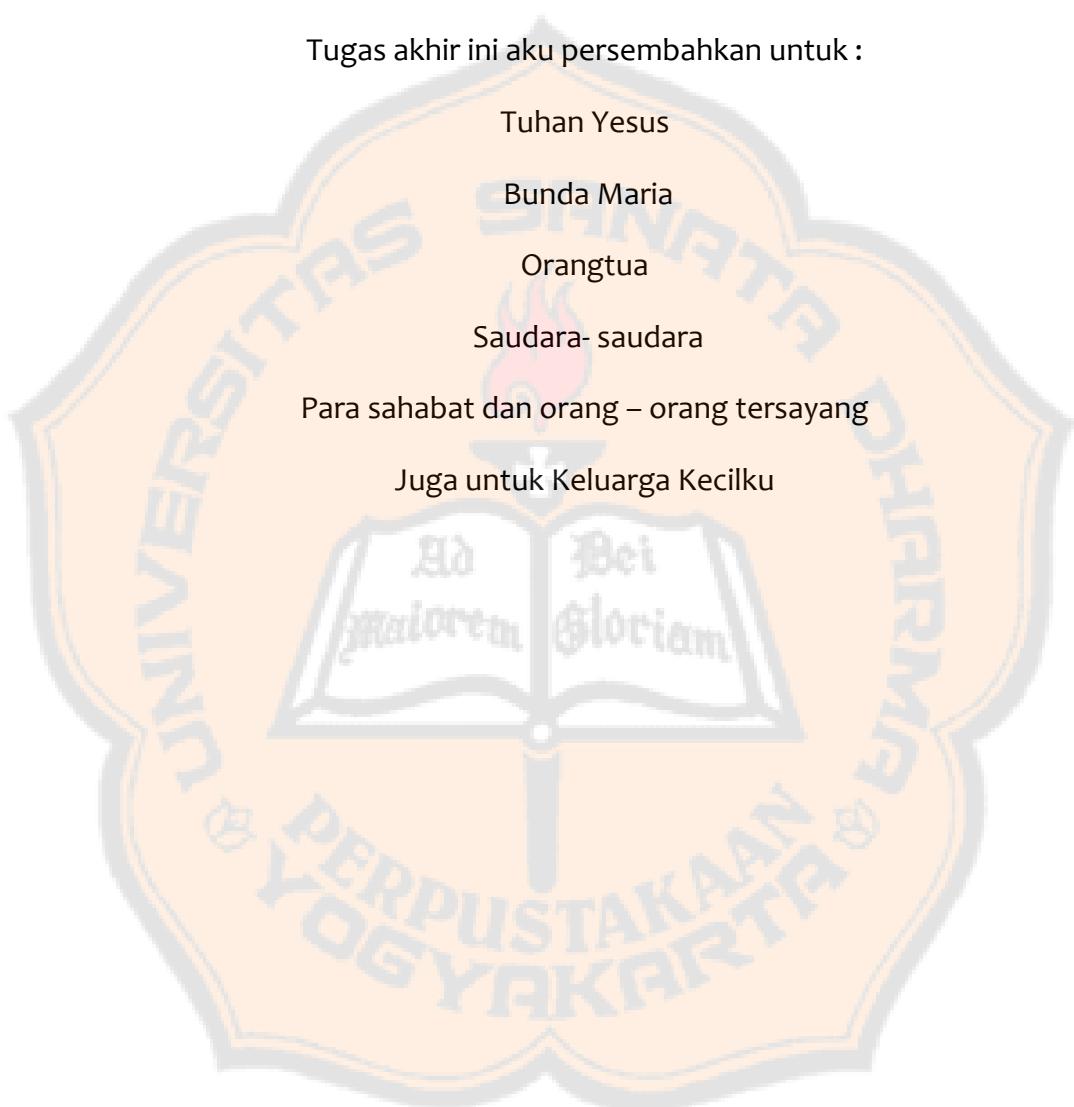
Bunda Maria

Orangtua

Saudara- saudara

Para sahabat dan orang – orang tersayang

Juga untuk Keluarga Kecilku



## ABSTRAK

Penelitian ini digunakan dalam klasifikasi bahasa Jawa. Hasil yang dikeluarkan berupa informasi mengenai kategori dokumen, yaitu ekonomi, kesehatan, pendidikan atau politik. Proses awal, yaitu menginputkan dokumen yang akan digunakan sebagai data *training* ke dalam sistem, berdasarkan kategori yang telah diketahui. Kemudian dilakukan proses *pre-processing* berupa *tokenisasi* (pemenggalan kata dan penghapusan tanda baca dan karakter), *case folding* (mengubah kata kedalam huruf kecil), *stopword* (penghapusan kata yang dianggap tidak penting), *stemming* (pengembalian kata kebentuk dasar), dan menghitung *term frequency*. Setelah menghasilkan kata unik, diolah untuk dihitung W (bobot kata) dan *Laplace Smoothing* dan digunakan dalam proses klasifikasi. Pada data *testing*, dokumen juga melewati proses *pre-processing*. Dari kedua data, dilakukan proses *matching*, yaitu mendapatkan kata – kata yang sama dari data *training* dan *testing*. Jika data *matching* telah diperoleh, maka akan digunakan untuk menjalankan proses klasifikasi menggunakan metode Naïve Bayesian. Pada penelitian ini dilakukan pengujian *cross validation* kemudian dilakukan uji presisi. Data yang digunakan sebanyak 40 dokumen. Tingkat akurasi untuk 3 *fold* mencapai 69,78 %, untuk 5 *fold* mencapai 77,5%.

**Kata kunci :** klasifikasi dokumen bahasa Jawa, *Naïve Bayesian*, pemerolehan informasi

## ABSTRACT

This research is used for javanese classification. The output are information about document category, there are economic, health, education, or politic. The first process is inputting document that will be used for training data into the system, based on known category. Then the process continue with preprocessing for make model of documents collection that inputted like tokenizing (slice of words and erasing punctuation and character), case folding (change word into lower case), stopwords (erasing unimportant words), stemming (returning the word into first form), and counting term frequency. After producing unique word and will processed to count W (word weight) and Laplace smoothing and used for classification process. At testing data, documents also need preprocessing. From both process, will be doing matching process, that is accuiring the same words from training data and testing. If matching data is done, then it will be used for classification process using Naïve Bayesian method. At this research will be using cross validation. Data that is used are 40 documents. Accuration for 3 fold reach 69,78%, and for 5 fold reach 77,5%.

**Keywords** : Javanese languange classification, Naïve Bayesian, Information Retrieval

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

## LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPERLUAN AKADEMIS

Yang bertanda tangan di bawah ini, saya mahasiswa Universitas Sanata Dharma :

Nama : Y. Violya Yosnaningsih

Nomor Mahasiswa : 085314098

Demi mengembangkan ilmu pengetahuan, saya memberikan kepada perpustakaan Universitas Sanata Dharma karya ilmiah saya yang berjudul:

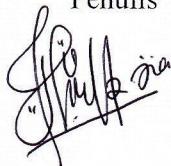
### KLASIFIKASI DOKUMEN BAHASA JAWA MENGGUNAKAN METODE NAÏVE BAYESIAN

Beserta perangkat yang diperlukan. Dengan demikian saya memberikan kepada Perpustakaan Universitas Sanata Dharma hak untuk menyimpan, mengalihkan dalam bentuk media lain, mengelolanya dalam bentuk pangkalan data, mendistribusikan secara terbatas, dan mempublikasikannya di Internet atau media lain untuk kepentingan akademis tanpa perlu meminta izin dari saya maupun memberikan royalti kepada saya selama tetap mencantumkan nama saya sebagai penulis.

Demikian pernyataan ini saya buat dengan sebenarnya.

Yogyakarta, Februari 2015

Penulis



Y. Violya Yosnaningsih

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa atas kasih dan penyertaanNyalah sehingga penulis dapat menyelesaikan penyusunan skripsi dengan judul "**Klasifikasi Dokumen Bahasa Jawa Menggunakan Metode Naïve Bayesian**". Penulisan skripsi ini ditujukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer Universitas Sanata Dharma Yogyakarta.

Penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, dan peran berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terimakasih kepada pihak-pihak berikut:

1. Tuhan Yesus Kristus dan Bunda Maria yang selalu membimbing dan menuntun untuk menyelesaikan tugas skripsi ini.
2. Ibu Paulina Heruningsih Prima Rosa, S.Si., M.Sc selaku Dekan Fakultas Sains dan Teknologi Universitas Sanata Dharma.
3. Ibu Ridowati Gunawan, S.Kom., M.T. selaku Kepala Program Studi Teknik Informatika sekaligus selaku dosen penguji.
4. Ibu Sri Hartati Wijono, S.Si., M.Kom. selaku dosen pembimbing skripsi sekaligus dosen pembimbing akademik yang telah meluangkan banyak waktu untuk membimbing dan memotivasi penulis untuk terus membaca dan belajar.
5. Bapak Alb. Agung Hadhiatma, S.T., M.T. selaku dosen penguji.
6. Seluruh staff pengajar dan karyawan Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Sanata Dharma.
7. Kedua orang tua saya, Bapak Suparno dan Ibu Susana Sukinem, adik Yohanes Seffan Handana dan adik Laurensius Edo Gita Ardana yang selalu mendoakan, menasehati, dan memberi semangat dalam mengerjakan tugas akhir ini.

## PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

8. Bapak Sukiman, Ibu Ning Rahayu, adik Carollina Swastika Lisdiyani, adik Juliaus Bagas Triatmoko, adik Ignatius Rikat Wijanarko, adik Alif Farhan yang terus memberikan dukungan dan semangat, serta canda tawa sehingga dapat menyelesaikan skripsi ini.
9. Keluarga kecilku dengan Suami tercinta Yustinus Euzhan Yogatama, serta malaikat kecilku Clareta Angela Widya Palupi yang selalu memberikan kasih sayang dan semangat dalam mengerjakan skripsi ini.
10. Sahabat-sahabatku, makk Wikk (Veverly Widyastuti Palinoan), Andrea Pratama, Tri Suwanta, Nenek (Maria Kristilia) atas semua dukungan dan semangat serta canda tawa dalam penyelesaian skripsi ini.
11. Semua pihak yang telah membantu penyelesaian skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari masih banyak kekurangan dalam menyusun skripsi ini, namun penulis tetap berharap skripsi ini bermanfaat bagi pengembangan ilmu pengetahuan.

Yogyakarta, Maret 2015

Penulis

Y. Violya Yosnaningsih

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN .....	iv
PERNYATAAN KEASLIAN KARYA .....	v
HALAMAN MOTTO .....	vi
HALAMAN PERSEMBAHAN .....	vii
ABSTRAK .....	viii
ABSTRACT.....	ix
LEMBAR PENYATAAN PERSETUJUAN.....	x
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL.....	xviii
DAFTAR LIST CODE .....	xix
BAB I PENDAHULUAN	
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian .....	4
1.5. Metodologi Penelitian .....	4
1.6. Sistematika Penulisan .....	5

## BAB II LANDASAN TEORI

2.1 <i>Information Retrieval</i> .....	7
2.2 <i>Pre-Processing</i> .....	8
2.2.1 Tokenisasi dan <i>case folding</i> .....	8
2.2.2 <i>Stopword</i> .....	9
2.2.3 <i>Stemming</i> .....	10
2.2.4 TF-IDF ( <i>Term Frequency Inverse Document Frequency</i> )..	14

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

2.3	Klasifikasi Teks.....	15
2.3.1	Metode Naïve Bayesian .....	15
2.4	Evaluasi Information Retrieval .....	18
2.4.1	<i>K-fold Cross Validation</i> .....	18
2.4.2	<i>Precision</i> .....	19
 BAB III ANALIS DAN PERANCANGAN .....		
3.1	Gambaran Sistem .....	20
3.2	Gambaran Proses pada Sistem .....	22
3.3	Analisa Kebutuhan .....	24
3.3.1	Definisi Aktor.....	24
3.3.2	Use Case .....	24
3.3.3	Narasi Use Case .....	25
3.4	Perancangan Model Penyimpanan Data.....	27
3.5	Diagram Konteks .....	28
3.6	Diagram Aktifitas.....	28
3.6.1	Diagram Aktifitas <i>Pre- Processing</i> .....	28
3.6.2	Diagram Aktifitas Klasifikasi.....	29
3.6.3	Diagram Aktifitas <i>Trainer</i> .....	30
3.7	Perancangan Diagram Sekuensial .....	31
3.7.1	Diagram Sekuensial <i>Preprocessing</i> .....	31
3.7.2	Diagram Sekuensial Klasifikasi .....	32
3.8	Cara pengujian dan Analisis Hasil .....	33
3.9	Contoh Langkah Pengerjaan .....	36
3.9.1	Dokumen .....	36
3.9.2	<i>Preprocessing</i> .....	37
3.9.3	Klasifikasi.....	38
3.10	Perancangan Antarmuka ( <i>Interface</i> ) .....	42
3.10.1	Menu Utama .....	42
3.10.2	Menu Klasifikasi Dokumen .....	42
3.10.3	<i>Menu Pre-Processing</i> .....	42

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

3.10.4 Menu <i>Trainer</i> .....	43
----------------------------------	----

## BAB IV IMPLEMENTASI

4.1 Spesifikasi <i>Software</i> dan <i>Hardware</i> .....	44
4.2 Implementasi Antarmuka .....	45
4.2.1 Antarmuka MainFrame .....	45
4.2.2 Antarmuka Klasifikasi.....	45
4.2.3 Antarmuka Preprocessing .....	46
4.2.4 Antarmuka Trainer .....	46
4.3 Implementasi Preprocessing .....	47
4.3.1 Implementasi Membaca File Dokumen .....	47
4.3.2 Implementasi Tokenisasi dan Case Folding.....	47
4.3.3 Implementasi Stopwords .....	48
4.3.4 Implementasi Stemming.....	48
4.4 Implementasi Klasifikasi.....	54
4.5 Implementasi Trainer .....	57

## BAB V HASIL DAN PEMBAHASAN

5.1 Hasil Pengujian .....	60
5.1.1 Hasil Pengujian menggunakan <i>Feature tfidf(W)</i> .....	62
1) 3-Fold menggunakan <i>Feature tfidf(W)</i> .....	62
2) 5-Fold menggunakan <i>Feature tfidf(W)</i> .....	63
5.1.2 Hasil Pengujian menggunakan <i>Feature tf</i> .....	64
1) 3-Fold menggunakan <i>Feature tf</i> .....	64
2) 5-Fold menggunakan <i>Feature tf</i> .....	65
5.2 Analisis Hasil .....	66

## BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan .....	68
6.2 Saran.....	68

DAFTAR PUSTAKA .....	69
LAMPIRAN .....	71



## **DAFTAR GAMBAR**

Gambar 2.1. Gambaran umum IR .....	7
Gambar 2.2. Tahapan <i>Pre-Processing</i> .....	8
Gambar 2.3. Proses Tokenisasi dan <i>Case Folding</i> .....	9
Gambar 2.4. Proses <i>Stopword</i> .....	9
Gambar 3.1. Skema Proses Klasifikasi .....	22
Gambar 3.2. Diagram Use Case .....	25
Gambar 3.3. Diagram Konteks .....	28
Gambar 3.4. Diagram Aktivitas <i>Pre Processing</i> .....	28
Gambar 3.5. Diagram Aktivitas Klasifikasi.....	29
Gambar 3.6. Diagram Aktivitas <i>Trainer</i> .....	30
Gambar 3.7. Diagram Sekuensial <i>Pre Processing</i> .....	31
Gambar 3.8. Diagram Sekuensial Klasifikasi .....	32
Gambar 3.9. Desain Menu Utama.....	42
Gambar 3.10. Desain Klasifikasi .....	42
Gambar 3.11. Desain <i>Pre-processing</i> .....	43
Gambar 3.12. Desain <i>Trainer</i> .....	43
Gambar 4.1. Antarmuka <i>MainFrame</i> .....	45
Gambar 4.2. Antarmuka Klasifikasi .....	45
Gambar 4.3. Antarmuka <i>Pre-processing</i> .....	46
Gambar 4.5. Antarmuka <i>Trainer</i> .....	47

## DAFTAR TABEL

Tabel 2.1. Aturan untuk <i>suffix</i> .....	10
Tabel 2.2. Aturan untuk <i>prefix</i> .....	11
Tabel 2.3. Aturan untuk <i>infix</i> .....	12
Tabel 3.1. Narasi Use Case Klasifikasi.....	25
Tabel 3.2. Narasi Use Case <i>Preprocessing</i> .....	25
Tabel 3.3. Narasi Use Case <i>Trainer</i> .....	26
Tabel 3.4. Data <i>training</i> dan <i>testing</i> .....	36
Tabel 3.5. Contoh Perhitungan Manual .....	39
Tabel 3.6. Perhitungan pada <i>Matching</i> .....	40
Tabel 3.7. Perhitungan <i>Laplace Smoothing</i> .....	41
Tabel 5.1. Daftar Seluruh Dokumen .....	60
Tabel 5.2. Pemetaan Data untuk <i>3-fold</i> .....	61
Tabel 5.3. Fungsi Data <i>3 fold</i> .....	61
Tabel 5.4. Pemetaan Data untuk <i>5-fold</i> .....	61
Tabel 5.5. Fungsi Data <i>5 fold</i> .....	62
Tabel 5.6. Hasil Klasifikasi <i>3 fold</i> ( <i>feature W</i> ) .....	62
Tabel 5.7. Akurasi <i>3 fold</i> ( <i>feature W</i> ) .....	63
Tabel 5.8. Hasil Klasifikasi <i>5 fold</i> ( <i>feature W</i> ) .....	63
Tabel 5.9. Akurasi <i>5 fold</i> ( <i>feature W</i> ) .....	64
Tabel 5.10. Hasil Klasifikasi <i>3 fold</i> ( <i>feature tf</i> ) .....	64
Tabel 5.11. Akurasi <i>3 fold</i> ( <i>feature tf</i> ) .....	65
Tabel 5.12. Hasil Klasifikasi <i>5 fold</i> ( <i>feature tf</i> ) .....	65
Tabel 5.13. Akurasi <i>3 fold</i> ( <i>feature tf</i> ) .....	66
Tabel 5.14. Akurasi Klasifikasi <i>feature tf</i> dan <i>tf-idf</i> .....	66

## DAFTAR LIST CODE

List Code 4.3.1 Membaca File .....	47
List Code 4.3.2 Tokenisasi dan <i>case folding</i> .....	48
List Code 4.3.3 <i>Stopwords</i> .....	48
List Code 4.3.4a <i>Stemming</i> .....	49
List Code 4.3.4b <i>Stemming Perl</i> .....	54
List Code 4.4.1 Membaca hasil <i>training</i> .....	55
List Code 4.4.2 Proses <i>preprocessing</i> pada data <i>testing</i> .....	56
List Code 4.4.3 <i>Matching</i> .....	56
List Code 4.4.4. Memangkatkan <i>Laplace Smoothing</i> dengan <i>tf testing</i> .....	56
<i>List Code 4.4.5.</i> Mengalikan <i>prior probabilities</i> dengan <i>Laplace Smoothing</i> .....	57
List Code 4.4.6 Membandingkan hasil perkalian <i>prior probabilities</i> .....	57
List Code 4.5.1 <i>Trainer</i> .....	59

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Penggunaan komputer tidak dapat dipisahkan dari kehidupan manusia berbagai bidang, baik dibidang pendidikan, bisnis ataupun penelitian. Pemanfaatan komputer tersebut antara lain untuk mengolah dan menyimpan berbagai jenis dokumen dalam bentuk digital. Penyimpanan yang terus menerus dalam bentuk digital akan menimbulkan penumpukan informasi, sehingga diperlukan penyaringan atau klasifikasi terhadap informasi yang ada.

Dokumen berbahasa Jawa semakin banyak ditulis dalam bentuk digital. Namun tidak semua orang mengerti isi dari dokumen tersebut. Diperlukan waktu yang lama jika harus membaca satu per satu dokumen untuk dapat mengetahui termasuk golongan kategori/kelas yang mana dokumen tersebut. Tentu akan sulit untuk mengolah dan menentukan suatu artikel termasuk dalam kelas yang mana jika terdapat ratusan artikel atau dokumen.

Penyaringan atau klasifikasi diperlukan untuk memilah dokumen, baik dokumen berupa teks, gambar, video ataupun suara. Memerlukan waktu yang lama jika harus mengolah atau menganalisa satu per satu apalagi dengan jumlah dokumen yang sangat besar, maka akan lebih mudah

mencari suatu dokumen apabila dokumen tersebut terorganisir dan dikelompokkan sesuai dengan kategorinya.

Klasifikasi sendiri memiliki tujuan untuk memisahkan dokumen – dokumen dalam beberapa kelas atau kategori dengan menilai kemiripan antar dokumen. Berdasarkan kemiripan tersebut, maka pembaca akan dapat menemukan informasi yang dibutuhkan.

Ada banyak metode klasifikasi dokumen, salah satunya menggunakan metode Naïve Bayesian, dimana dalam prosesnya, akan memeriksa kesamaan kata yang muncul dalam setiap dokumen, serta memperhitungkan probabilitas kata yang muncul.

Berdasarkan penelitian mengenai Sistem Klasifikasi Surat Masuk menggunakan Multinomial Naïve Bayes (Hanopo, 2007), yang menggunakan *term frequency* dalam penerapannya, didapatkan kesimpulan bahwa hasil pengujian menggunakan *5-fold cross validation* memperoleh akurasi rata-rata benar sebesar 83% dan salah 16%, sedangkan pada *3-fold cross validation* memperoleh akurasi rata-rata benar 79% dan salah 20%, maka penulis tertarik untuk mengklasifikasikan dokumen bahasa Jawa dengan menggunakan *feature bobot kata (tf-idf)*.

Algoritma Multinomial Naïve Bayes adalah pengembangan dari algoritma Naïve Bayes yang memiliki keunggulan dalam memproses teks. Naive Bayes (Witten & Frank, 2005) yaitu salah satu teknik klasifikasi yang banyak digunakan untuk klasifikasi teks karena metode ini sangat cepat dan cukup akurat.

Naïve Bayes Classifier (NBC) memiliki beberapa kelebihan antara lain, sederhana, cepat dan berakurasi tinggi. Metode NBC untuk klasifikasi atau kategorisasi teks menggunakan atribut kata yang muncul dalam suatu dokumen sebagai dasar klasifikasinya (Hamzah, 2012).

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, rumusan masalah yang didapat adalah

1. Bagaimana ketepatan metode Naïve Bayesian dalam pengklasifikasian dokumen bahasa Jawa.

## 1.3. Batasan Masalah

Batasan masalah dalam pembuatan sistem ini adalah sebagai berikut :

1. Pengklasifikasian dokumen hanya dilakukan pada dokumen berbahasa Jawa dengan berekstensi .txt.
2. Jumlah data dokumen yang akan diproses pada sistem ini berjumlah 40 dokumen berbahasa Jawa.
3. Pengklasifikasian hanya mendeteksi *full text*, sedangkan gambar dan tabel tidak di proses.
4. Dokumen akan diklasifikasi kedalam 4 kategori, diantaranya *ekonomi, kesehatan, pendidikan* dan *politik*.
5. Perhitungan yang digunakan untuk menghitung Naïve Bayes adalah dengan menggunakan  $w = tf \cdot idf$ .

6. Menggunakan aplikasi *perl* sebagai aplikasi tambahan yang digunakan untuk membantu proses *stemming*.

## 1.4. Tujuan Penelitian

Tujuan penelitian yang ingin dicapai adalah :

1. Mempelajari metode Naïve Bayesian untuk pengklasifikasian dokumen.
2. Menemukan akurasi dari metode Naïve Bayesian dalam klasifikasi dokumen bahasa Jawa.

## 1.5. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penyelesaian tugas akhir ini adalah sebagai berikut :

### 1. Studi Pustaka

Studi pustaka bertujuan untuk memberikan pengetahuan tentang hal-hal yang berkaitan dengan pengklasifikasian dokumen. Studi pustaka dilakukan dengan mempelajari buku referensi, jurnal dan artikel yang berkaitan dengan pengklasifikasian dokumen teks, metode Naïve Bayesian, dan bahasa pemrograman Java.

### 2. Pengumpulan data

Pada tahap ini dilakukan pencarian dan pengumpulan data. Data didapat dari majalah berbahasa Jawa *Jaka Lodang*, *Mekarsari* dan majalah *Praba*.

### 3. Perancangan

Pada tahap ini dilakukan perancangan sistem.

### 4. Pembuatan Sistem

Berdasarkan hasil analisis dan perancangan sistem, maka tahapan selanjutnya adalah membuat sistem yang akan digunakan.

### 5. Implementasi dan Pengujian

Implementasi sistem dengan cara menjalankan sistem yang telah dibuat dan dilakukan pengujian dengan menginputkan dokument teks dalam bahasa Jawa untuk mengetahui pengklasifikasinya.

### 6. Evaluasi

Menganalisis hasil implementasi dan membuat kesimpulan terhadap penelitian tugas akhir yang telah dikerjakan.

## 1.6. Sistematika Penulisan

Sistematika penulisan pada tulisan ini terdiri dari beberapa bab, yaitu :

### **BAB I PENDAHULUAN**

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi landasan teori yang merupakan dasar – dasar teori yang dipergunakan dalam membuat Tugas Akhir, yaitu teori tentang metode Naïve Bayesian dan *information retrieval*.

### **BAB III ANALISIS DAN PERANCANGAN**

Bab ini berisi analisis dan perancangan yang akan digunakan dalam membangun sistem.

### **BAB IV IMPLEMENTASI**

Bab ini berisi implementasi dan penjelasan fungsi program dari sistem yang dibuat.

### **BAB V HASIL DAN PEMBAHASAN**

Bab ini berisi analisis dan hasil dari pengujian yang dilakukan berdasarkan hasil dari sistem.

### **BAB VI KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran atas hasil penelitian dari Tugas akhir ini.

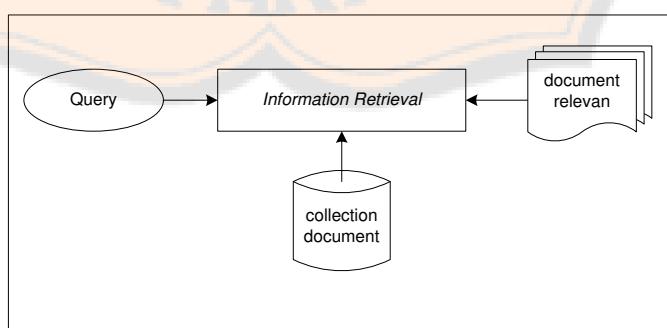
## BAB II

### LANDASAN TEORI

#### 2.1. *Information Retrieval*

*Information Retrieval* (IR) adalah menemukan bahan, biasanya dokumen, yang bersifat tidak terstruktur, biasanya teks, yang memenuhi sebuah kebutuhan informasi dari dalam koleksi besar, biasanya disimpan di komputer (Manning, 2008).

*Information Retrieval* merupakan suatu konsep tentang menemukan kembali data yang tersimpan, penyimpanan, pengorganisasian dan pengaksesan informasi. Data yang digunakan dapat berupa teks, tabel, gambar maupun video. Sistem IR yang baik memungkinkan pengguna menentukan secara cepat dan akurat apakah isi dari dokumen yang diterima memenuhi kebutuhannya. Agar representasi dokumen lebih baik, dokumen-dokumen dengan topik atau isi yang mirip dikelompokkan bersama-sama (Murad, Trevor, 2007).

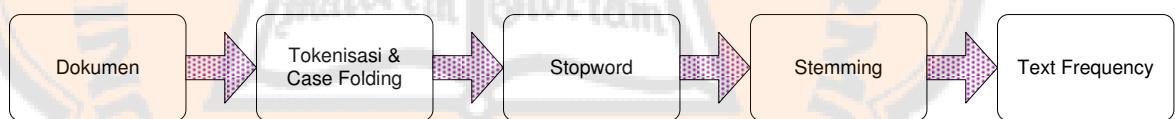


Gambar 2.1. Gambaran umum IR

Proses dalam *Information Retrieval* dapat digambarkan sebagai sebuah proses untuk mendapatkan *relevant documents* dari *collection documents* yang ada melalui pencarian *query* yang diinputkan user.

## 2.2. *Pre-processing*

Dokumen yang akan diklasifikasi, diolah terlebih dahulu melalui proses *pre-processing* untuk mendapatkan kata yang akan dibandingkan atau yang akan diberi bobot. Proses *pre-processing* menyederhanakan teks yang terdapat dalam suatu dokumen yang bersifat tidak terstruktur, terdapat banyak noise, dan struktur teks yang tidak baik. Proses *pre-processing* juga merupakan pembentukan indeks. Tahapan *pre-processing* antara lain :

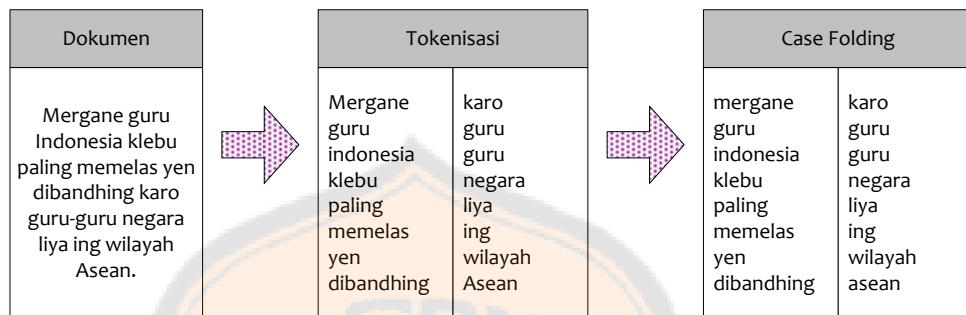


Gambar 2.2 Tahapan *pre-processing*

### 2.2.1. Tokenisasi dan *case folding*

Tokenisasi merupakan proses pemenggalan kata dalam suatu dokumen menjadi potongan – potongan kata yang berdiri sendiri (token). Proses ini juga akan menghilangkan tanda baca atau karakter yang melekat pada kata tersebut dan semua kata menjadi huruf kecil. (Manning, 2008)

Berikut gambaran proses tokenisasi dan *case folding*:



**Gambar 2.3.** Proses Tokenisasi dan *Case Folding*

### 2.2.2. *Stopword*

Kata yang sering muncul pada setiap dokumen tidak terlalu membantu atau kurang berpengaruh dalam proses klasifikasi. *Stopwords* adalah proses dimana kata – kata yang sering muncul ataupun kata yang tidak memiliki arti (misalnya kata sambung) akan dihapus. Misalnya : *aja, aku, ala, amarga, amargi, antara, apa, ta, tah, ewadhene*.

Tujuan *stopwords* adalah untuk mengefisienkan dan meningkatkan akurasi terhadap kata – kata yang dianggap penting.



**Gambar 2.4.** Proses *Stopwords*

### 2.2.3. *Stemming*

Pada umumnya setiap kata memiliki variasi kombinasi imbuhan yang beragam, tak terkecuali dalam dokumen bahasa Jawa. Variasi imbuhan dapat berupa *prefix* (awalan), *suffix* (akhiran), *infix* (sisipan). *Stemming* dapat mengurangi variasi kata yang sebenarnya memiliki kata dasar yang sama. Dengan kata lain, *stemming* merupakan proses pengembalian berbagai bentuk kata kedalam bentuk dasarnya. Sebagai contoh, kata *nyebutke* memiliki kata dasar *sebut*.

Sibelius membuat aturan *stemming* untuk bahasa Jawa, Beberapa simbol yang digunakan sebagai *stemmer rule*, adalah (Widjono, dkk, 2011) :

1. Aturan substitusi/penghapusan :

ny = "" berarti : "ny" akan dihapus

ny = s berarti : "ny" diganti "s"

2. Simbol <> digunakan untuk menyatakan tingkat *affix* yang mempengaruhi urutan pengecekan di algoritma stemming.

Peraturan yang digunakan adalah sebagai berikut :

**Tabel 2.1.** Aturan untuk *suffix*

<b>SUFFIX</b>	
<1>	ekken=>"i", kaken=>"n", okken=>"u", ekake=>"i", ekke=>"i", okake=>"u", okke=>"u", kaken=>"", kken=>"", ekaken=>"i", okaken=>"u"
<2>	ne=>"", kake=>"", kken=>"n", aken=>"", kke=>"n", enana=>"i", enono=>"i", onen=>"u", enen=>"i", onana=>"u", onono=>"u",

SUFFIX	
	ekna=>"i", ekno=>"i", okno=>"u", okna=>"u"
<3>	kake=>"n", ken=>"", kke=>"", nana=>"", nono=>"", ane=>"", nen=>"", kna=>"", kno=>"", ekne=>"i", onan=>"u", enan=>"i"
<4>	ake=>"", en=>"i", kna=>"n", kno=>"n", ana=>"", ono=>"", nane=>"", kne=>"", nan=>"", yan=>"", nipun=>"", oni=>"u", eni=>"i", nira=>""
<5>	ke=>"", ki=>"", wa=>"", ya=>"", na=>"", en=>"", an=>"", ni=>"", ipun=>"", on=>"u", ning=>""
<6>	e=>"", n=>"", a=>"", i=>"", ing=>"", ku=>"", mu=>""

Tabel 2.2. Aturan untuk *prefix*

PREFIX	
<1>	te=>"", dipun=>"", peng=>"", peny=>"", pem=>"", pam=>"", pany=>"", pra=>"", kuma=>"", kapi=>"", bok=>"", ber=>"", be=>"", ce=>"", ne=>"", mbok=>"", dak=>"", tak=>"", kok=>"", tok=>"", ing=>"", ang=>"", any=>"", am=>"", sak=>"", dhe=>"", se=>"", mang=>"", meng=>"", nge=>"", nya=>"", pi=>"", ge=>"", ke=>"", u=>"", po=>"u"
<2>	mer=>"", mi=>"", sa=>"", ku=>"", an=>"", ka=>"", ny=>"s", ng=>"k", di=>"", peng=>"k", pang=>"k", pam=>"p", ke=>"i", mang=>"k", meng=>"k", je=>""
<3>	a=>"", k=>"", pam=>"w", pan=>"t", pen=>"t", mang=>"w", meng=>"w", ny=>"c", ng=>"", ke=>"u"
<4>	n=>"t", pan=>"s", pen=>"s", man=>"s", men=>"s"
<5>	pan=>"", pen=>"", man=>"t", men=>"t", n=>""
<6>	pa=>"", pe=>"", man=>"", men=>""
<7>	p=>"", ma=>"", me=>""
<8>	m=>"w"
<9>	m=>"p"
<10>	m=>""

**Tabel 2.3.** Aturan untuk *infix*

<b>INFIX</b>	
<1>	gum=>"b",gem=>"b",kum=>"p",kem=>"p"
<2>	kum=>"w", kem=>"w"

Algoritma untuk melakukan proses stemming terhadap kata tunggal atau duplikasi.

1. Kata berimbuhan adalah word. Kata sebagai hasil adalah stemW.
2. Cek jumlah karakter word, jika < 2. Keluar.
3. Jika word mengandung “-“, maka pecah kata berdasar “-“ menjadi w1 dan w2. Dan lakukan langkah 4-13
4. w11 = w1 tanpa vokal dan w21 = w2 tanpa vokal.
5. Jika w11 = w21 dan panjang w1=w2 maka lakukan langkah 6-8
6. Jika w2 ada di kamus maka stemW=w2 dan keluar.
7. Jika w2 tidak ada di kamus, w22= hilangkan imbuhan(w2).
8. Jika w22 ada di kamus maka stemW=w22, jika tidak stemW=w1-w2 dan keluar.
9. Jika w11 != w21, lakukan langkah 10-13
10. ws11=hilangkan imbuhan(w1) dan ws21 = hilangkan imbuhan(w2).
11. Cek ws21 di kamus, jika ada maka stemW=ws21 dan keluar.
12. Cek ws11 di kamus, jika ada maka stemW=ws11 dan keluar.
13. Jika tidak maka stemW=ws11-ws21 dan keluar.

14. stemW = hilangkan imbuhan(stemW). Cek stemW di dictionary. Jika ada stemW dikembalikan dan keluar.

Algoritma untuk menghilangkan *afiks* pada kata berimbuhan.

1. Kata yang akan dihilangkan imbuhan adalah word.
2. ws1=hapus suffix (word). Cek di kamus. Jika ada kembalikan kata.
3. ws1s2=hapus suffix (ws1). Cek di kamus. Jika ada kembalikan kata.
4. ws1i1=hapus infix (ws1). Cek di kamus. Jika ada kembalikan kata.
5. dws1= pengulangan parsial (ws1). Cek di kamus. Jika ada kembalikan kata.
6. dws1s2= pengulangan parsial (ws1s2). Cek di kamus. Jika ada kembalikan kata.
7. wp1=hapus prefix (word). Cek di dictionary. Jika ada kembalikan kata.
8. dwp1= pengulangan parsial (wp1). Cek di kamus. Jika ada kembalikan kata.
9. wp1s1=hapus suffix(wp1). Cek di kamus. Jika ada kembalikan kata.
10. dwp1s1= pengulangan parsial (wp1s1). Cek di kamus. Jika ada kembalikan kata.
11. wp1s1s2=hapus suffix (wp1s1). Cek di kamus. Jika ada kembalikan kata.
12. wp1p2=hapus prefix (wp1). Cek di kamus. Jika ada kembalikan kata.

13. wp1p2s1=hapus suffix (wp1p2). Cek di kamus. Jika ada kembalikan kata.
14. wp1p2s1s2=hapus suffix (wp1p2s1). Cek di kamus. Jika ada kembalikan kata.
15. wi1=hapus infix (word). Cek di dictionary. Jika ada kembalikan kata.
16. wi1s1=hapus suffix (wi1). Cek di dictionary. Jika ada kembalikan kata.

#### **2.2.4. TF-IDF (*Term Frequency Inverse Document Frequency*)**

Setiap *term* atau kata yang telah diolah pada proses sebelumnya diberikan bobot dengan cara menghitung frekuensi kata tersebut muncul dalam dokumen. Pemberian bobot kata berdasarkan jumlah kemunculan kata t dalam dokumen d. Pembobotan ini disebut *term frequency (tf)*. Sedangkan *document frequency (df atau nt)* merupakan banyaknya dokumen yang dimiliki oleh kata t. *tf-idf* adalah nilai bobot dari suatu kata yang diambil dari nilai *tf* dan nilai *inverse idf*.

Adapun rumus pembobotan Salton (1989) adalah sebagai berikut :

$$w(t, d) = tf_{t,d} * idf_t = tf(t, d) * \log\left(\frac{N}{nt}\right) \quad (2.1)$$

Dimana :

- $w(t, d)$  = bobot dari kata t dalam dokumen d.
- $tf(t, d)$  = frekuensi kemunculan kata t dalam dokumen d.
- $idf_t$  = *inverse document frequency* dari kata t.
- N = jumlah seluruh dokumen
- nt = jumlah dokumen yang mengandung kata t.

### 2.3. Klasifikasi Teks

Klasifikasi dokumen merupakan proses untuk mengklasifikasi atau memberi label pada dokumen ke dalam kelas tertentu agar lebih mudah dikelola (Davies & Goker, 2009).

#### 2.3.1. Metode *Naïve Bayesian*

Metode Naïve Bayesian memanfaatkan probabilitas atau nilai kemungkinan. Konsep dasar yang digunakan oleh Naïve Bayes adalah Teorema Bayes, yaitu melakukan klasifikasi dengan melakukan perhitungan nilai probabilitas  $P(c|d)$ , yaitu probabilitas kelas c jika diketahui dokumen d.

Naïve Bayes menganggap sebuah dokumen sebagai kumpulan dari kata-kata yang menyusun dokumen tersebut, dan tidak memperhatikan urutan kemunculan kata pada dokumen. Perhitungan

probabilitasnya dapat dianggap sebagai hasil perkalian dari probabilitas kemunculan kata – kata pada dokumen.

Menurut Manning, Raghavan, & Schutze (2008), probabilitas sebuah dokumen d berada di kelas c dihitung dengan:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n} P(t_k|c) \quad (2.2)$$

$P(t_k|c)$  adalah *conditional probability* dari kata  $t_k$  yang terdapat dalam kelas c.  $P(t_k|c)$  dianggap sebagai ukuran seberapa banyak komponen  $t_k$  berada dalam kelas c sehingga menentukan bahwa c adalah kelas yang tepat.

$P(c)$  adalah *prior probability* dari sebuah dokumen yang terdapat dalam kelas c.

$(t_1, t_2, \dots, t_{nd})$  kumpulan kata dalam dokumen d yang digunakan untuk klasifikasi.  $d_f$  adalah jumlah kata tersebut dalam dokumen d.

Untuk memperkirakan *prior probability*  $P(c)$  digunakan persamaan sebagai berikut:

$$P(c) = \frac{N_c}{N} \quad (2.3)$$

$N_c$  adalah jumlah dokumen kelas c dalam *training*. Sedangkan N adalah jumlah keseluruhan dokumen *training* dari seluruh kelas.

Untuk memperkirakan *conditional probability*  $P(t|c)$  persamaan yang digunakan, yaitu:

$$P(W_k|c) = \frac{w_{ct}}{\sum_{W' \in V} w_{ct'}} \quad (2.4)$$

$w_{ct}$  nilai pembobotan *tfidf* atau w pada kata t dalam sebuah dokumen dari kelas c.

$\sum_{W' \in V} W_{ct}'$  jumlah total w dari keseluruhan kata yang terdapat dalam sebuah dokumen training.

Jika tidak terdapat kombinasi (term|class) pada sebuah dokumen, maka akan bernilai nol. Untuk menghilangkan nilai nol tersebut, akan digunakan *add-one* atau *Laplace smoothing*, yaitu menambahkan nilai satu pada setiap nilai  $W_{ct}$  dari perhitungan *conditional probabilities*.

Maka persamaan untuk *conditional probabilities* yaitu :

$$P(t_k | c) = \frac{w_{ct} + 1}{(\sum_{W' \in V} W_{ct}') + B'} \quad (2.5)$$

$w_{ct}$  nilai pembobotan *tfidf* atau w dari kata t di kelas c.

$\sum_{W' \in V} W_{ct}'$  jumlah total W dari keseluruhan kata (termasuk frequency) yang berada di kelas c.

B' adalah jumlah W kata unik (tidak dikali dengan tf) di semua kelas.

Untuk sebuah kata yang kemunculannya lebih dari satu kali, pangkatkan nilai *conditional probabilities* dari kelas *training* dengan *term frequency* dari kelas *testing* yang sebelumnya telah diketahui melalui proses *matching*. Kemudian jumlahkan nilainya untuk masing-masing kelas.

Untuk mendapatkan probabilitas dari kelas yang diuji terhadap seluruh kelas, maka akan dikalikan *prior probabilities* dengan total nilai *conditional probabilities* untuk masing – masing kelas.

Setelah didapat nilai probabilitas masing-masing kelas, akan dicari nilai maksimumnya, yang menunjukkan letak dokumen tersebut.

## 2.4. Evaluasi Information Retrieval

### 2.4.1. *K-fold Cross Validation*

*Cross Validation* merupakan salah satu metode yang bisa digunakan untuk mengukur kinerja sebuah sistem. Dalam *k-fold Cross validation*, data akan dipartisi secara acak ke dalam k partisi (D1, D2, ..., Dk masing – masing D memiliki jumlah yang sama). Pada iterasi pertama partisi D1 digunakan sebagai data *testing*, sedangkan sisanya akan digunakan sebagai data *training*. Maka dari itu pada iterasi pertama, D1 digunakan sebagai data *testing* dan D2, D3, ....Dk digunakan sebagai data *training*. Pada iterasi kedua, D2 digunakan sebagai data *testing*, sedangkan D1, D3, ....Dk digunakan sebagai data *training*. Pada iterasi ketiga, D3 digunakan sebagai data *testing*, sedangkan D1, D2, ...Dk digunakan sebagai data *training* dan seterusnya. Setiap sample D, hanya digunakan sekali sebagai *testing* dan berkali-kali sebagai *training* (Han&Kamber, 2006).

Pada setiap pengulangan, diukur performa dari masing – masing model yang terbentuk. Berfungsi untuk menentukan model mana yang terbaik atau efektif untuk diaplikasikan ke dalam sistem. Untuk mengukur performa sebuah model, akan digunakan perhitungan *precision* untuk mengetahui tingkat akurasinya.

#### 2.4.2. *Precision*

*Precision* adalah tingkat ketepatan atau akurasi hasil klasifikasi terhadap suatu kejadian.

$$\text{Precision } (P) = \frac{\text{jumlah data yang sesuai dengan sistem}}{\text{jumlah data testing}} \times 100\% \quad (2.6)$$

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1. Gambaran Sistem

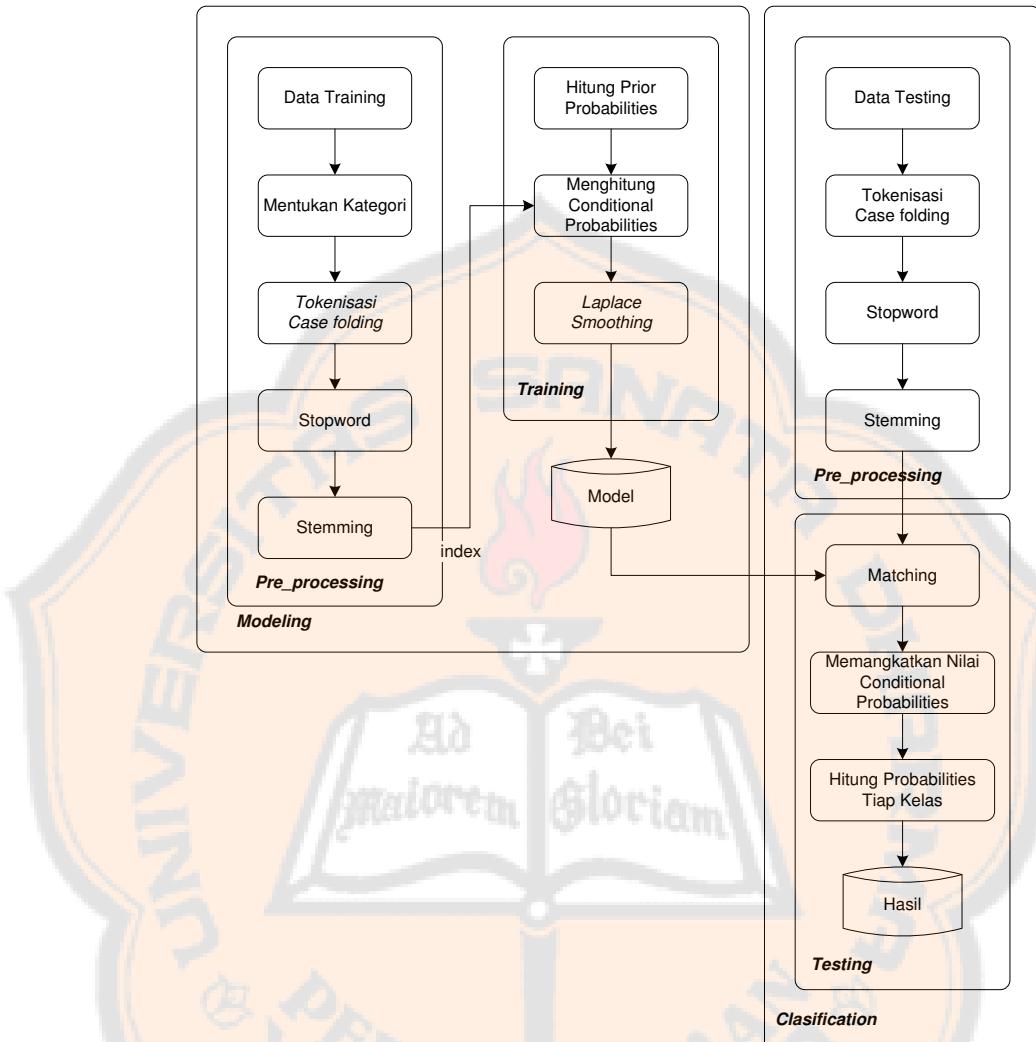
Sistem yang akan dibangun merupakan sistem berbasis teknologi informasi, digunakan dalam pengklasifikasian dokumen terutama dokumen berbahasa Jawa. Hasil yang dikeluarkan oleh sistem berupa informasi mengenai dokumen yang diolah tersebut dikategorikan atau masuk dikelas yang mana : ekonomi, kesehatan, pendidikan atau politik. Sistem ini ditujukan untuk semua kalangan yang membutuhkan bantuan dalam pengklasifikasian dokumen berbahasa Jawa yang kadang sulit dimengerti secara langsung.

Sistem terdiri atas satu bagian saja, yaitu user. Pada bagian ini, sistem akan mengklasifikasikan sebuah dokumen berbahasa Jawa dengan membandingkan dengan dokumen – dokumen yang ada di data *training* atau data pelatihan, yang sudah diketahui kategorinya.

Dokumen yang diinputkan oleh user berekstensi .txt. Proses awal, yaitu menginputkan dokumen yang akan digunakan sebagai data *training* ke dalam sistem, berdasarkan kategori yang telah diketahui. Kemudian akan dilakukan proses *pre-processing*. Proses *pre-processing* dilakukan untuk membentuk model terhadap koleksi dokumen yang diinputkan.

Proses *pre-processing* yang berupa *tokenisasi* (pemenggalan kata dan penghapusan tanda baca dan karakter), *case folding* (mengubah kata

kedalam kuruf kecil), *stopword* (penghapusan kata yang dianggap tidak penting), *stemming* (pengembalian kata kebentuk dasar), dan menghitung *tf-idf*. Setelah dilakukan *pre-processing*, maka akan menghasilkan kata unik dan bobot kata yang akan diolah untuk dihitung W dan *Laplace Smoothing* dan digunakan dalam proses klasifikasi. Kemudian pada tahap selanjutnya, yaitu tahap pengolahan data *testing*, dokumen juga akan melewati proses *pre-processing*. Dari kedua data, akan dilakukan proses *matching*, yaitu mendapatkan kata – kata yang sama dari data *training* dan data *testing*. Jika data *matching* telah diperoleh, maka akan digunakan untuk menjalankan proses klasifikasi menggunakan metode Naïve Bayesian.



Gambar 3.1. Skema Proses Klasifikasi

### 3.2. Gambaran Proses Pada Sistem

Bagian ini akan menjelaskan proses pada sistem Klasifikasi dokumen Bahasa Jawa menggunakan metode Naïve Bayesian.

Keseluruhan tahap yang akan dilalui dalam melakukan klasifikasi yaitu :

1. Pemrosesan data *training*

- a. *Pre-processing*

- i. Tokenisasi dan *case folding*

- ii. *Stopword*
- iii. *Stemming*
- iv. Menghitung *tf* dan *w* (halaman 14)

b. *Training*

- i. Menghitung *Prior Probabilities* (halaman 16)
- ii. Menghitung *Laplace Smoothing* (halaman 17)

2. Pemrosesan data *testing*

a. *Pre-processing*

- i. Tokenisasi dan *case folding*
- ii. *Stopword*
- iii. *Stemming*
- iv. Menghitung *tf*

b. *Testing*

- i. *Matching* (mendapatkan kata yang sama antara *training* dan *testing*)
- ii. Memangkatkan *Laplace Smoothing* dengan *tf* kata yang sama (hasil *matching*).
- iii. Mengalikan setiap hasil yang diperoleh dari perhitungan ii.
- iv. Menghitung probabilitas setiap kelas dan mencari nilai maksimalnya.

### 3.3. Analisa Kebutuhan

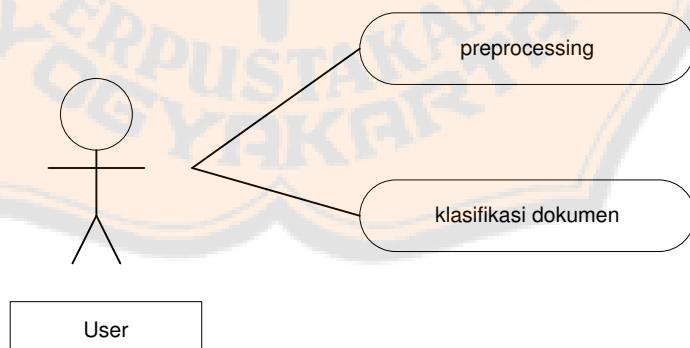
#### 3.3.1. Definisi Aktor

Aktor yang berperan menjalankan sistem ini adalah *user*. User dapat mengakses dan mengelola semua kebutuhan sistem, yaitu membentuk model dan mengklasifikasikan dokumen. Hak akses user diantaranya :

Aktor	Hak Akses
User	<ul style="list-style-type: none"><li>- <i>Pre-processing</i></li><li>- Klasifikasi Dokumen</li></ul>

#### 3.3.2. Use Case

Diagram use case merupakan gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem memahami kegunaan sistem yang akan dibangun.



**Gambar 3.2.** Diagram Use Case

### 3.3.3. Narasi Use Case

#### 1) Klasifikasi

**Tabel 3.1.** Narasi Use Case Klasifikasi

Nama Use Case	Klasifikasi												
Aktor	User												
Deskripsi Use Case	Use case ini menggambarkan proses klasifikasi dokumen bahasa Jawa ke dalam 4 kategori dengan algoritma Naïve Bayesian, menggunakan bantuan model yang telah dibangun melalui proses training.												
Pra kondisi	User berada pada halaman utama												
Langkah Umum	<table border="1"> <thead> <tr> <th>Kegiatan Aktor</th><th>Respon Sistem</th></tr> </thead> <tbody> <tr> <td>1. Menampilkan menu utama</td><td></td></tr> <tr> <td>2. Memilih menu item Klasifikasi di menu File</td><td></td></tr> <tr> <td></td><td>3. Menampilkan halaman Klasifikasi</td></tr> <tr> <td></td><td>4. Menekan tombol “Mulai”</td></tr> <tr> <td></td><td>5. Melakukan perhitungan dan menampilkan hasil klasifikasi.</td></tr> </tbody> </table>	Kegiatan Aktor	Respon Sistem	1. Menampilkan menu utama		2. Memilih menu item Klasifikasi di menu File			3. Menampilkan halaman Klasifikasi		4. Menekan tombol “Mulai”		5. Melakukan perhitungan dan menampilkan hasil klasifikasi.
Kegiatan Aktor	Respon Sistem												
1. Menampilkan menu utama													
2. Memilih menu item Klasifikasi di menu File													
	3. Menampilkan halaman Klasifikasi												
	4. Menekan tombol “Mulai”												
	5. Melakukan perhitungan dan menampilkan hasil klasifikasi.												
Langkah Alternatif													
Kesimpulan	Use case akan berhenti jika user mendapatkan hasil rekomendasi												

#### 2) Pre-processing

**Tabel 3.2.** Narasi Use Case Pre-processing

Nama Use Case	Pre-processing
Aktor	User
Deskripsi Use Case	Use case ini menggambarkan proses pre-processing, yang terdiri dari proses tokenisaasi, case folding, stopword, stemming, menghitung term frequency dari setiap kata.
Prakondisi	User berada pada halaman utama.

Langkah Umum	Kegiatan Aktor	Respon Sistem
	1. Menampilkan menu utama	
	2. Memilih menu item <i>Pre-processing</i> di menu File	
		3. Menampilkan halaman Train Dokumen
	4. Menekan tombol “Ambil Dokumen”	
		5. Menampilkan <i>file chooser</i>
	6. Memilih file yang akan di-train	
	7. Memilih kategori dokumen	
	8. Menekan tombol Train	
		9. Menampilkan konfirmasi train
		10. Menampilkan pesan jika proses train telah selesai dilakukan.
Langkah Alternatif	Jika tidak menyetujui konfirmasi train, maka akan kembali pada halaman train dokumen yang kosong.	
Kesimpulan	Use case akan berhenti jika user mendapatkan hasil rekomendasi.	

### 3) Trainer

Tabel 3.3. Narasi Use Case Trainer

Nama Use Case	Trainer
Aktor	User
Deskripsi Use Case	Use case ini menggambarkan proses perhitungan <i>Laplace Smoothing</i> pada setiap dokumen <i>training</i>
Prakondisi	User berada pada halaman utama

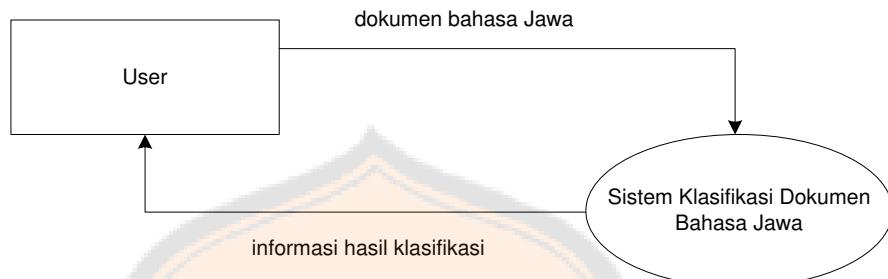
Langkah Umum	Kegiatan Aktor	Respon Sistem
	1. Menampilkan menu utama	
	2. Memilih menu item Trainer di menu File	
		3. Menampilkan halaman Trainer
	4. Menekan tombol “Mulai”	
		5. Menampilkan konfirmasi Train
		6. Menampilkan pesan bahwa proses train selesai.
<b>Langkah Alternatif</b>	Jika tidak menyetujui konfirmasi train, maka akan kembali pada halaman utama.	
<b>Kesimpulan</b>	Use case akan berhenti jika user mendapatkan hasil rekomendasi.	

### 3.4. Perancangan Model Penyimpanan Data

Media penyimpanan data yang dikelola oleh sistem berupa file yang disimpan dengan ekstensi .txt. Setiap satu file mewakili satu dokumen. File tersebut disimpan dalam folder yang mewakili masing – masing kategori. Berikut adalah daftar file dan folder yang akan digunakan oleh sistem:

1. *stopwods.txt*  
File yang berisi stopwords yang digunakan dalam sistem.
2. *kamus.txt*  
File yang berisi kata dasar dalam bahasa Jawa.
3. *tanda baca.txt*  
File yang berisi tanda baca yang akan dihilangkan dalam proses *preprocessing*.
4. *stemWord.pl*  
File yang berisi metode *stemming*.

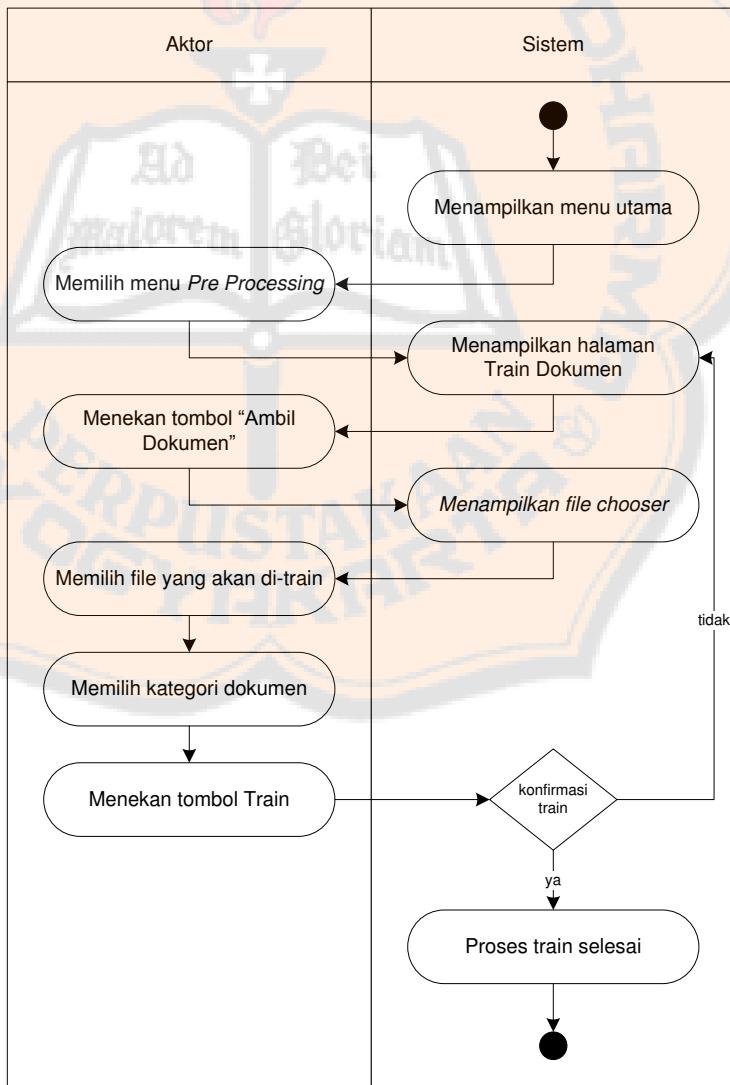
### 3.5. Diagram Konteks



Gambar 3.3. Diagram Konteks

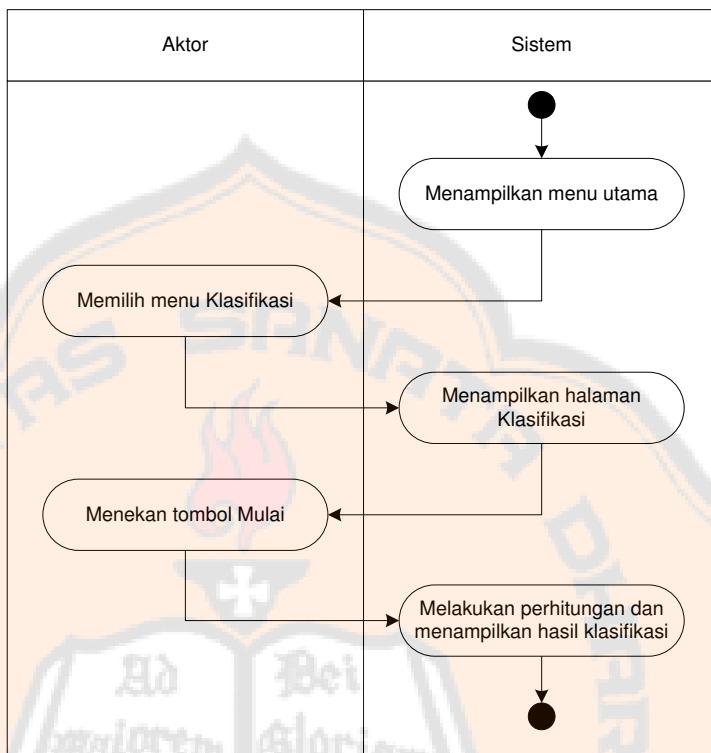
### 3.6. Diagram Aktifitas

#### 3.6.1. Diagram Aktivitas *Pre-processing*



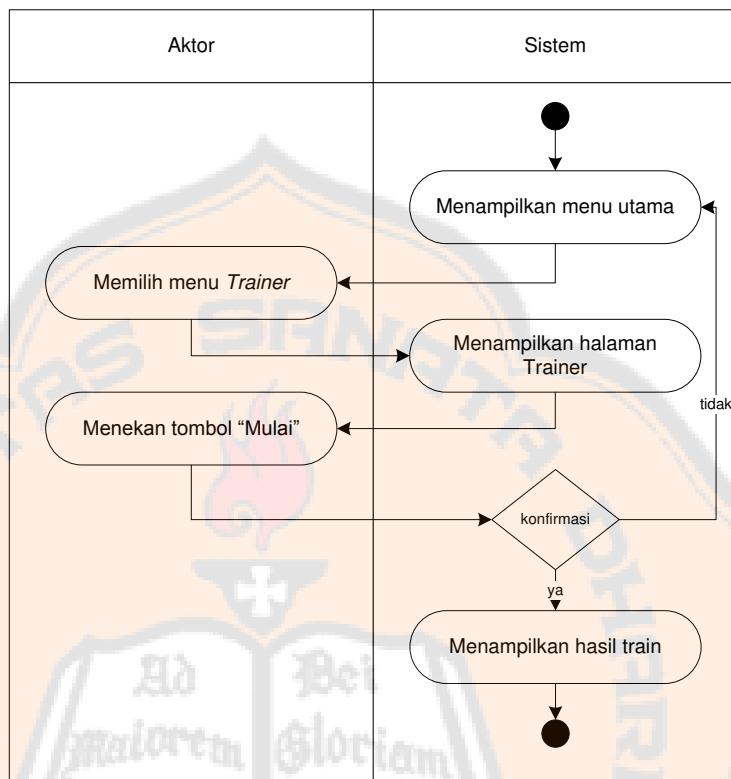
Gambar 3.4. Diagram Aktivitas *Pre-processing*

### 3.6.2. Diagram Aktifitas Klasifikasi



Gambar 3.5. Diagram Aktivitas Klasifikasi

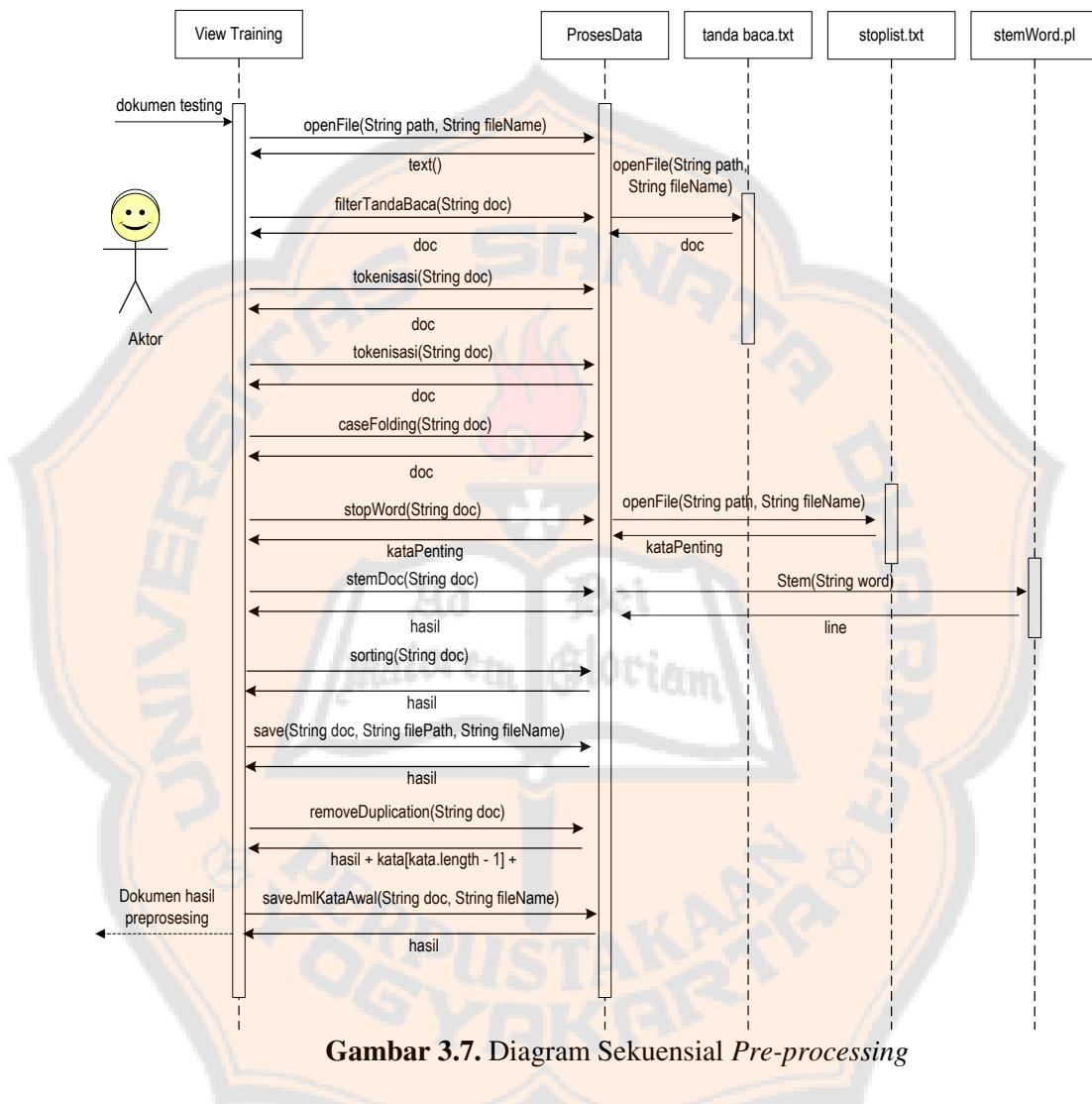
### 3.6.3. Diagram Aktifitas Trainer



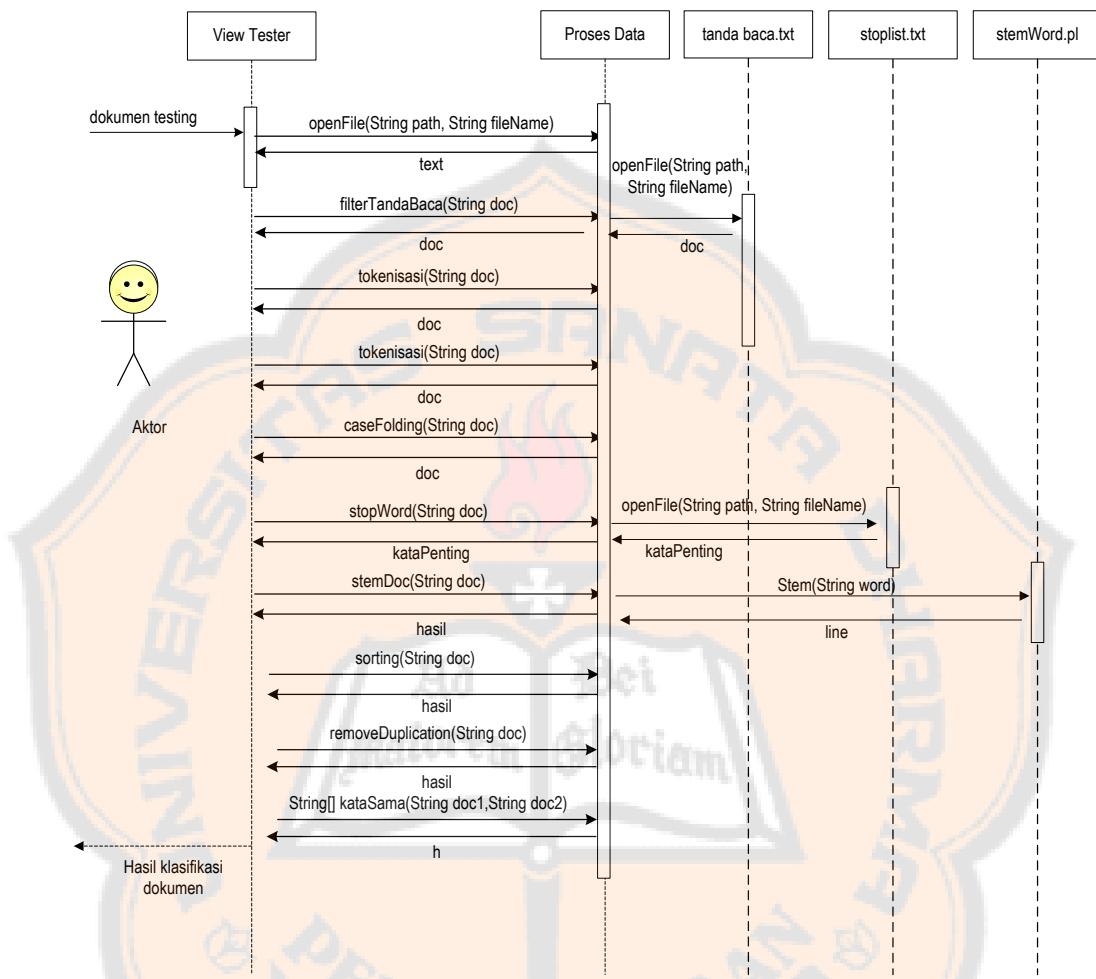
Gambar 3.6. Diagram Aktivitas Trainer

### 3.7. Perancangan Diagram Sekuensial

#### 3.7.1. Diagram Sekuensial *Pre-processing*



### 3.7.2. Diagram Sekuensial Klasifikasi



Gambar 3.8. Diagram Sekuensial Klasifikasi

### 3.8. Cara Pengujian dan Analisis Hasil

Proses pengujian penerapan algoritma berfungsi untuk mengetahui apakah sistem yang dibangun telah menerapkan algoritma Naïve Bayesian dengan tepat atau tidak. Pengujian ini dilakukan dengan membandingkan hasil dari klasifikasi manual dengan klasifikasi sistem. Akan dibandingkan pula akurasi dari hasil klasifikasi manual dengan klasifikasi sistem.

Klasifikasi manual adalah klasifikasi yang ditentukan secara manual oleh pakar atau tenaga ahli. Kelemahan dari klasifikasi manual adalah klasifikasinya bersifat subjektif, dimana apabila terdapat beberapa pakar, bisa saja hasil klasifikasi akan berbeda – beda.

Mengukur keberhasilan klasifikasi :

1. Berikut adalah pemetaan penggerjaan klasifikasi :

a) Metode pengukuran *3-fold cross validation*

Tahap I

i. *Fold 1* sebagai data uji/data *testing*

ii. *Fold 2* sebagai data pelatihan/data *training*

iii. *Fold 3* sebagai data pelatihan/data *training*

Tahap II

i. *Fold 1* sebagai data pelatihan/data *training*

ii. *Fold 2* sebagai data uji/data *testing*

iii. *Fold 3* sebagai data pelatihan/data *training*

Tahap III

- i. *Fold 1* sebagai data pelatihan/data *training*
- ii. *Fold 2* sebagai data pelatihan/data *training*
- iii. *Fold 3* sebagai data uji/data *testing*

- b) Metode pengukuran 5 fold *cross validation*

Tahap I

- i. *Fold 1* sebagai data uji/data *testing*
- ii. *Fold 2* sebagai data pelatihan/data *training*
- iii. *Fold 3* sebagai data pelatihan/data *training*
- iv. *Fold 4* sebagai data pelatihan/data *training*
- v. *Fold 5* sebagai data pelatihan/data *training*

Tahap II

- i. *Fold 1* sebagai data pelatihan/data *training*
- ii. *Fold 2* sebagai data uji/data *testing*
- iii. *Fold 3* sebagai data pelatihan/data *training*
- iv. *Fold 4* sebagai data pelatihan/data *training*
- v. *Fold 5* sebagai data pelatihan/data *training*

Tahap III

- i. *Fold 1* sebagai data pelatihan/data *training*
- ii. *Fold 2* sebagai data pelatihan/data *training*
- iii. *Fold 3* sebagai data uji/data *testing*
- iv. *Fold 4* sebagai data pelatihan/data *training*

- v. *Fold 5 sebagai data pelatihan/data training*

Tahap IV

- i. *Fold 1 sebagai data pelatihan/data training*
- ii. *Fold 2 sebagai data pelatihan/data training*
- iii. *Fold 3 sebagai data pelatihan/data training*
- iv. *Fold 4 sebagai data uji/data testing*
- v. *Fold 5 sebagai data pelatihan/data training*

Tahap V

- i. *Fold 1 sebagai data pelatihan/data training*
- ii. *Fold 2 sebagai data pelatihan/data training*
- iii. *Fold 3 sebagai data pelatihan/data training*
- iv. *Fold 4 sebagai data pelatihan/data training*
- v. *Fold 5 sebagai data uji/data testing*

## 2. *Precision*

Berikut ini adalah formula dari uji *precision* (rumus 2.6)

$$Precision (P) = \frac{\text{jumla h dokumen yang sesuai dengan sistem}}{\text{jumla h seluru h dokumen}}$$

### 3.9. Contoh Langkah Pengerjaan

#### 3.9.1. Dokumen

Diketahui terdapat 4 dokumen : pendidikan1, pendidikan2, politik1 dan politik2 yang akan menjadi data training dan digunakan untuk membangun model. Masing – masing nama dokumen mewakili nama kelasnya, misalnya pendidikan1 termasuk kelas pendidikan. Sedangkan dokumen testing akan diuji masuk ke dalam kelas pendidikan atau politik.

Berikut adalah isi dokumen yang akan digunakan :

**Tabel 3.4.** Data Training dan testing

Nama Dokumen	Isi Dokumen
pendidikan1.txt	Sasi Mei wis arep angslup. Tanggal 2 Mei wis wiwit kesilep, nanging kegiatan Hardhiknas (Hari pendidikan) isih katon marak ing saben dhaerah. Akeh pameran lan kegiatan sing nyangkut Hardhiknas mau ditindakake ing ngendi-endi. Lan ing tengah kahanan mau dadakan ana kabar sing sumebar sing asale saka statistik asing nyebutke pendidikan Indonesia saya merosot, saya melorot mudhun.
pendidikan2.txt	Kanggo biyantu ningkatake kualitas pendidikan ing Kabupaten Sleman, durung suwe iki kadhapuk pengurus Dewan pendidikan Kabupaten (DPK) Sleman. Kanthi anane DPK kasebut kaangkah masarakat ing Kabupaten Sleman bisa menehi sumbangan awujud saran, kritik lan liya-liyane kang tujuane kanggo ningkatake mutune pendidikan ing Kabupaten Sleman.
politik1.txt	Indonesia lagi ribet. Propinsi Aceh lagi panas. Perang TNI lumawan kelompok mbalela separatis GAM. Sing dha gugur wis

Nama Dokumen	Isi Dokumen
	akeh, kejaba wong-wong GAM, anggota TNI utawa Polri wis ana sing dadi tumbal kelangan nyawa. Nalare, tumrape TNI lan pemerintah, mbrasta kaum pemberontakan kaya GAM kuwi mau dudu barang sing gampang.
politik2.txt	Sawise ambruke Uni Soviet utawa USSR (Uni Soviet Sosialis Republik) taun 1991 sing ditututi negara-negara uni ing laladan Balkan (Eropa Tenggara) kaya Cekoslowakia lan Yugoslavia, akeh ramalan lamen negara uni (serikat) sing kaancam disintegrasi (perpecahan) yaiku Amerika Serikat, Cina, lan Indonesia. Saka negara uni cacah telu iki prnyata sing paling ringkiah ambruke yaiku Indonesia. Dene Amerika Serikat isih klebu negara paling kukuh minangka negara uni lan Cina durung ngatonake kahanan mutawatiri.
testing.txt	Jaman saiki "pendhidhikan" wus dudu bab sing aneh, nanging dadi barang sing larang regane dhuwur pangajine. Mung wae mutune durung mesthi. Kurikulum ing sekolah, mligine ing tingkat Sekolah Dasar wulangan Basa Jawa babagan aksara jawa durung selaras karo cak-cakane utawa prakteke. Awit ing "lapangan" wulangan mligine bab aksara Jawa durung laras karo kurikulum.

### 3.9.2. *Pre-processing*

Proses *pre-processing* dapat dilihat pada lampiran 1.

### 3.9.3. Klasifikasi

#### 1) Training

- a. Menghitung *prior probabilities* :

Menghitung *prior probabilities*  $P(c)$  dari setiap kelas, menggunakan rumus :

$$P(c) = \frac{N_c}{N}$$
$$P(\text{pendidikan}) = \frac{2}{4} = 0,5$$
$$P(\text{politik}) = \frac{2}{4} = 0,5$$

Nilai  $N_c$  = jumlah dokumen *training* dalam masing – masing kategori atau kelas.

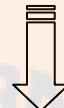
Nilai  $N$  = jumlah seluruh dokumen *training*.

- b. Menghitung *Laplace Smoothing*

Digunakan untuk menghilangkan nilai nol. Merupakan tahap akhir dari proses *training*. Hasil yang didapat dari proses ini akan menjadi Model untuk melakukan klasifikasi.

**Tabel 3.5.** Contoh Perhitungan Manual

term	tf				df	W			
	d1	d2	d3	d4		d1	d2	d3	d4
aceh	0	0	1	0	1	0	0	0,602059991	0
akeh	1	0	1	1	3	0,124938737	0	0,124938737	0,124938737
ambruke	0	0	0	2	1	0	0	0	1,204119983
amerika	0	0	0	2	1	0	0	0	1,204119983
ancam	0	0	0	1	1	0	0	0	0,602059991



wujud	0	1	0	0	1	0	0,602059991	0	0
yugoslavia	0	0	0	1	1	0	0	0	0,602059991
$\Sigma$					17,70961722	16,85767976	17,10755723	27,16648582	

term	$\Sigma W$ kata t		idf	LS	
	pendidikan	politik		pendidikan	politik
aceh	0	0,602059991	0,602059991	0,010832604	0,015703291
akeh	0,124938737	0,249877473	0,124938737	0,012186016	0,01225122
ambruke	0	1,204119983	0,602059991	0,010832604	0,021604646
amerika	0	1,204119983	0,602059991	0,010832604	0,021604646
ancam	0	0,602059991	0,602059991	0,010832604	0,015703291



wujud	0,602059991	0	0,602059991	0,017354482	0,009801937
yugoslavia	0	0,602059991	0,602059991	0,010832604	0,015703291
$\Sigma$		34,56729698	44,27404305	57,74660665	

Jumlah W(pendidikan) = 34,56730

Jumlah W(politik) = 44,27404

Jumlah idf = 57,7266

$$P(t_k | c) = \frac{W_{ct} + 1}{(\sum_{W' \in V} W'_{ct}) + B'}$$

$$\hat{P}(\text{aceh}|\text{pendidikan}) = \frac{0 + 1}{0,60206 + 34,5673} = 0,01083$$

## 2) Testing

- a. *Matching* : mencari term yang sama pada data *training* dan *testing* :

**Tabel 3.6.** Perhitungan pada *Matching*

term	tf testing	LS	
		pendidikan	politik
barang	1	0,010832604	0,015703291
dhidhik	1	0,027137299	0,009801937
mutu	1	0,017354482	0,009801937
tingkat	1	0,023876360	0,009801937

Menghitung probabilitas :

Untuk memudahkan penghitungan pada bagian  $\prod_{1 \leq k \leq n_d} P(t_k|c)$ , maka persamaan tersebut akan dihitung terlebih dahulu dalam bentuk tabel seperti di bawah. Untuk sebuah term yang kemunculannya lebih dari satu kali, pangkatkan nilai *Laplace smoothing*-nya dengan *term frequency* testing berdasarkan kata yang sama. Kemudian kalikan nilainya untuk masing-masing kelas.

Misalnya, term ‘barang’ memiliki *term frequency* sebanyak 3 kali. Pangkatkan nilai LS-nya untuk menyederhanakan penghitungan.

$$P(\text{barang}|\text{pendidikan}) = 0,009802^3 = 9,4175E-07$$

**Tabel 3.7.** Perhitungan *Laplace Smoothing*

term	tf testing	LS		LS^tf testing	
		pendidikan	politik	pendidikan	politik
barang	1	0,010832604	0,015703291	0,010832604	0,015703291
dhidhik	1	0,027137299	0,009801937	0,027137299	0,009801937
mutu	1	0,017354482	0,009801937	0,017354482	0,009801937
tingkat	1	0,023876360	0,009801937	0,023876360	0,009801937
		hasil perkalian		1,21809E-07	1,47886E-08
		perkalian dengan prior probabilities		6,09045E-08	7,39429E-09
		nilai maksimal		6,09045E-08	

Kemudian mendapatkan nilai probabilitas dari testing terhadap seluruh kelas dengan cara mengalikan nilai prior probabilities dengan total nilai *Laplace Smoothing* untuk masing – masing kelas.

Probabilitas masing-masing kelas terhadap kelas testing:

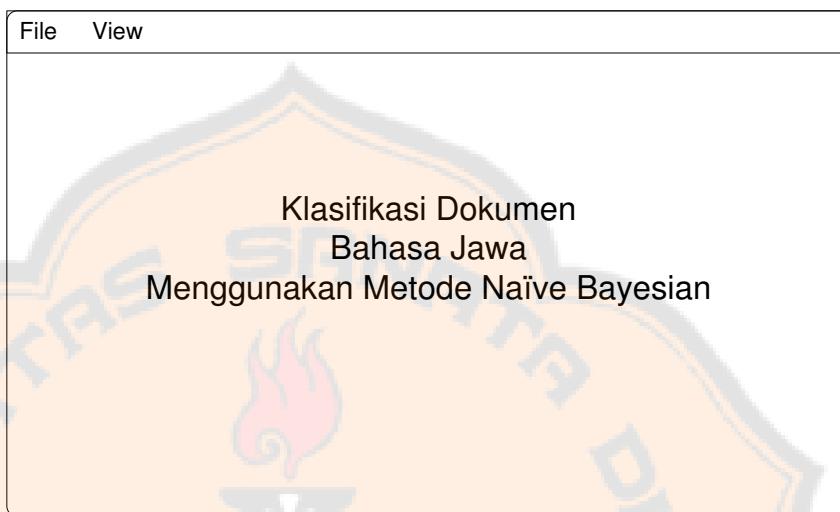
$$P(\text{pendidikan}|\text{testing}) = 0,5 * 1,21809\text{E-}07 = 6,09045\text{E-}08$$

$$P(\text{politik}|\text{testing}) = 0,5 * 1,47886\text{E-}08 = 7,39429\text{E-}09$$

Dari hasil perhitungan probabilitas diketahui bahwa probabilitas kelas pendidikan memiliki nilai yang paling tinggi, sehingga *testing* masuk ke dalam kategori pendidikan (hasil uji benar)

### 3.10. Perancangan Antar Muka (Interface)

#### 3.10.1. Menu Utama



Gambar 3.9. Desain Menu Utama

#### 3.10.2. Menu Klasifikasi Dokumen

Halaman ini akan mengolah klasifikasi dokumen.

A user interface design for document classification. It features a sidebar on the left with options "Direktori Dokumen" and "Daftar Dokumen". The main area has input fields for "direktori" (directory) and "nama dokumen" (document name), a "Mulai" (Start) button, and output fields for "Hasil" (result) and "hasil klasifikasi" (classification result). There is also a "Ubah" (Change) button.

Gambar 3.10. Desain Klasifikasi

#### 3.10.3. Menu Pre-processing

Antarmuka ini dibutuhkan untuk menghasilkan model yang digunakan dalam proses klasifikasi. Model dalam sistem ini

bersifat statis, maka proses training hanya dilakukan sekali. Namun, apabila dibutuhkan perubahan pada model, maka proses training atau *pre-processing* dapat dilakukan kembali.

The screenshot shows a user interface for 'Pre-processing'. At the top, there is a text input field labeled 'direktori dokumen' and three buttons: 'Ambil Dokumen', 'Train', and 'Reset'. Below this, a section titled 'Kategori Dokumen' contains four checkboxes: 'Ekonomi', 'Politik', 'Kesehatan', and 'Pendidikan'. At the bottom, there is a table with columns 'No', 'Dokumen', and 'Kategori'.

No	Dokumen	Kategori

Gambar 3.11. Desain *Pre-processing*

#### 3.10.4. Menu Trainer

The screenshot shows a user interface for 'Trainer'. On the left, there is a section titled 'Direktori Dokumen Train' with a 'Ubah' button. Below it, a table titled 'Daftar File Train' lists categories with their respective counts: Ekonomi (jumlah ekonomi), Politik (jumlah politik), Pendidikan (jumlah pendidikan), Kesehatan (jumlah kesehatan), and Total (total dokumen). To the right, there is a large text area labeled 'daftar file train' and a 'Mulai' button at the bottom.

	jumlah ekonomi
Ekonomi	
Politik	
Pendidikan	
Kesehatan	
Total	

Gambar 3.12. Desain *Trainer*

Menu *trainer* berfungsi sebagai menu perhitungan saja.

## BAB IV

### IMPLEMENTASI SISTEM

Penelitian ini telah diimplementasikan menjadi sebuah aplikasi oleh Yustinus Euzhan Yogatama, yang siap digunakan dan dibangun dengan tahapan-tahapan berikut :

#### 4.1. Spesifikasi *Software* dan *Hardware*

Spesifikasi *software* yang digunakan adalah sebagai berikut :

1. Sistem operasi : Windows 8 32-bit
2. Java NetBeans IDE 6.8
3. Java JDK 1.6.0\_20
4. Perl, digunakan dalam membantu proses *stemming*.

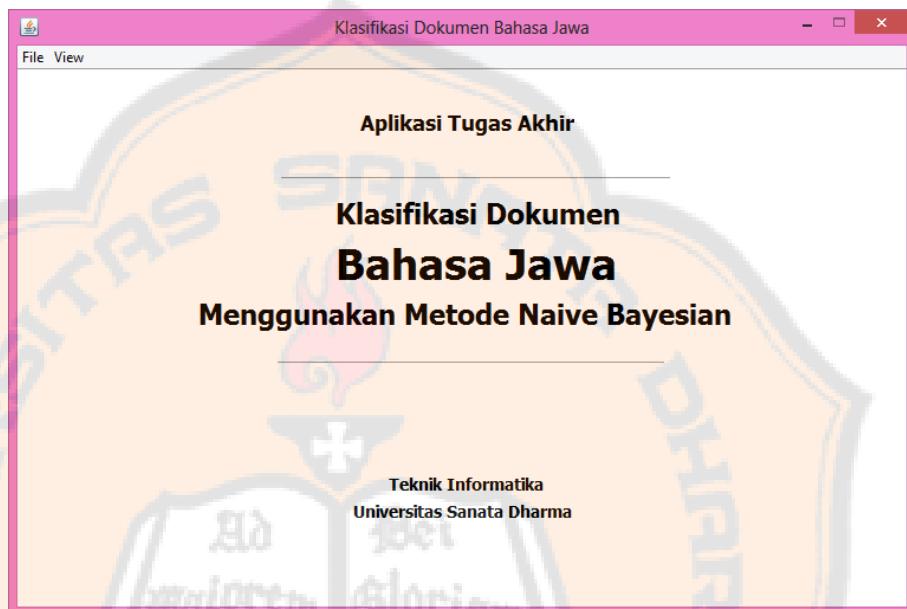
Spesifikasi *hardware* yang digunakan adalah sebagai berikut :

1. Processor : Intel Core 2 Duo
2. Memori : 2 GB
3. Hard Disk : 320 GB

## 4.2. Implementasi Antar Muka

Implementasi ini digunakan untuk mempermudah penggunaan sistem.

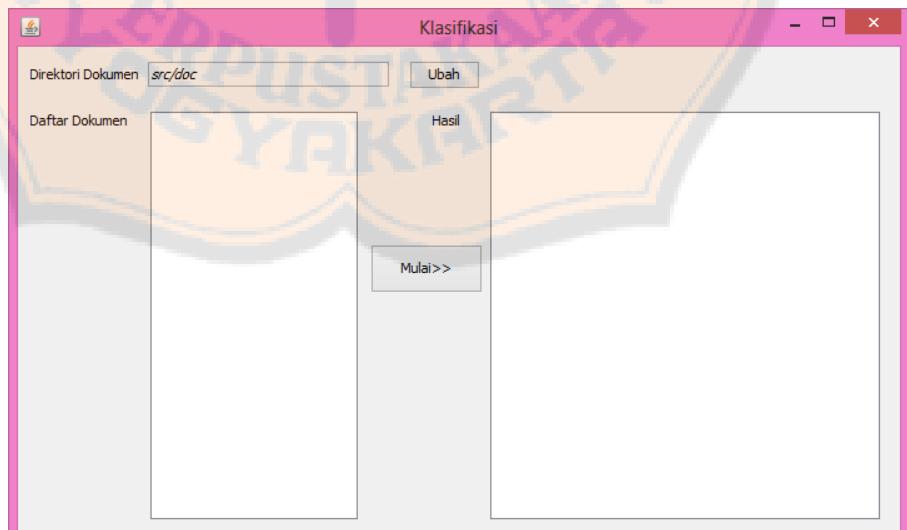
### 4.2.1. Antarmuka MainFrame



Gambar 4.1. Antarmuka MainFrame

Halaman ini merupakan halaman utama sistem.

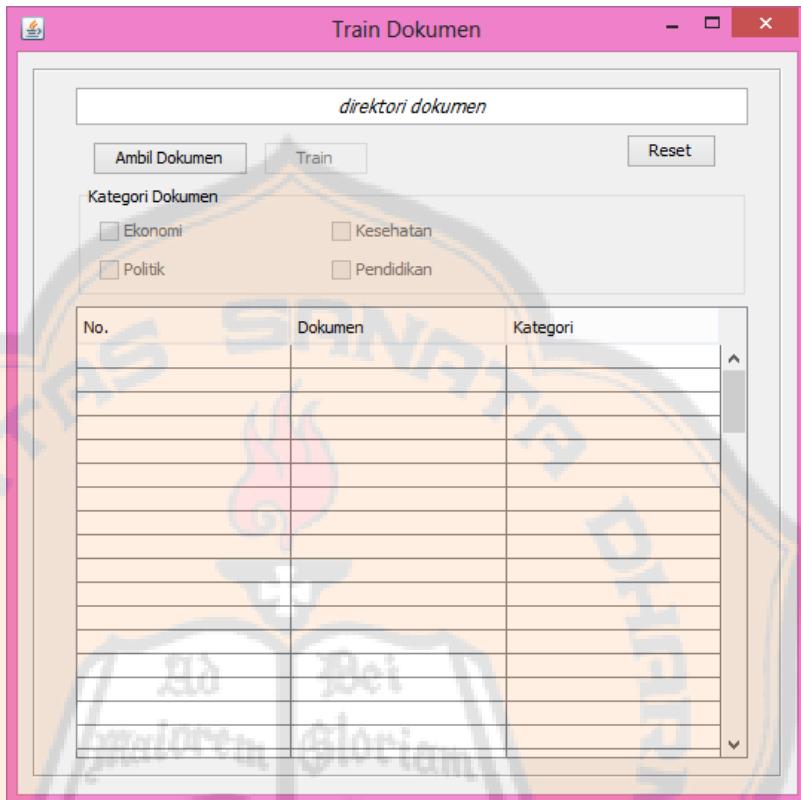
### 4.2.2. Antarmuka Klasifikasi



Gambar 4.2. Antarmuka Klasifikasi

Berfungsi untuk melakukan proses klasifikasi.

#### 4.2.3. Antarmuka *Pre-processing*

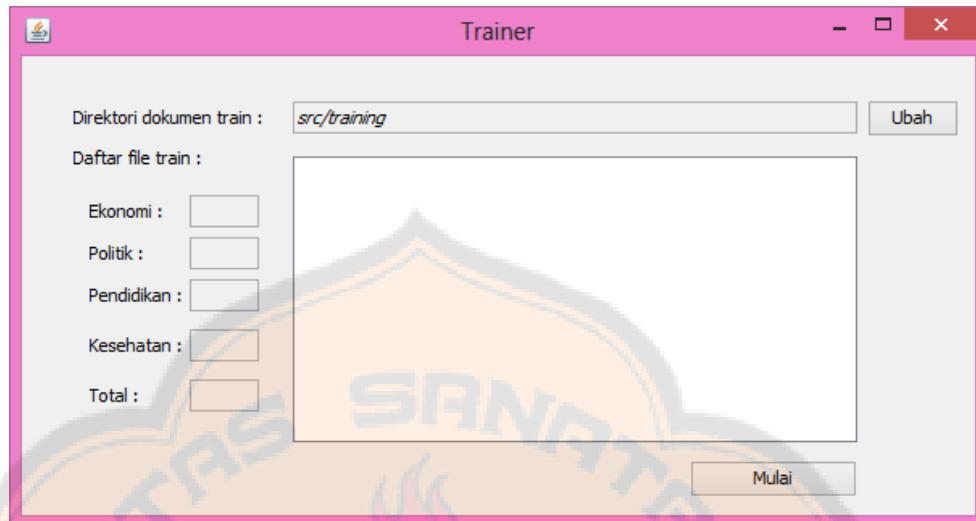


Gambar 4.3. Antarmuka *Pre-processing*

Halaman ini untuk menghasilkan model yang digunakan dalam proses klasifikasi. Model dalam sistem ini bersifat statis, sehingga proses *train* pada dasarnya hanya dilakukan sekali. Apabila dalam proses terdapat perubahan data, maka proses *train* dapat dilakukan kembali.

#### 4.2.4. Antarmuka Trainer

Halaman ini berfungsi untuk melakukan perhitungan terhadap file hasil preprocessing, yaitu untuk menghitung bobot kata hingga *Laplace Smoothing*.



Gambar 4.5. Antarmuka Trainer

### 4.3. Implementasi *Preprocessing*

#### 4.3.1. Implementasi Membaca File Dokumen

Proses ini berfungsi untuk membaca isi dokumen/file.

```
public static String openFile(String path, String fileName)
throws FileNotFoundException, IOException {
    String text = "", teks = "";
    FileReader fr = new FileReader(path + "" + fileName);
    BufferedReader br = new BufferedReader(fr);

    while ((teks = br.readLine()) != null) {
        text = text + teks + "\n";
    }
    br.close();
    fr.close();
    return text;
}
```

List Code 4.3.1. Membaca File

#### 4.3.2. Tokenisasi dan *Case Folding*

Proses ini berfungsi untuk mengubah spasi menjadi enter, agar menjadi per kata dan mengubah huruf besar menjadi huruf kecil.

```
//proses tokenisasi
public static String tokenisasi(String doc) {
    doc = replace(doc, " ", "\n");
    return doc;
```

```

    }

//proses casefolding
public static String caseFolding(String doc) {
    doc = doc.toLowerCase();
    return doc;
}

```

*List Code 4.3.2. Tokenisasi dan Case Folding*

#### **4.3.3. Implementasi Stopwords**

Method ini berfungsi untuk mengerjakan proses stopword atau penghilangan kata – kata yang terdaftar dalam stoplist.

```

public static String stopWord(String doc) throws
FileNotFoundException, IOException {
String stoplist = openFile("src/klasifikasidokumen/",
"stoplist.txt");
StringTokenizer stop = new StringTokenizer(stoplist);
String[] stopA = new String[stop.countTokens()];
for (int i = 0; i < stopA.length; i++) {
    stopA[i] = stop.nextToken();
}
StringTokenizer token = new StringTokenizer(doc);
String[] tokenA = new String[token.countTokens()];
for (int i = 0; i < tokenA.length; i++) {
    tokenA[i] = token.nextToken();
}
String kataPenting = "";
for (int i = 0; i < tokenA.length; i++) {
    String t = "";
    for (int j = 0; j < stopA.length; j++) {
        if (tokenA[i].equalsIgnoreCase(stopA[j])) {
            tokenA[i] = "";
        }
    }
}
for (int i = 0; i < tokenA.length; i++) {
    if (tokenA[i].isEmpty()) {
        //do nothing
    } else {
        kataPenting = kataPenting + tokenA[i] + "\n";
    }
}
return kataPenting;
}

```

*List Code 4.3.3 Stopwords*

#### **4.3.4. Implementasi Stemming**

Berfungsi untuk menghilangkan imbuhan dan akhiran sehingga didapatkan kata dasar.

```
public static String stem(String word) {  
    String[] cmd = {"C:/Perl/bin/perl",  
    "D:/Kape/KlasifikasiDokumen/src/klasifikasidokumen/stemWord.  
pl", word};  
    Process process;  
    String line = "";  
    try {  
        process = Runtime.getRuntime().exec(cmd);  
        BufferedReader output = new BufferedReader(new  
InputStreamReader(process.getInputStream()));  
  
        line = output.readLine();  
        if (line == null) {  
            line = word;  
        }  
  
        output.close();  
    } catch (Exception e) {  
        System.out.println("Exception: " + e.toString());  
    }  
    return line;  
}
```

*List Code 4.3.4a Stemming*

```
#1. make a rule  
#2. open text file  
#3. get one word  
#4. stem  
#5. compare with the real root word  
#6. count the true word stem  
  
local %suffix_1;  
local %suffix_2;  
local %suffix_3;  
local %suffix_4;  
local %suffix_5;  
  
local %prefix_1;  
local %prefix_2;  
local %prefix_3;  
local %prefix_4;  
local %prefix_5;  
local %prefix_6;  
local %prefix_7;  
local %prefix_8;  
local %prefix_9;  
local %prefix_10;  
  
local %infix_1;  
local %infix_2;  
local %dict;  
  
my $word = $ARGV[0];  
  
my $fileOp;  
#  
$fileOp="E:\\test.txt";  
#  
open FILE, "<", $fileOp or die "Can't open";
```

```

my $fileTest="E:\\testhasil2.txt";
#
open FILETESTH, ">", $fileTest or die $!;
initial();

    my $stemWord=stem(lc $word);
    print $stemWord;

sub initial{
    #dictionary
    #hash pasangan substitusi
    #list prefix, suffix, infix
    $fileOp="D:\\Kape\\Aplikasi
piol\\kamus.txt";                                Klasifikasi
    open FILEDIC, "<", $fileOp or die "Can't open";
    while (<FILEDIC>)
    {
        chomp;
        $dict{$_}=$_;
    }

    #daftar tingkat dan substitusinya
    %suffix_1=(ekken=>"i", kaken=>"n", okken=>"u",
ekake=>"i", ekke=>"i", okake=>"u", okke=>"u", kaken=>"",
kken=>"", ekaken=>"i", okaken=>"u");
    %suffix_2=(ne=>"", kake=>"", kken=>"n", aken=>"",
kke=>"n", enana=>"i", enono=>"i", onen=>"u", enen=>"i",
onana=>"u", onono=>"u", ekna=>"i", ekno=>"i", okno=>"u",
okna=>"u");
    %suffix_3=(kake=>"n", ken=>"", kke=>"", nana=>"",
nono=>"", ane=>"", nen=>"", kna=>"", kno=>"", ekne=>"i",
onan=>"u", enan=>"i");
    %suffix_4=(ake=>"", en=>"i", kna=>"n", kno=>"n",
ana=>"", ono=>"", nane=>"", kne=>"", nan=>"", yan=>"",
nipun=>"", oni=>"u", eni=>"i", nira=>"");
    %suffix_5=(ke=>"", ki=>"", wa=>"", ya=>"", na=>"",
en=>"", an=>"", ni=>"", ipun=>"", on=>"u", ning=>"");
    %suffix_6=(e=>"", n=>"", a=>"", i=>"", ing=>"",
ku=>"", mu=>"");
    %prefix_1=(te=>"",
dipun=>"", peng=>"", peny=>"", pem=>"", pam=>"", pany=>"",
pra=>"",
kuma=>"", kapi=>"", bok=>"", ber=>"", be=>"", ce=>"",
ne=>"",
mbok=>"", dak=>"", tak=>"", kok=>"", tok=>"", ing=>"",
ang=>"", any=>"",
am=>"", sak=>"", dhe=>"", se=>"", mang=>"", meng=>"",
nge=>"", nya=>"",
pi=>"", ge=>"", ke=>"", u=>"", po=>"u");
    %prefix_2=(mer=>"", mi=>"", sa=>"", ku=>"",
an=>"", ka=>"",
ny=>"s", ng=>"k", di=>"", peng=>"k", pang=>"k", pam=>"p",
ke=>"i",
mang=>"k", meng=>"k", je=>"");
    %prefix_3=(a=>"", k=>"", pam=>"w", pan=>"t", pen=>"t",
mang=>"w", meng=>"w", ny=>"c", ng=>"",
ke=>"u");
    %prefix_4=(n=>"t", pan=>"s", pen=>"s", man=>"s", men=>"s");
    %prefix_5=(pan=>"", pen=>"", man=>"t", men=>"t", n=>"");
    %prefix_6=(pa=>"", pe=>"", man=>"", men=>"");
    %prefix_7=(p=>"", ma=>"", me=>"");
    %prefix_8=(m=>"w");

    %prefix_9=(m=>"p");
    %prefix_10=(m=>"");

%infix_1=(gum=>"b", gem=>"b", kum=>"p");

```

```
%infix_2=(kum=>"w");
}
sub hilangPref{
    my $word = @_ [0];
    my $w=$word;

    if ($w =~ /(^te|dipun|peng|peny|pem|pam|pan|pany|pra|kuma|
kapi|bok|ber|be|ce|ne|mbok|dak|tak|kok|tok|ing|ang|any|am|sa|
k|dhe|se|mang|meng|nge|nya|pi|ge|ke|u|po)/)
    {
        $stem=$prefix_1{$1}.$';
        print FILETESTH $stem." p1 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
    }
    if($w=~ /(^mer|mi|sa|ku|an|ka|ny|ng|di|peng|pang|pam|
ke|mang|meng|je)/)
    {
        $stem=$prefix_2{$1}.$';
        print FILETESTH $stem." p2 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
    }
    if($w=~ /(^a|k|pam|pan|pen|mang|meng|ny|ng|ke)/)
    {
        $stem=$prefix_3{$1}.$';
        print FILETESTH $stem." p3 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
    }
    if($w=~ /(^n|pan|pen|man|men)/)
    {
        $stem=$prefix_4{$1}.$';
        print FILETESTH $stem." p4 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
    }
    if($w=~ /(^pan|pen|man|men|n)/)
    {
        $stem=$prefix_5{$1}.$';
        print FILETESTH $stem." p5 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
    }
    if($w=~ /(^pa|pe|man|men)/)
    {
        $stem=$prefix_6{$1}.$';
        print FILETESTH $stem." p6 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
    }
    if($w=~ /(^p|ma|me)/)
    {
        $stem=$prefix_7{$1}.$';
        print FILETESTH $stem." p7 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
    }
    if($w=~ /(^m)/)
    {
        $stem=$prefix_8{$1}.$';
        print FILETESTH $stem." p8 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
        $stem=$prefix_9{$1}.$';
        print FILETESTH $stem." p9 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem;}
        $stem=$prefix_10{$1}.$';
    }
}
```

```

        print FILETESTH $stem." p10 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem; }

    }
    return $w;
}
sub hilangSuf{
    my $word = @_ [0];

    my $w=$word;
    if ($w =~ /(ekken|kaken|okken|ekake|ekke|okake|okke|
kaken|kken|ekaken|okaken)$/)
    {
        $stem=$`.$suffix_1{$1};
        print FILETESTH $stem." 1 ".$w."\n";

    } #hilang akhiran 2
    elsif ($w =~ /(ne|kake|kken|aken|kke|enana|enono|onen|
enen|onana|onono|ekna|ekno|okno|okna)$/)
    {
        $stem=$`.$suffix_2{$1};
        print FILETESTH $stem." 2 ".$w."\n";

    } #hilang akhiran 3
    elsif ($w =~ /(kake|ken|kke|nana|nono|ane|nen|kna|
kno|ekne|onan|enan)$/)
    {
        $stem=$`.$suffix_3{$1};
        print FILETESTH $stem." 3 ".$w."\n";

    } #hilang akhiran 4
    elsif ($w =~ /(ake|en|kna|kno|ana|ono|nane|kne|nan|
yan|nipun|oni|eni|nira)$/)
    {
        $stem=$`.$suffix_4{$1};
        print FILETESTH $stem." 4 ".$w."\n";

    } #hilang akhiran 5
    elsif ($w =~ /(ke|ki|wa|ya|na|en|an|ni|ipun|on|
ning)$/)
    {
        $stem=$`.$suffix_5{$1};
        print FILETESTH $stem." 5 ".$w."\n";

    } #hilang akhiran 6
    elsif ($w =~ /(e|n|a|i|ing|ku|mu)$/)
    {
        $stem=$`.$suffix_6{$1};
        print FILETESTH $stem." 6 ".$w."\n";
    }

    if (exists $dict{$stem})
    {
        return $stem;
    }
    else
    {
        #hilang prefix
        my $stemPref=hilangPref($stem);
        if (exists $dict{$stemPref}){ return $stemPref; }
    }
}
sub stem{
    my $word = @_ [0];
    #jika panjang kata < 2 keluar
    if (length($word)<2){return $word;}
    #print $word."\n";
    #loop
}

```

```
# hilangkan akhiran tingkat 1 , cek kamus, jika ada break
# hilangkan awalan tingkat 1, cek kamus, jika ada break
# kembalikan akhiran tingkat 1, cek kamus, jika ada break
#
my $w=$word;
if (exists $dict{$w}){ return $w; }

#hilang infix
if (index($w,"in") == 1 || index($w,"um") == 1 || index($w,"em") == 1 || index($w,"el") == 1 || index($w,"er") == 1)
{
    $_= $w;
    s/(in|um|em|el|er)//;
    print FILETESTH $_." i1 ".$w."\n";
    if (exists $dict{$_}){ return $_; }
    elsif ($w =~ /^(gum|kum|gem)/)
    {
        $stem=$infix_1{$1}.$';
        print FILETESTH $stem." i2 ".$w."\n";
        if (exists $dict{$stem})
        { return $stem; }
    }
    else
    {
        my $stemPref=hilangPref($_);
        if(exists $dict{$stemPref}){ return $stemPref; }

        #hilang suffix
        my $hs=hilangSuf($_);
        if (exists $dict{$hs}){return $hs; }

    }
}

#kata reduplikasi
if ($w =~ m/[-]/)
{
    $_= $w; split/-/;
    if (exists $dict{$'}){ return $'; }
    else
    {
        #hilang suffix
        #if (exists $dict{hilangSuf($')}){return $'; }
        $w=$';
    }
}

#hilang awalan saja
my $stemPref=hilangPref($w);
if (exists $dict{$stemPref}){ return $stemPref; }

#hilang suffix
my $hs=hilangSuf($w);
if (exists $dict{$hs}){return $hs; }

#hilang reduplikasi tanpa -
if ((index($w,"e") == 1 || index($w,substr($w,0,1),2) == 2)
```

```

    {
        $dua=substr($w,0,2);
        $_[0] = $w; $_[0] =~ s/$dua//;
        if (exists $dict{$_}) { return $_[0]; }
        else { $w = $_[0]; }
    }
    return $w;
}

```

*List Code 4.3.4b Stemming Perl*

#### 4.4. Implementasi Klasifikasi

Hal pertama yang dilakukan oleh proses klasifikasi adalah membaca hasil training yang telah dilakukan sebelumnya.

```

public Tester() throws FileNotFoundException, IOException {
    initComponents();
    namaDok = new String[4];
    namaDok[0] = "Ekonomi";
    namaDok[1] = "Politik";
    namaDok[2] = "Pendidikan";
    namaDok[3] = "Kesehatan";

    jTextField1.setText("src/doc");
    listFile = null;
    try {
        listFile = ProsesData.listFiles(jTextField1.getText());
    } catch (IOException ex) {

        Logger.getLogger(Trainer.class.getName()).log(Level.SEVERE, null,
        ex);
    }

    String list = "";
    for (int i = 0; i < listFile.length; i++) {
        list = list + listFile[i] + "\n";
    }
    jTextArea1.setText(list);

    // membuka file hasilTraining utk dibaca kata dan LS
    String docLS = ProsesData.openFile("src/hasilTraining/",
    "hasilTraining.txt");
    StringTokenizer tok1 = new StringTokenizer(docLS, "#\n");

    term = new String[tok1.countTokens()];
    LS = new double[term.length][4];
    String[] kata = new String[1];
    double[] dataLS = new double[4];
    jmlDoc = new double[4];

    int idx = 0;
    int temp = 0;
    int idxKata = 0;
    int indexLS = 0;
    while (tok1.hasMoreTokens()) {
        StringTokenizer tok2 = new StringTokenizer(tok1.nextToken(),
        "#\n");
        String token = tok2.nextToken();
        if (token.equals("LS")) {
            indexLS++;
            for (int i = 0; i < 4; i++) {
                LS[temp][i] = Double.parseDouble(tok2.nextToken());
            }
        } else if (token.equals("DOC")) {
            temp++;
            for (int i = 0; i < 4; i++) {
                jmlDoc[i] = Double.parseDouble(tok2.nextToken());
            }
        } else {
            for (int i = 0; i < 4; i++) {
                dataLS[i] = Double.parseDouble(tok2.nextToken());
            }
            for (int i = 0; i < 4; i++) {
                LS[temp][i] += dataLS[i];
            }
        }
    }
}

```

```

"=");
    while (tok2.hasMoreTokens()) {
        kata[idxKata] = tok2.nextToken();
//System.out.println("*****tok2 >>> *****" + kata[idxKata]);
        if (temp % 2 != 1) {
            term[idx] = kata[idxKata];
//System.out.println("*****Term >> *****" + term[idx]);
            idx++;
        } else {
            StringTokenizer tok3 = new StringTokenizer(kata[idxKata], ";");
            int idxLS = 0;
            while (tok3.hasMoreTokens()) {
                LS[indexLS][idxLS] = Double.parseDouble(tok3.nextToken());
                idxLS++;
            }
            indexLS++;
        }
        temp++;
    }
}

String daftar = "";
for (int i = 0; i < term.length; i++) {
    daftar = daftar + term[i] + "\n";
}
daftarKata = daftar;
// System.out.println(daftar); // daftar kata gabungan

String d = ProsesData.openFile("src/hasilTraining/",
"jmlDoc.txt");
StringTokenizer dtok = new StringTokenizer(d, "; ");
int in = 0;
for (int i = 0; i < 4; i++) {
    jmlDoc[in] = Integer.parseInt(dtok.nextToken());
// System.out.println(jmlDoc[in]);
    in++;
}
}
}

```

*List Code 4.4.1 Membaca hasil training*

Setelah membaca hasil training, dilakukan proses *pre-processing* terhadap data testing.

```

String tempDoc;
tempDoc = ProsesData.openFile(jTextField1.getText() + "/",
listFile[x].toString());
//System.out.println(tempDoc);
tempDoc = ProsesData.filterTandaBaca(tempDoc);
tempDoc = ProsesData.tokenisasi(tempDoc);
tempDoc = ProsesData.caseFolding(tempDoc);
tempDoc = ProsesData.stopWord(tempDoc);
tempDoc = ProsesData.stemDoc(tempDoc);

String[] listTermDoc;
StringTokenizer tok = new StringTokenizer(tempDoc);
InvertedIndex inv = new InvertedIndex();
while (tok.hasMoreTokens()) {
    inv.add(tok.nextToken()),
}

```

```

listFile[x].toString());
}
tempDoc = ProsesData.sorting(tempDoc);
//System.out.println("#####\n" + tempDoc); //sorting kata testing
tempDoc = ProsesData.removeDuplication(tempDoc);

```

**List Code 4.4.2** Proses preprocessing pada data testing

Melakukan proses *matching*, yaitu mencari kata yang sama dari *training* dan *testing*.

```

String[] kataSama;
kataSama = ProsesData.kataSama(tempDoc, daftarKata);
// System.out.println("Temdoc :" + tempDoc);
// System.out.println("Daftar Kata :" + daftarKata);
System.out.println("matching : \n");
for (int i = 0; i < kataSama.length; i++) {
    System.out.println(kataSama[i]);
}

```

**List Code 4.4.3** Matching

Memangkatkan *Laplace Smoothing* dari dokumen *training* dengan *term frequency testing*.

```

double[][] prob = new double[kataSama.length][4];
for (int k = 0; k < kataSama.length; k++) {
//System.out.println(kataSama[k]);
    for (int j = 0; j < term.length; j++) {
//System.out.println(term[j]);
        if (kataSama[k].equalsIgnoreCase(term[j].toString())) {
//System.out.println("MASUKKK"+inv.cariKata(term[j].toString()));
//System.out.println("-----"); // pembatas antar kelas
            for (int l = 0; l < 4; l++) {
                prob[k][l] = Math.pow(LS[j][l],
inv.cariKata(term[j].toString()));
            }
        }
    }
}

```

**List Code 4.4.4.** Memangkatkan Laplace Smoothing dengan tf testing

Mengalikan *prior probabilities* masing – masing kelas dengan keempat hasil perkalian *Laplace Smoothing* dengan tf *testing*.

```

//mengalikan prob tiap dokumen
double[] jmlProb = new double[4];
//System.out.println(" LS^tf testing");
for (int i = 0; i < 4; i++) {
    double temp = 0;
//System.out.println("#####");

```

```

        for (int j = 0; j < kataSama.length; j++) {
            if (temp == 0) {
                temp = prob[j][i];
            } else {
                temp = temp * prob[j][i];
            }
        }

double jmldokumen = (jmlDoc[0] + jmlDoc[1] + jmlDoc[2] +
jmlDoc[3]);
System.out.println("\njumlah dokumen = " + jmldokumen);
double[] probabilitas = new double[4];
for (int i = 0; i < probabilitas.length; i++) {
    System.out.println(" " + jmlProb[i] + " dengan jumlah dokumen
" + namaDok[i] + " = " + jmlDoc[i]);
    double a = jmlProb[i];
    double b = (jmlDoc[i] / jmldokumen);
    probabilitas[i] = a * b;
    System.out.println("Probabilitas " + probabilitas[i]);
}

```

*List Code 4.4.5. Mengalikan prior probabilities dengan Laplace Smoothing*

Membandingkan diantara keempat kategori, mana yang memiliki nilai maksimal.

```

double tempp = 0;
String namaDokumen = "";
for (int i = 0; i < 4; i++) {
    if (probabilitas[i] > tempp) {
        tempp = probabilitas[i];
        namaDokumen = namaDok[i];
    } else {
    }
}
System.out.println("nilai yang paling tinggi = " + tempp + " dan
masuk ke dalam kategori = " + namaDokumen);

```

*List Code 4.4.6 Membandingkan hasil perkalian prior probabilities*

#### 4.5. Implementasi Trainer

Pada bagian *list code* ini, terdapat beberapa fungsi, yaitu mencari *term frequency* setiap dokumen, menghitung DF, menghitung IDF, menghitung W, menghitung jumlah W per kelas, menghitung jumlah W kata T per kelas, serta menghitung *laplace smoothing*.

```
//cari term freq tiap dok
for (int i = 0; i < namaDok.length; i++) {
// System.out.print("\n" + namaDok[i]);
```

```
res = res + "\n" + namaDok[i];
InvertedIndex a = new InvertedIndex();
String tempDok = ProsesData.openFile(path + "/", namaDok[i]);
StringTokenizer tok = new StringTokenizer(tempDok);
String temp = "";
while (tok.hasMoreTokens()) {
    temp = tok.nextToken();
    a.add(temp, namaDok[i]);
    xdf.add(temp, namaDok[i]);
}

for (int j = 0; j < term.length; j++) {
    termFreq[i][j] = a.cariKata(term[j]);
}
}

//menghitung DF
df = new int[term.length];
for (int i = 0; i < term.length; i++) {
    df[i] = xdf.cariDF(term[i]);
}

// menghitung IDF
for (int j = 0; j < term.length; j++) {
    iDF[j] = Math.log10((double)namaDok.length / (double)df[j]);

    jmlIDF = jmlIDF + iDF[j];
}

//menghitung W
for (int i = 0; i < namaDok.length; i++) {
    for (int j = 0; j < term.length; j++) {
        W[i][j] = termFreq[i][j] * (Math.log10(namaDok.length / df[j]));
        res = res + " " + W[i][j];
    }
}

// menghitung jumlah W per kelas
for (int i = 0; i < namaDok.length; i++) {
    for (int j = 0; j < term.length; j++) {
        if (namaDok[i].contains("ekonomi")) {
            wekonomi = wekonomi + W[i][j];
        }
        if (namaDok[i].contains("politik")) {
            wpolitik = wpolitik + W[i][j];
        }
        if (namaDok[i].contains("pendidikan")) {
            wpendidikan = wpendidikan + W[i][j];
        }
        if (namaDok[i].contains("kesehatan")) {
            wkesehatan = wkesehatan + W[i][j];
        }
    }
}

jmlW[0] = wekonomi;
jmlW[1] = wpolitik;
jmlW[2] = wpendidikan;
jmlW[3] = wkesehatan;
```

```
//menghitung jumlah W kata T per kelas

for (int i = 0; i < term.length; i++) {
    double eko = 0;
    double pol = 0;
    double pen = 0;
    double kes = 0;
    for (int j = 0; j < namaDok.length; j++) {
        if (namaDok[j].contains("ekonomi")) {
            eko = eko + W[j][i];
        }
        if (namaDok[j].contains("politik")) {
            pol = pol + W[j][i];
        }
        if (namaDok[j].contains("pendidikan")) {
            pen = pen + W[j][i];
        }
        if (namaDok[j].contains("kesehatan")) {
            kes = kes + W[j][i];
        }
    }
    wt[i][0] = eko;
    wt[i][1] = pol;
    wt[i][2] = pen;
    wt[i][3] = kes;
}

//menghitung Laplace Smooting
String saveLS="";
for (int i = 0; i < term.length; i++) {
    saveLS=saveLS+term[i].toString()+"=";
    for (int j = 0; j < 4; j++) {
        LS[i][j] = (wt[i][j] + 1) / (jmlW[j] + jmlIDF);
        saveLS=saveLS+" "+LS[i][j]+";";
    }
    saveLS=saveLS+"\n";
}
ProsesData.save(saveLS,      "src/"      +      "hasilTraining"      +      "/",
"hasilTraining.txt");
System.out.println("Training File Selesai");
res = res + "\n\nTraining File Selesai\n\n";
```

**List Code 4.5.1 Trainer**

## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1. Hasil Pengujian

Pengujian menggunakan *cross-validation* adalah dengan membagi data ke dalam *n-fold*. Nilai *n* dapat ditentukan sesuai dengan keinginan, dan pengelompokan data akan dilakukan secara random tetapi jumlah data dari tiap kelompok harus setara. Masing – masing kelompok akan mengalami posisi sebagai data *testing* (data uji) dan sebagai data *training* (data pelatihan) secara bergantian.

Pada pengujian ini, data yang tersedia akan dibagi secara merata menggunakan metode *cross validation*. Berikut 40 dokumen yang akan diuji :

**Tabel 5.1.** Daftar Seluruh Dokumen

ekonomi	pendidikan	politik	kesehatan
ekonomi(1)	pendidikan(1)	politik(1)	kesehatan(1)
ekonomi(2)	pendidikan(2)	politik(2)	kesehatan(2)
ekonomi(3)	pendidikan(3)	politik(3)	kesehatan(3)
ekonomi(4)	pendidikan(4)	politik(4)	kesehatan(4)
ekonomi(5)	pendidikan(5)	politik(5)	kesehatan(5)
ekonomi(6)	pendidikan(6)	politik(6)	kesehatan(6)
ekonomi(7)	pendidikan(7)	politik(7)	kesehatan(7)
ekonomi(8)	pendidikan(8)	politik(8)	kesehatan(8)
ekonomi(9)	pendidikan(9)	politik(9)	kesehatan(9)
ekonomi(10)	pendidikan(10)	politik(10)	kesehatan(10)

Pembagian data untuk *3-fold* adalah sebagai berikut :

**Tabel 5.2.** Pemetaan Data untuk *3-fold*

Fold 1	Fold 2	Fold 3
ekonomi(1)	ekonomi(4)	ekonomi(7)
ekonomi(2)	ekonomi(5)	ekonomi(8)
ekonomi(3)	ekonomi(6)	ekonomi(9)
kesehatan(1)	kesehatan(4)	ekonomi(10)
kesehatan(2)	kesehatan(5)	kesehatan(8)
kesehatan(3)	kesehatan(6)	kesehatan(9)
pendidikan(1)	kesehatan(7)	kesehatan(10)
pendidikan(2)	pendidikan(5)	pendidikan(8)
pendidikan(3)	pendidikan(6)	pendidikan(9)
pendidikan(4)	pendidikan(7)	pendidikan(10)
politik(1)	politik(4)	politik(8)
politik(2)	politik(5)	politik(9)
politik(3)	politik(6)	politik(10)
	politik(7)	

Skenario penggeraan *3 fold* adalah sebagai berikut :

**Tabel 5.3.** Fungsi Data *3 fold*

Tahap I	Tahap II	Tahap III
Fold 1 = testing	Fold 1 = training	Fold 1 = training
Fold 2 = training	Fold 2 = testing	Fold 2 = training
Fold 3 = training	Fold 3 = training	Fold 3 = testing

Pembagian data untuk *5-fold* adalah sebagai berikut :

**Tabel 5.4.** Pemetaan Data untuk *5-fold*

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
ekonomi(1)	ekonomi(3)	ekonomi(5)	ekonomi(7)	ekonomi(9)
ekonomi(2)	ekonomi(4)	ekonomi(6)	ekonomi(8)	ekonomi(10)
kesehatan(1)	kesehatan(3)	kesehatan(5)	kesehatan(7)	kesehatan(9)
kesehatan(2)	kesehatan(4)	kesehatan(6)	kesehatan(8)	kesehatan(10)
pendidikan(1)	pendidikan(3)	pendidikan(5)	pendidikan(7)	pendidikan(9)
pendidikan(2)	pendidikan(4)	pendidikan(6)	pendidikan(8)	pendidikan(10)
politik(1)	politik(3)	politik(5)	politik(7)	politik(9)
politik(2)	politik(4)	politik(6)	politik(8)	politik(10)

Pemetaan penggeraan *5 fold* adalah sebagai berikut :

**Tabel 5.5.** Fungsi Data *5 fold*

Tahap I	Tahap II	Tahap III	Tahap IV	Tahap V
Fold 1 = testing	Fold 1 = training			
Fold 2 = training	Fold 2 = testing	Fold 2 = training	Fold 2 = training	Fold 2 = training
Fold 3 = training	Fold 3 = training	Fold 3 = testing	Fold 3 = training	Fold 3 = training
Fold 4 = training	Fold 4 = training	Fold 4 = training	Fold 4 = testing	Fold 4 = training
Fold 5 = training	Fold 5 = testing			

### 5.1.1. Hasil Pengujian menggunakan Feature *tfidf* (W)

#### 1) 3-Fold menggunakan Feature *tfidf* (W)

Hasil 3 – *fold* menggunakan Feature *tfidf* (W)

**Tabel 5.6.** Hasil Klasifikasi 3 fold (feature W)

Tahap I		Tahap II		Tahap III	
Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi
ekonomi(1)	kesehatan	ekonomi(4)	kesehatan	ekonomi(7)	ekonomi
ekonomi(2)	ekonomi	ekonomi(5)	ekonomi	ekonomi(8)	ekonomi
ekonomi(3)	ekonomi	ekonomi(6)	ekonomi	ekonomi(9)	ekonomi
kesehatan(1)	kesehatan	kesehatan(4)	ekonomi	ekonomi(10)	ekonomi
kesehatan(2)	kesehatan	kesehatan(5)	kesehatan	kesehatan(8)	politik
kesehatan(3)	kesehatan	kesehatan(6)	kesehatan	kesehatan(9)	pendidikan
pendidikan(1)	pendidikan	kesehatan(7)	pendidikan	kesehatan(10)	kesehatan
pendidikan(2)	pendidikan	pendidikan(5)	pendidikan	pendidikan(8)	pendidikan
pendidikan(3)	pendidikan	pendidikan(6)	pendidikan	pendidikan(9)	pendidikan
pendidikan(4)	pendidikan	pendidikan(7)	pendidikan	pendidikan(10)	ekonomi
politik(1)	kesehatan	politik(4)	politik	politik(8)	politik
politik(2)	pendidikan	politik(5)	politik	politik(9)	politik
politik(3)	ekonomi	politik(6)	politik	politik(10)	pendidikan
		politik(7)	politik		

Akurasi 3 –fold Feature *tfidf* (W)

**Tabel 5.7.** Akurasi 3 fold (feature W)

	Jumlah dokumen testing	Jumlah dokumen relevan	Jumlah dokumen tidak relevan	Akurasi dokumen relevan (dalam %)	Akurasi dokumen tidak relevan (dalam %)
Tahap 1	13	9	4	69,23	30,77
Tahap 2	14	11	3	78,57	21,43
Tahap 3	13	8	5	61,54	38,46
Rata-rata				69,78	30,22

## 2) 5-Fold Cross Validation menggunakan Feature *tfidf* (W)

Hasil 5 –fold Feature *tfidf* (W):

**Tabel 5.8.** Hasil Klasifikasi 5 fold (feature W)

Tahap I		Tahap II		Tahap III	
Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi
ekonomi(1)	politik	ekonomi(3)	kesehatan	ekonomi(5)	ekonomi
ekonomi(2)	ekonomi	ekonomi(4)	kesehatan	ekonomi(6)	ekonomi
kesehatan(1)	politik	kesehatan(3)	kesehatan	kesehatan(5)	pendidikan
kesehatan(2)	kesehatan	kesehatan(4)	kesehatan	kesehatan(6)	kesehatan
pendidikan(1)	pendidikan	pendidikan(3)	pendidikan	pendidikan(5)	pendidikan
pendidikan(2)	pendidikan	pendidikan(4)	poltik	pendidikan(6)	pendidikan
politik(1)	politik	politik(3)	ekonomi	politik(5)	politik
politik(2)	politik	politik(4)	poltik	politik(6)	politik
Tahap IV		Tahap V			
Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi		
ekonomi(7)	ekonomi	ekonomi(9)	ekonomi		
ekonomi(8)	ekonomi	ekonomi(10)	ekonomi		
kesehatan(7)	ekonomi	kesehatan(9)	pendidikan		
kesehatan(8)	kesehatan	kesehatan(10)	kesehatan		
pendidikan(7)	politik	pendidikan(9)	pendidikan		
pendidikan(8)	pendidikan	pendidikan(10)	ekonomi		
politik(7)	politik	politik(9)	poltik		
politik(8)	politik	politik(10)	poltik		

Akurasi dari 5 –fold (*feature W*)

**Tabel 5.9.** Akurasi 5 fold (*feature W*)

	Jumlah dokumen testing	Jumlah dokumen relevan	Jumlah dokumen tidak relevan	Akurasi dokumen relevan (dalam %)	Akurasi dokumen tidak relevan (dalam %)
Tahap 1	8	6	2	75	25
Tahap 2	8	4	4	50	50
Tahap 3	8	7	1	87,5	12,5
Tahap 4	8	7	1	87,5	12,5
Tahap 5	8	7	1	87,5	12,5
Rata-rata				77,5	22,5

### 5.1.2. Hasil Pengujian menggunakan *Feature tf*

#### 1) 3-Fold Cross Validation menggunakan *Feature tf*

Hasil dari 3 –fold (*feature tf*)

**Tabel 5.10.** Hasil Klasifikasi 3 fold (*feature tf*)

Tahap I		Tahap II		Tahap III	
Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi
ekonomi(1)	politik	ekonomi(4)	pendidikan	ekonomi(7)	pendidikan
ekonomi(2)	ekonomi	ekonomi(5)	ekonomi	ekonomi(8)	pendidikan
ekonomi(3)	ekonomi	ekonomi(6)	ekonomi	ekonomi(9)	ekonomi
kesehatan(1)	kesehatan	kesehatan(4)	ekonomi	ekonomi(10)	pendidikan
kesehatan(2)	kesehatan	kesehatan(5)	pendidikan	kesehatan(8)	kesehatan
kesehatan(3)	kesehatan	kesehatan(6)	kesehatan	kesehatan(9)	kesehatan
pendidikan(1)	pendidikan	kesehatan(7)	kesehatan	kesehatan(10)	kesehatan
pendidikan(2)	pendidikan	pendidikan(5)	pendidikan	pendidikan(8)	pendidikan
pendidikan(3)	ekonomi	pendidikan(6)	pendidikan	pendidikan(9)	pendidikan
pendidikan(4)	pendidikan	pendidikan(7)	pendidikan	pendidikan(10)	pendidikan
politik(1)	politik	politik(4)	pendidikan	politik(8)	politik
politik(2)	politik	politik(5)	pendidikan	politik(9)	politik
politik(3)	politik	politik(6)	politik	politik(10)	politik
		politik(7)	politik		

Akurasi dari 3 –fold (*feature tf*)

**Tabel 5.11.** Akurasi 3 fold (*feature tf*)

	Jumlah dokumen <i>testing</i>	Jumlah dokumen relevan	Jumlah dokumen tidak relevan	Akurasi dokumen relevan (dalam %)	Akurasi dokumen tidak relevan (dalam %)
Tahap 1	13	11	2	84,62	15,38
Tahap 2	14	10	4	71,43	28,57
Tahap 3	13	10	3	76,92	23,08
Rata-rata				77,66	22,34

## 2) 5-Fold Cross Validation menggunakan Feature tf

Hasil dari 5 – fold cross validation :

**Tabel 5.12.** Hasil Klasifikasi 5 fold (*feature tf*)

Tahap I		Tahap II		Tahap III	
Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi
ekonomi(1)	politik	ekonomi(3)	ekonomi	ekonomi(5)	ekonomi
ekonomi(2)	ekonomi	ekonomi(4)	pendidikan	ekonomi(6)	ekonomi
kesehatan(1)	kesehatan	kesehatan(3)	kesehatan	kesehatan(5)	pendidikan
kesehatan(2)	kesehatan	kesehatan(4)	ekonomi	kesehatan(6)	kesehatan
pendidikan(1)	pendidikan	pendidikan(3)	pendidikan	pendidikan(5)	pendidikan
pendidikan(2)	pendidikan	pendidikan(4)	pendidikan	pendidikan(6)	pendidikan
politik(1)	politik	politik(3)	pendidikan	politik(5)	politik
politik(2)	politik	politik(4)	politik	politik(6)	politik
Tahap IV		Tahap V			
Dokumen	Hasil Klasifikasi	Dokumen	Hasil Klasifikasi		
ekonomi(7)	ekonomi	ekonomi(9)	ekonomi		
ekonomi(8)	ekonomi	ekonomi(10)	ekonomi		
kesehatan(7)	kesehatan	kesehatan(9)	kesehatan		
kesehatan(8)	kesehatan	kesehatan(10)	kesehatan		
pendidikan(7)	pendidikan	pendidikan(9)	pendidikan		
pendidikan(8)	pendidikan	pendidikan(10)	pendidikan		
politik(7)	politik	politik(9)	politik		
politik(8)	politik	politik(10)	ekonomi		

Akurasi dari 3 –fold (*feature tf*)

**Tabel 5.13.** Akurasi 3 fold (*feature tf*)

	Jumlah dokumen <i>testing</i>	Jumlah dokumen relevan	Jumlah dokumen tidak relevan	Akurasi dokumen relevan (dalam %)	Akurasi dokumen tidak relevan (dalam %)
Tahap 1	8	7	1	87,5	12,5
Tahap 2	8	6	2	75	25
Tahap 3	8	7	1	87,5	12,5
Tahap 4	8	8	0	100	0
Tahap 5	8	7	1	87,5	12,5
Rata-rata				87,5	12,5

### 5.1.3. Analisa Hasil

Berdasarkan percobaan yang telah dilakukan, persentase 3-fold menggunakan *feature tfidif* persentase benar 69,78%, dan salah 30,77%. Sedangkan 5-fold persentase benar 77,66% dan salah 22,5%.

**Tabel 5.14.** Akurasi Klasifikasi *feature tf* dan *tf-idf*

	3- fold	5-fold
<i>Feature tfidf</i>	benar 69,78 %	benar 77,5 %
	salah 30,77 %	salah 22,5 %
<i>Feature tf</i>	benar 77,66 %	benar 87,5 %
	salah 22,34 %	salah 12,5 %

Selain itu, keterkaitan antar kata dalam setiap kelas juga mempengaruhi presentase.

Proses *matching* mempengaruhi nilai akhir yang didapatkan, karena akan menggunakan nilai *tf* yang didapat dari langkah *matching* sebagai pemangkat dari *laplace smoothing* yang telah dihitung. Semakin tinggi nilai *tf* yang ditemukan pada proses *macthing*, maka nilai *laplace smoothing* akan semakin kecil.

Pada *feature tf*, semakin sering sebuah kata muncul di suatu dokumen, semakin relevan kata tersebut dalam mempresentasikan kelas tersebut. Namun, penggunaan *tf-idf* dalam klasifikasi teks tidak efektif karena menggunakan *inverse* dari *term frequency*, sehingga semakin sering sebuah kata muncul di kumpulan dokumen *training*, semakin tidak efektif dalam membedakan satu dokumen dengan dokumen lain.

## BAB VI

### KESIMPULAN DAN SARAN

Bagian ini memberikan kesimpulan dan saran berdasarkan hasil penelitian yang telah dilakukan.

#### 6.1. Kesimpulan

Kesimpulan yang dapat diambil dari pembangunan sistem klasifikasi bahasa Jawa menggunakan metode Naïve Bayes adalah sebagai berikut :

1. Berdasarkan percobaan yang telah dilakukan, persentase *3-fold* menggunakan *feature tfidf* persentase benar 69,78%, dan salah 30,77%. Sedangkan *5-fold* persentase benar 77,5% dan salah 22,5%.
2. Nilai *5-fold* lebih besar dibandingkan nilai *3-fold* dipengaruhi oleh jumlah data *training* yang digunakan.
3. Banyaknya dokumen yang digunakan dalam proses *testing* ataupun *training* akan mempengaruhi hasil klasifikasi.

#### 6.2. Saran

Beberapa saran yang berguna untuk memperbaiki sistem :

1. Menambah daftar stoplist, sehingga kemunculan kata unik akan lebih sedikit.

## DAFTAR PUSTAKA

- Davies, J., & Goker, A. (2009). *Information Retrieval: Searching in the 21st Century*. A John Wiley and Sons, Ltd.
- Feldman, Ronen & James Sanger. 2007. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, Cambridge University Press.
- Grossma, David A., & Ophir Frieder. 2004. *Information Retrieval Algorithms And Heuristics, 2nd edition*, Springer.
- Han, J. & Kamber, M. 2006. *Second Edition : Data Mining concepts and Techniques*.
- Hanopo, F.S. (2013). *Klasifikasi Surat Masuk menggunakan Multinomial Naïve Bayes*. Naskah skripsi yang tidak diterbitkan, Yogyakarta : Universitas Sanata Dharma.
- Joachims, T. (1997). *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. International Conference on Machine Learning (ICML).
- Manning, Christopher D., Prabhakar Raghavan and Hinrich Schütze. 2008. *Introduction to Information Retrieval*, Cambridge University Press.
- Salton, Gerard. 1983. *Introduction to Modern Information Retrieval*, McGraw Hill
- Widjono,S.H.,Darmawan,J.B.,& Adji,S.E. (2011-2012). *Pengaruh Stemming untuk Perolehan Informasi dalam Bahasa Jawa*.Penelitian Hibah Pekerti DIKTI.
- Witten, I. H., & Frank, E. (2005). *Data Mining: practical machine learning tools and techniques, 2nd edition*. Morgan Kaufmann.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI



## LAMPIRAN 1

A. Berikut adalah tahap *pre-processing* :

1. Pendidikan1

**Tabel *pre-processing* Pendidikan1**

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
sasi	akeh	akeh	akeh		akeh	akeh	1	akeh	1
mei	ana	ana	angslup		angslup	angslup	1	angslup	1
wis	angslup	angslup	arep		arep	arep	1	arep	1
arep	arep	arep	asale	-e	asal	asal	1	asal	1
angslup	asale	asale	asing		asing	asing	1	asing	1
tanggal	asing	asing	dadakan	-an	dadak	dadak	1	dadak	1
mei	dadakan	dadakan	dhaerah		dhaerah	dhaerah	1	dhaerah	1
wis	dhaerah	dhaerah	ditindakake		ditindakake	ditindakake	1	dhidhik	2
wiwit	ditindakake	ditindakake	hardhiknas		hardhiknas	giyat	1	ditindakake	1
kesilep	endi	endi	hardhiknas		hardhiknas	giyat	1	giyat	2
nanging	hardhiknas	hardhiknas	hari		hari	hardhiknas	1	hardhiknas	2
kegiyatan	hardhiknas	hardhiknas	indonesia		indonesia	hardhiknas	1	hari	1
hardhiknas	hari	hari	kabar		kabar	hari	1	indonesia	1
hari	indonesia	indonesia	kahanan		kahanan	indonesia	1	kabar	1
pendhidhikan	ing	ing	kegiyatan	ke-an	giyat	kabar	1	kahanan	1
isih	ing	ing	kegiyatan	ke-an	giyat	kahanan	1	marak	1
katon	ing	ing	kesilep	ke-	silep	lorot	1	mei	2
marak	isih	isih	marak		marak	marak	1	melorot	1

## PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

71

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
ing	kabar	kabar	mei		mei	mei	1	merosot	1
saben	kahanan	kahanan	mei		mei	mei	1	mudhun	1
dhaerah	katon	katon	melorot		melorot	merosot	1	pameran	1
akeh	kegiyatan	kegiyatan	merosot		merosot	mudhun	1	sangkut	1
pameran	kegiyatan	kegiyatan	mudhun		mudhun	pameran	1	sebar	1
lan	kesilep	kesilep	nyangkut	ny=s	sangkut	dhidhik	1	sebut	1
kegiyatan	lan	lan	nyebutke	ny=s; -ke	sebut	dhidhik	1	silep	1
sing	lan	lan	pameran		pameran	sangkut	1	statistik	1
nyangkut	marak	marak	pendhidhikan		dhidhik	sebar	1	tanggal	1
hardhiknas	mau	mau	pendhidhikan		dhidhik	sebut	1	tengah	1
mau	mau	mau	statistik		statistik	silep	1	wiwit	1
ditindakake	mei	mei	sumebar	^um	sebar	statistik	1		
ing	mei	mei	tanggal		tanggal	tanggal	1		
ngendi	melorot	melorot	tengah		tengah	tengah	1		
endi	merosot	merosot	wiwit		wiwit	wiwit	1		
lan	mudhun	mudhun							
ing	nanging	nanging							
tengah	ngendi	ngendi							
kahanan	nyangkut	nyangkut							
mau	nyebutke	nyebutke							
dadakan	pameran	pameran							
ana	pendhidhikan	pendhidhikan							
kabar	pendhidhikan	pendhidhikan							
sing	saben	saben							
sumebar	saka	saka							
sing	sasi	sasi							
asale	saya	saya							
saka	saya	saya							

71

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

72

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
statistik	sing	sing							
asing	sing	sing							
nyebutke	sing	sing							
pendhidhikan	statistik	statistik							
indonesia	sumebar	sumebar							
saya	tanggal	tanggal							
merosot	tengah	tengah							
saya	wis	wis							
melorot	wis	wis							
mudhun	wiwit	wiwit							

## 2. Pendidikan2

**Tabel pre-processing Pendidikan2**

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
kanggo	anane	anane	awujud	a-	wujud	angkah	1	angkah	1
biyantu	awujud	awujud	biyantu		biyantu	biyantu	1	biyantu	1
ningkatake	bisa	bisa	dewan		dewan	dewan	1	dewan	1
kualitas	biyantu	biyantu	dpk		dpk	dhapuk	1	dhapuk	1
pendhidhikan	dewan	dewan	dpk		dpk	dpk	1	dhidhik	3
ing	dpk	dpk	kaangkah	ka-	angkah	dpk	1	dpk	2
kabupaten	dpk	dpk	kabupaten		kabupaten	kabupaten	1	kabupaten	4
sleman	durung	durung	kabupaten		kabupaten	kabupaten	1	kritik	1
durung	iki	iki	kabupaten		kabupaten	kabupaten	1	kualitas	1
suwe	ing	ing	kabupaten		kabupaten	kabupaten	1	masarakat	1
iki	ing	ing	kadhapuk	ka-	dhapuk	kritik	1	meneh	1

72

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

73

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
kadhapuk	ing	ing	kasebut	ka-	sebut	kualitas	1	mutu	1
pengurus	kaangkah	kaangkah	kritik		kritik	masarakat	1	saran	1
dewan	kabupaten	kabupaten	kualitas		kualitas	meneh	1	sebut	1
pendhidhikan	kabupaten	kabupaten	masarakat		masarakat	mutu	1	sleman	4
kabupaten	kabupaten	kabupaten	menehi	-i	meneh	dhidhik	1	sumbangan	1
dpk	kabupaten	kabupaten	mutune	-ne	mutu	dhidhik	1	tingkat	2
sleman	kadhapuk	kadhapuk	ningkatake	n=t; -ake	tingkat	dhidhik	1	tujuane	1
kanthi	kang	kang	ningkatake	n=t; -ake	tingkat	saran	1	urus	1
anane	kanggo	kanggo	pendhidhikan		dhidhik	sebut	1	wujud	1
dpk	kanggo	kanggo	pendhidhikan		dhidhik	sleman	1		
kasebut	kanthi	kanthi	pendhidhikan		dhidhik	sleman	1		
kaangkah	kasebut	kasebut	pengurus	peng-	urus	sleman	1		
masarakat	kritik	kritik	saran		saran	sleman	1		
ing	kualitas	kualitas	sleman		sleman	sumbangan	1		
kabupaten	lan	lan	sleman		sleman	tingkat	1		
sleman	liya	liya	sleman		sleman	tingkat	1		
bisa	liyane	liyane	sleman		sleman	tujuane	1		
menehi	masarakat	masarakat	sumbangan		sumbangan	urus	1		
sumbangan	menehi	menehi	tujuane		tujuane	wujud	1		
awujud	mutune	mutune							
saran	ningkatake	ningkatake							
kritik	ningkatake	ningkatake							
lan	pendhidhikan	pendhidhikan							
liya	pendhidhikan	pendhidhikan							
liyane	pendhidhikan	pendhidhikan							
kang	pengurus	pengurus							
tujuane	saran	saran							
kanggo	sleman	sleman							
ningkatake	sleman	sleman							

73

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

74

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
mutune	sleman	sleman							
pendhidhikan	sleman	sleman							
ing	sumbangan	sumbangan							
kabupaten	suwe	suwe							
sleman	tujuane	tujuane							

### 3. Politik1

**Tabel pre-processing Politik1**

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
indonesia	aceh	aceh	aceh		aceh	aceh	1	aceh	1
lagi	akeh	akeh	akeh		akeh	akeh	1	akeh	1
ribet	ana	ana	anggota		anggota	anggota	1	anggota	1
propinsi	anggota	anggota	barang		barang	barang	1	barang	1
aceh	barang	barang	gam		gam	berontak	1	berontak	1
lagi	dadi	dadi	gam		gam	mbrasta	1	gam	3
panas	dha	dha	gam		gam	gam	1	gampang	1
perang	dudu	dudu	gampang		gampang	gam	1	gugur	1
tni	gam	gam	gugur		gugur	gam	1	indonesia	1
lumawan	gam	gam	indonesia		indonesia	gampang	1	kaum	1
kelompok	gam	gam	kaum		kaum	gugur	1	kelangan	1
mbalela	gampang	gampang	kelangan		kelangan	kelangan	1	kelompok	1
separatis	gugur	gugur	kelompok		kelompok	indonesia	1	lumawan	1
gam	indonesia	indonesia	lumawan		lumawan	kaum	1	mbalela	1
sing	kaum	kaum	mbalela		mbalela	kelompok	1	mbrasta	1
dha	kaya	kaya	mbrasta		mbrasta	lawan	1	nalar	1
gugur	kejaba	kejaba	nalare	-e	nalar	mbalela	1	nyawa	1

74

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

75

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
wis	kelangan	kelangan	nyawa		nyawa	nalar	1	panas	1
akeh	kelompok	kelompok	panas		panas	nyawa	1	pemerintah	1
kejaba	kuwi	kuwi	pemberontakan	pem-an	berontak	panas	1	perang	1
wong	lagi	lagi	pemerintah		pemerintah	pemerintah	1	polri	1
wong	lagi	lagi	perang		perang	perang	1	propinsi	1
gam	lan	lan	polri		polri	polri	1	ribet	1
anggota	lumawan	lumawan	propinsi		propinsi	propinsi	1	separatis	1
tni	mau	mau	ribet		ribet	ribet	1	tni	3
utawa	mbalela	mbalela	separatis		separatis	separatis	1	tumbal	1
polri	mbrasta	mbrasta	tni		tni	tni	1		
wis	nalare	nalare	tni		tni	tni	1		
ana	nyawa	nyawa	tni		tni	tni	1		
sing	panas	panas	tumbal		tumbal	tumbal	1		
dadi	pemberontakan	pemberontakan							
tumbal	pemerintah	pemerintah							
kelangan	perang	perang							
nyawa	polri	polri							
nalare	propinsi	propinsi							
tumrape	ribet	ribet							
tni	separatis	separatis							
lan	sing	sing							
pemerintah	sing	sing							
mbrasta	sing	sing							
kaum	tni	tni							
pemberontakan	tni	tni							
kaya	tni	tni							
gam	tumbal	tumbal							
kuwi	tumrape	tumrape							
mau	utawa	utawa							

75

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

76

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
dudu	wis	wis							
barang	wis	wis							
sing	wong	wong							
gampang	wong	wong							

## 4. Politik2

**Tabel pre-processing Politik2**

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
sawise	akeh	akeh	akeh		akeh	akeh	1	akeh	1
ambruke	ambruke	ambruke	ambruke		ambruke	ambruke	1	ambruke	2
uni	ambruke	ambruke	ambruke		ambruke	ambruke	1	amerika	2
soviet	amerika	amerika	amerika		amerika	amerika	1	ancam	1
utawa	amerika	amerika	amerika		amerika	amerika	1	balkan	1
ussr	balkan	balkan	balkan		balkan	ancam	1	cacah	1
uni	cacah	cacah	cacah		cacah	balkan	1	cecoslowakia	1
soviet	cecoslowakia	cecoslowakia	cecoslowakia		cecoslowakia	cacah	1	cina	2
sosialis	cina	cina	cina		cina	cecoslowakia	1	disintegrasi	1
republik	cina	cina	cina		cina	cina	1	eropa	1
taun	dene	dene	disintegrasi		disintegrasi	cina	1	indonesia	2
sing	disintegrasi	disintegrasi	ditututi	di-i	tutut	disintegrasi	1	kahanan	1
ditututi	ditututi	ditututi	eropa		eropa	eropa	1	katon	1
negara	durung	durung	indonesia		indonesia	indonesia	1	kukuh	1
negara	eropa	eropa	indonesia		indonesia	indonesia	1	laladan	1
uni	iki	iki	kaancam	ka-	ancam	kahanan	1	lamun	1
ing	indonesia	indonesia	kahanan		kahanan	katon	1	mutawatiri	1

76

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

77

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
laladan	indonesia	indonesia	kukuh		kukuh	kukuh	1	negara	6
balkan	ing	ing	laladan		laladan	laladan	1	pecah	1
eropa	isih	isih	lamun		lamun	lamun	1	pranyata	1
tenggara	kaancam	kaancam	mutawatiri		mutawatiri	mutawatiri	1	ramal	1
kaya	kahanan	kahanan	negara		negara	negara	1	republik	1
cekoslowakia	kaya	kaya	negara		negara	negara	1	ringkikh	1
lan	klebu	klebu	negara		negara	negara	1	serikat	3
yugoslavia	kukuh	kukuh	negara		negara	negara	1	sosialis	1
akeh	laladan	laladan	negara		negara	negara	1	soviet	2
ramalan	lamun	lamun	negara		negara	negara	1	tenggara	1
lamun	lan	lan	ngatonake	ng=k; -ake	katon	pecah	1	tutut	1
negara	lan	lan	perpecahan	per-an	pecah	pranyata	1	uni	6
uni	lan	lan	pranyata		pranyata	ramal	1	ussr	1
serikat	minangka	minangka	ramalan	-an	ramal	republik	1	yugoslavia	1
sing	mutawatiri	mutawatiri	republik		republik	ringkikh	1		
kaancam	negara	negara	ringkikh		ringkikh	serikat	1		
disintegrasi	negara	negara	serikat		serikat	serikat	1		
perpecahan	negara	negara	serikat		serikat	serikat	1		
yaiku	negara	negara	serikat		serikat	sosialis	1		
amerika	negara	negara	sosialis		sosialis	soviet	1		
serikat	negara	negara	soviet		soviet	soviet	1		
cina	ngatonake	ngatonake	soviet		soviet	tenggara	1		
lan	paling	paling	tenggara		tenggara	tutut	1		
indonesia	paling	paling	uni		uni	uni	1		
saka	perpecahan	perpecahan	uni		uni	uni	1		
negara	pranyata	pranyata	uni		uni	uni	1		
uni	ramalan	ramalan	uni		uni	uni	1		
cacah	republik	republik	uni		uni	uni	1		
telu	ringkikh	ringkikh	uni		uni	uni	1		

77

## PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

78

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
iki	saka	saka	ussr		ussr	ussr	1		
pranyata	sawise	sawise	yugoslavia		yugoslavia	yugoslavia	1		
sing	serikat	serikat							
paling	serikat	serikat							
ringkih	serikat	serikat							
ambruke	sing	sing							
yaiku	sing	sing							
indonesia	sing	sing							
dene	sosialis	sosialis							
amerika	soviet	soviet							
serikat	soviet	soviet							
isih	taun	taun							
klebu	telu	telu							
negara	tenggara	tenggara							
paling	uni	uni							
kukuh	uni	uni							
minangka	uni	uni							
negara	uni	uni							
uni	uni	uni							
lan	uni	uni							
cina	ussr	ussr							
durung	utawa	utawa							
ngatonake	yaiku	yaiku							
kahanan	yaiku	yaiku							
mutawatiri	yugoslavia	yugoslavia							

78

## 5. Testing

**Tabel pre-processing Testing**

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
jaman	aksara	aksara	aksara		aksara	aksara	1	aksara	2
saiki	aksara	aksara	aksara		aksara	aksara	1	aneh	1
pendhidhikan	aneh	aneh	aneh		aneh	aneh	1	babag	1
wus	awit	awit	babagan	-an	babag	babag	1	barang	1
dudu	bab	bab	barang		barang	barang	1	basa	1
bab	bab	bab	basa		basa	basa	1	cak	1
sing	babagan	babagan	cak		cak	cak	1	cakane	1
aneh	barang	barang	cakane		cakane	cakane	1	dasar	1
nanging	basa	basa	dasar		dasar	dasar	1	dhidhik	1
dadi	cak	cak	dhuwur		dhuwur	dhuwur	1	dhuwur	1
barang	cakane	cakane	jaman		jaman	jaman	1	jaman	1
sing	dadi	dadi	jawa		jawa	jawa	1	jawa	3
larang	dasar	dasar	jawa		jawa	jawa	1	kaji	1
regane	dhuwur	dhuwur	jawa		jawa	jawa	1	kurikulum	2
dhuwur	dudu	dudu	kurikulum		kurikulum	kaji	1	lapang	1
pangajine	durung	durung	kurikulum	-e	kurikulum	kurikulum	1	larang	1
mung	durung	durung	lapangan		lapang	kurikulum	1	laras	1
wae	durung	durung	larang		larang	lapang	1	mligi	2
mutune	ing	ing	laras		laras	larang	1	mutu	1
durung	ing	ing	mligine	-ne	mligi	laras	1	prakteke	1
mesthi	ing	ing	mligine	-ne	mligi	laras	1	rega	1
kurikulum	jaman	jaman	mutune	-ne	mutu	mligi	1	sekolah	2
ing	jawa	jawa	pangajine	pang-k; -ne	kaji	mligi	1	selaras	1
sekolah	jawa	jawa	pendhidhikan	pen-an	dhidhik	mutu	1	tingkat	1
mligine	jawa	jawa	prakteke		prakteke	dhidhik	1	wulang	2
ing	karo	kare	regane	-ne	rega	prakteke	1		

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

80

Tokenisasi & Case Folding	sorting	penghilangan stopword	hasil stopword	stemming	hasil stemming	term jadi	tf	term jadi	tf
tingkat	karo	kare	sekolah		sekolah	regae	1		
sekolah	kurikulum	kurikulum	sekolah		sekolah	sekolah	1		
dasar	kurikulume	kurikulume	selaras	se-	laras	sekolah	1		
wulangan	lapangan	lapangan	tingkat		tingkat	tingkat	1		
basa	larang	larang	wulangan	-an	wulang	wulang	1		
jawa	laras	laras	wulangan	-an	wulang	wulang	1		
babagan	mesthi	mesthi							
aksara	mligine	mligine							
jawa	mligine	mligine							
durung	mung	mung							
selaras	mutune	mutune							
karo	nanging	nanging							
cak	pangajine	pangajine							
cakane	pendhidhikan	pendhidhikan							
utawa	prakteke	prakteke							
prakteke	regane	regane							
awit	saiki	saiki							
ing	sekolah	sekolah							
lapangan	sekolah	sekolah							
wulangan	selaras	selaras							
mligine	sing	sing							
bab	sing	sing							
aksara	tingkat	tingkat							
durung	wae	wae							
laras	wulangan	wulangan							
karo	wulangan	wulangan							
kurikulume	wus	wus							

80

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

81

- B. Menghitung  $df$ ,  $idf$  dan  $W$  masing-masing dokumen

**Tabel perhitungan  $df$ ,  $idf$  dan  $W$**

term	tf				df	idf	W				$\Sigma W$ kata t	
	d1	d2	d3	d4			d1	d2	d3	d4	pendidikan	politik
a	b	c	d	e	f	g	h	i	j	k	l	m
aceh	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
akeh	1	0	1	1	3	0,12494	0,12494	0	0,12494	0,12494	0,12494	0,24988
ambruke	0	0	0	2	1	0,60206	0	0	0	1,20412	0	1,20412
amerika	0	0	0	2	1	0,60206	0	0	0	1,20412	0	1,20412
ancam	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
anggota	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
angkah	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
angslup	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
arep	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
asal	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
asing	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
balkan	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
barang	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
berontak	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
biyantu	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
cacah	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
cekoslowakia	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
cina	0	0	0	2	1	0,60206	0	0	0	1,20412	0	1,20412
dadak	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
dewan	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
dhaerah	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
dhapuk	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
dhidhik	2	3	0	0	2	0,30103	0,60206	0,90309	0	0	1,50515	0

81

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

82

term	tf				df	idf	W				$\Sigma W$ kata t	
	d1	d2	d3	d4			d1	d2	d3	d4	pendidikan	politik
a	b	c	d	e	f	g	h	i	j	k	l	m
disintegrasi	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
ditindakake	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
dpk	0	2	0	0	1	0,60206	0	1,20412	0	0	1,20412	0
eropa	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
gam	0	0	3	0	1	0,60206	0	0	1,80618	0	0	1,80618
gampang	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
giyat	2	0	0	0	1	0,60206	1,20412	0	0	0	1,20412	0
gugur	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
hardhiknas	2	0	0	0	1	0,60206	1,20412	0	0	0	1,20412	0
hari	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
indonesia	1	0	1	2	3	0,12494	0,12494	0	0,12494	0,24988	0,12494	0,37482
kabar	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
kabupaten	0	4	0	0	1	0,60206	0	2,40824	0	0	2,40824	0
kahanan	1	0	0	1	2	0,30103	0,30103	0	0	0,30103	0,30103	0,30103
katon	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
kaum	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
kelangan	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
kelompok	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
kritik	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
kualitas	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
kukuh	0	0	0	1	1	0,60206	0	0	0,60206	0	0	0,60206
laladan	0	0	0	1	1	0,60206	0	0	0,60206	0	0	0,60206
lamun	0	0	0	1	1	0,60206	0	0	0,60206	0	0	0,60206
lumawan	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
marak	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
masarakat	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0

82

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

83

term	tf				df	idf	W				$\Sigma W$ kata t	
	d1	d2	d3	d4			d1	d2	d3	d4	pendidikan	politik
a	b	c	d	e	f	g	h	i	j	k	l	m
mbalela	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
mbrasta	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
mei	2	0	0	0	1	0,60206	1,20412	0	0	0	1,20412	0
melorot	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
meneh	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
merosot	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
mudhun	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
mutawatiri	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
mutu	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
nalar	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
negara	0	0	0	6	1	0,60206	0	0	0	3,61236	0	3,61236
nyawa	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
pameran	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
panas	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
pecah	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
pemerintah	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
perang	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
polri	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
pranyata	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
propinsi	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
ramal	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
republik	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
ribet	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
ringkih	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
sangkut	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
saran	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0

83

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

84

term	tf				df	idf	W				$\Sigma W$ kata t	
	d1	d2	d3	d4			d1	d2	d3	d4	pendidikan	politik
a	b	c	d	e	f	g	h	i	j	k	l	m
sebar	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
sebut	1	1	0	0	2	0,30103	0,30103	0,30103	0	0	0,60206	0
separatis	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
serikat	0	0	0	3	1	0,60206	0	0	0	1,80618	0	1,80618
silep	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
sleman	0	4	0	0	1	0,60206	0	2,40824	0	0	2,40824	0
sosialis	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
soviet	0	0	0	2	1	0,60206	0	0	0	1,20412	0	1,20412
statistik	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
sumbangan	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
tanggal	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
tengah	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
tenggara	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
tingkat	0	2	0	0	1	0,60206	0	1,20412	0	0	1,20412	0
tni	0	0	3	0	1	0,60206	0	0	1,80618	0	0	1,80618
tujuane	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
tumbal	0	0	1	0	1	0,60206	0	0	0,60206	0	0	0,60206
tutut	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
uni	0	0	0	6	1	0,60206	0	0	0	3,61236	0	3,61236
urus	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
ussr	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206
wiwit	1	0	0	0	1	0,60206	0,60206	0	0	0	0,60206	0
wujud	0	1	0	0	1	0,60206	0	0,60206	0	0	0,60206	0
yugoslavia	0	0	0	1	1	0,60206	0	0	0	0,60206	0	0,60206

84

### C. Menghitung Laplace Smoothing

Dari tabel sebelumnya didapat nilai :

$\Sigma W$ (pendidikan)	34,56730
$\Sigma W$ (politik)	44,27404
$\Sigma idf$	57,746607

**Tabel Laplace Smoothing**

term	idf	$\Sigma W$ kata t		LS	
		pendidikan	politik	pendidikan	politik
a	g	l	m	n	o
aceh	0,60206	0	0,60206	0,01083	0,01570
akeh	0,12494	0,12494	0,24988	0,01219	0,01225
ambruke	0,60206	0	1,20412	0,01083	0,02160
amerika	0,60206	0	1,20412	0,01083	0,02160
ancam	0,60206	0	0,60206	0,01083	0,01570
anggota	0,60206	0	0,60206	0,01083	0,01570
angkah	0,60206	0,60206	0	0,01735	0,00980
angslup	0,60206	0,60206	0	0,01735	0,00980
arep	0,60206	0,60206	0	0,01735	0,00980
asal	0,60206	0,60206	0	0,01735	0,00980
asing	0,60206	0,60206	0	0,01735	0,00980
balkan	0,60206	0	0,60206	0,01083	0,01570
barang	0,60206	0	0,60206	0,01083	0,01570
berontak	0,60206	0	0,60206	0,01083	0,01570
biyantu	0,60206	0,60206	0	0,01735	0,00980
cacah	0,60206	0	0,60206	0,01083	0,01570
cekoslowakia	0,60206	0	0,60206	0,01083	0,01570
cina	0,60206	0	1,20412	0,01083	0,02160
dadak	0,60206	0,60206	0	0,01735	0,00980
dewan	0,60206	0,60206	0	0,01735	0,00980
dhaerah	0,60206	0,60206	0	0,01735	0,00980
dhapuk	0,60206	0,60206	0	0,01735	0,00980
dhidhik	0,30103	1,50515	0	0,02714	0,00980
disintegrasi	0,60206	0	0,60206	0,01083	0,01570
ditindakake	0,60206	0,60206	0	0,01735	0,00980
dpk	0,60206	1,20412	0	0,02388	0,00980
eropa	0,60206	0	0,60206	0,01083	0,01570
gam	0,60206	0	1,80618	0,01083	0,02751
gampang	0,60206	0	0,60206	0,01083	0,01570
giyat	0,60206	1,20412	0	0,02388	0,00980
gugur	0,60206	0	0,60206	0,01083	0,01570
hardhiknas	0,60206	1,20412	0	0,02388	0,00980
hari	0,60206	0,60206	0	0,01735	0,00980
indonesia	0,12494	0,12494	0,37482	0,01219	0,01348

# PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

86

term	idf	$\Sigma W_{kata\ t}$		LS	
		pendidikan	politik	pendidikan	politik
a	g	l	m	n	o
kabar	0,60206	0,60206	0	0,01735	0,00980
kabupaten	0,60206	2,40824	0	0,03692	0,00980
kahanan	0,30103	0,30103	0,30103	0,01409	0,01275
katon	0,60206	0	0,60206	0,01083	0,01570
kaum	0,60206	0	0,60206	0,01083	0,01570
kelangan	0,60206	0	0,60206	0,01083	0,01570
kelompok	0,60206	0	0,60206	0,01083	0,01570
kritik	0,60206	0,60206	0	0,01735	0,00980
kualitas	0,60206	0,60206	0	0,01735	0,00980
kukuh	0,60206	0	0,60206	0,01083	0,01570
laladan	0,60206	0	0,60206	0,01083	0,01570
lamun	0,60206	0	0,60206	0,01083	0,01570
lumawan	0,60206	0	0,60206	0,01083	0,01570
marak	0,60206	0,60206	0	0,01735	0,00980
masarakat	0,60206	0,60206	0	0,01735	0,00980
mbalela	0,60206	0	0,60206	0,01083	0,01570
mbrasta	0,60206	0	0,60206	0,01083	0,01570
mei	0,60206	1,20412	0	0,02388	0,00980
melorot	0,60206	0,60206	0	0,01735	0,00980
meneh	0,60206	0,60206	0	0,01735	0,00980
merosot	0,60206	0,60206	0	0,01735	0,00980
mudhun	0,60206	0,60206	0	0,01735	0,00980
mutawatiri	0,60206	0	0,60206	0,01083	0,01570
mutu	0,60206	0,60206	0	0,01735	0,00980
nalar	0,60206	0	0,60206	0,01083	0,01570
negara	0,60206	0	3,61236	0,01083	0,04521
nyawa	0,60206	0	0,60206	0,01083	0,01570
pameran	0,60206	0,60206	0	0,01735	0,00980
panas	0,60206	0	0,60206	0,01083	0,01570
pecah	0,60206	0	0,60206	0,01083	0,01570
pemerintah	0,60206	0	0,60206	0,01083	0,01570
perang	0,60206	0	0,60206	0,01083	0,01570
polri	0,60206	0	0,60206	0,01083	0,01570
pranyata	0,60206	0	0,60206	0,01083	0,01570
propinsi	0,60206	0	0,60206	0,01083	0,01570
ramal	0,60206	0	0,60206	0,01083	0,01570
republik	0,60206	0	0,60206	0,01083	0,01570
ribet	0,60206	0	0,60206	0,01083	0,01570
ringkih	0,60206	0	0,60206	0,01083	0,01570
sangkut	0,60206	0,60206	0	0,01735	0,00980
saran	0,60206	0,60206	0	0,01735	0,00980
sebar	0,60206	0,60206	0	0,01735	0,00980
sebut	0,30103	0,60206	0	0,01735	0,00980
separatis	0,60206	0	0,60206	0,01083	0,01570
serikat	0,60206	0	1,80618	0,01083	0,02751

term	idf	$\Sigma W_{kata\ t}$		LS	
		pendidikan	politik	pendidikan	politik
a	g	l	m	n	o
silep	0,60206	0,60206	0	0,01735	0,00980
sleman	0,60206	2,40824	0	0,03692	0,00980
sosialis	0,60206	0	0,60206	0,01083	0,01570
soviet	0,60206	0	1,20412	0,01083	0,02160
statistik	0,60206	0,60206	0	0,01735	0,00980
sumbangan	0,60206	0,60206	0	0,01735	0,00980
tanggal	0,60206	0,60206	0	0,01735	0,00980
tengah	0,60206	0,60206	0	0,01735	0,00980
tenggara	0,60206	0	0,60206	0,01083	0,01570
tingkat	0,60206	1,20412	0	0,02388	0,00980
tni	0,60206	0	1,80618	0,01083	0,02751
tujuane	0,60206	0,60206	0	0,01735	0,00980
tumbal	0,60206	0	0,60206	0,01083	0,01570
tutut	0,60206	0	0,60206	0,01083	0,01570
uni	0,60206	0	3,61236	0,01083	0,04521
urus	0,60206	0,60206	0	0,01735	0,00980
ussr	0,60206	0	0,60206	0,01083	0,01570
wiwit	0,60206	0,60206	0	0,01735	0,00980
wujud	0,60206	0,60206	0	0,01735	0,00980
yugoslavia	0,60206	0	0,60206	0,01083	0,01570

#### D. Proses Matching

Dari proses matching didapat beberapa kata yang sama, diantaranya :

Tabel hasil matching

term	tf testing
barang	1
dhidhik	1
mutu	1
tingkat	1

#### E. Memangkatkan Laplace Smoothing dengan tf-testing

term	tf testing	LS		LS^tf testing	
		pendidikan	politik	pendidikan	politik
barang	1	0,01083	0,01570	0,01083	0,01570
dhidhik	1	0,02714	0,00980	0,02714	0,00980
mutu	1	0,01735	0,00980	0,01735	0,00980
tingkat	1	0,02388	0,00980	0,02388	0,00980

Hasil perkalian setiap term pada masing – masing kelas :

$$\text{Kelas pendidikan} = 0,01083 \times 0,02714 \times 0,01735 \times 0,02388$$

$$= 1,218\text{E-07}$$

$$\text{Kelas politik} = 0,01570 \times 0,00980 \times 0,00980 \times 0,00980$$

$$= 1,479\text{E-08}$$

F. Menghitung *prior probabilities* masing – masing kelas

$$\text{Kelas pendidikan} = 2/4 = 0,5$$

$$\text{Kelas politik} = 2/4 = 0,5$$

G. Menghitung hasil perkalian di H dengan F, sehingga:

$$\text{Kelas Pendidikan} = 1,218\text{E-07} \times 0,5$$

$$= 6,090\text{E-08}$$

$$\text{Kelas Politik} = 1,479\text{E-08} \times 0,5$$

$$= 7,394\text{E-09}$$

Didapatkan  $6,090\text{E-08}$  sebagai nilai maksimal.

Dengan demikian, kelas testing termasuk dalam kategori pendidikan.

## LAMPIRAN II

Hasil *running* program

```
:Output - KlasifikasiDokumen (run)
run:
-> k_pendidikan1.txt
-> k_pendidikan2.txt
-> k_politik1.txt
-> k_politik2.txt
jumlah idf : 57.74660664503701

Menghitung jumlah W per kelas

Jumlah W kelas Ekonomi : 0.0

Jumlah W kelas Politik : 44.274043054318774

Jumlah W kelas Pendidikan : 34.56729697891046

Jumlah W kelas Kesehatan : 0.0
Training File Selesai
```

Gambar 1. Hasil *running* jumlah *idf* dan jumlah W per kelas

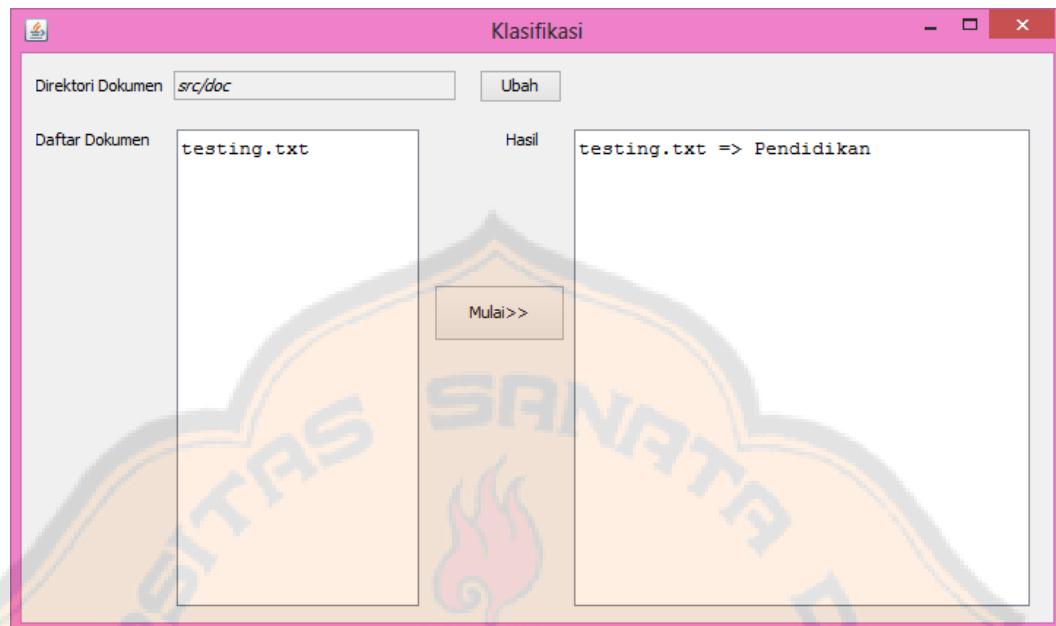
```
:Output - KlasifikasiDokumen (run)
run:
#####
matching :

barang
dhidhik
mutu
tingkat
jumlah kataUnik matching : 4

hasil perkalian (LS^tf testing)
8.992783216376228E-8
1.47885786402952E-8
1.2180897076333648E-7
8.992783216376228E-8

jumlah dokumen = 4.0
8.992783216376228E-8 dengan jumlah dokumen Ekonomi =0.0
Probabilitas 0.0
1.47885786402952E-8 dengan jumlah dokumen Politik =2.0
Probabilitas 7.3342893201476E-9
1.2180897076333648E-7 dengan jumlah dokumen Pendidikan =2
Probabilitas 6.090448538166824E-8
8.992783216376228E-8 dengan jumlah dokumen Kesehatan =0.0
Probabilitas 0.0
nilai yang paling tinggi = 6.090448538166824E-8
dan masuk ke dalam kategori = Pendidikan
```

Gambar 2. Hasil *running* matching dan hasil klasifikasi



Gambar 3. Hasil *running klasifikasi*