# DETEKSI KONTEN HOAX BERBAHASA INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN METODE LEVENSHTEIN DISTANCE

# **SKRIPSI**



# **DISUSUN OLEH:**

# FRISTA GIFTI WEDDININGRUM H76214015

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERISUNAN AMPEL

SURABAYA

2018

## LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama

: Frista Gifti Weddiningrum

NIM

: H76214015

Program Studi : Sistem Informasi

Angkatan

: 2014

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul: DETEKSI KONTEN HOAX BERBAHASA INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN METODE LEVENSHTEIN DISTANCE. Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya akan menerima sanksi yang telah ditetapkan. Demikian pernyataan keaslian ini saya buat dengan sebenarbenarnya.

Surabaya, 03 Agustus 2018

Yang membuat pernyataan,

Frista Gifti Weddiningrum

NIM: H76214015

# LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh :

NAMA : FRISTA GIFTI WEDDININGRUM

NIM : H76214015

Dosen Pembimbing I

Anang Kunaeff, M. Kom

NIP. 197911132014031001

JUDUL : DETEKSI KONTEN HOAX BERBAHASA INDONESIA PADA

MEDIA SOSIAL MENGGUNAKAN METODE LEVENSHTEIN

DISTANCE

Ini telah diperiksa dan disetujui untuk dujikan.

Surabaya, 23 Juli 2018

Dosen Pembimbing II

Noor Wahyudi, M.Kom NIP. 198403232014031002

ii

# LEMBAR PENGESAHAN

Skripsi oleh Frista Gifti Weddiningrum ini telah dipertahankan Di depan tim Penguji Skripsi Surabaya, 25 Juli 2018

> Mengesahkan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Ampel Surabaya

# Susunan Dewan Penguji

Penguji I

Anang Kunaefi, M. Kom

NIP. 197911132014031001

Penguji II

Noor Wahyudi, M.Kom

NIP. 198403232014031002

Penguji III

Mujib Ridwan M. T

NIP. 198604272014031004

/ /

Penguji IV

Khalid, M. Kom

NIP. 197906092014031002

Mengesahkan

Dekan Fakultas Sains dan Teknologi

RIAND Sunan Ampel Surabaya

Eni Purwati, M.Ag

196512211990022001



# KEMENTERIAN AGAMA UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA PERPUSTAKAAN

Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300 E-Mail: perpus@uinsby.ac.id

# LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama	: FRISTA GIFTI WEDDININGRUM					
NIM : H76214015						
Fakultas/Jurusan	: SAINTEK/SISTEM INFORMASI					
E-mail address : sk16tita@gmail.com						
UIN Sunan Ampe ■ Skripsi □ yang berjudul :	gan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan   Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :   Tesis					
MENGGUNAK	AN METODE LEVENSHTEIN DISTANCE					
Perpustakaan UIN mengelolanya da menampilkan/mer akademis tanpa p penulis/pencipta d Saya bersedia unt	yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Non-					
Demikian pernyata	an ini yang saya buat dengan sebenamya.					
	Surabaya, 8 Agustus 2018.					
	(FRISTA GIFTI WEDDININGRUM)					

#### **ABSTRAK**

# DETEKSI KONTEN HOAX BERBAHASA INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN METODE LEVENSHTEIN DISTANCE

## Oleh:

# Frista Gifti Weddiningrum

Media sosial merupakan wadah yang sangat mendukung untuk mempererat komunikasi antara sesama manusia. Namun tidak seluruh informasi yang disebarkan melalui media sosial berupa fakta. Telah terjadi berbagai macam kasus penyebaran berita yang bukan merupakan fakta atau sering disebut hoax. Untuk pengembangan teknologi penangkal hoax ini pun terdapat beberapa yang bermunculan, namun sejauh ini teknologi penangkal hoax yang diterapkan dalam sistem pendeteksi hoax masih jarang ditemukan. Dalam penelitian ini digunakan perhitungan Tf-Idf untuk mengukur bobot suatu kata dalam dokumen hoax dan metodeLevenshtein Distance (LD) untuk mengukur jarak antar kata dalam dokumen. Penerapan Metode Levenshtein Distance dalam Sistem Deteksi Hoaxmemiliki beberapa tahap yang dimulai dengan tahap pra-pemrosesan kata, dilanjutkan dengan tahap perhitungan Tf-Idf dankemudian tahap perhitungan jarak minimum antar kata menggunakan metode Levenshtein Distance. Hasil batas 0,0014 pada skenario 2 yang memiliki data latih sebanyak 100 berita terindikasi hoax dan 40 berita sebagai data uji. Pada batas 0,0014 tersebut memiliki nilai Precision, Recall dan Accuracy yang konsisten.

**Kata Kunci:** *Hoax, Levenshtein Distance*, Tf-Idf, Pra-pemrosesan, Sistem Deteksi. Media Sosial

#### **ABSTRACT**

# DETECTION OF INDONESIAN LANGUAGE HOAX CONTENT ON SOCIAL MEDIA USING LEVENSHTEIN DISTANCE METHOD

By:

# Frista Gifti Weddiningrum

Social media is a very supportive container to strengthen communication between human beings. But not all information that is spread through social media is a fact. There have been a wide range of news spreading cases that are not a fact or often called hoax. For the development of hoax antidote technology has also some popping, but so far the hoax antidote technology applied in the Hoax Detector System is still rare. In this study will be used Tf-Idf calculations to measure the weight of a word in the hoax document and the Levenshtein Distance method to measure the distance between words in the document. The application of Levenshtein Distance method in the Hoax Detection System begins with a word pre-processing, then added with Tf-Idf calculation and calculate the minimum distance between words using the Levenshtein Distance method. Limit Results 1.4 In scenario 2 that has a train data as much as 100 news indicated hoax and 40 news as test data. At the limit of 1.4 has Precision, Recall and Accuracy value that is likely to consistent.

**Keywords:**Hoax, Levenshtein Distance, Tf-Idf, Pre-processing, Detection System, Social Media

# **DAFTAR ISI**

LEMB	AR PERNYATAAN KEASLIAN	i
LEMB	AR PERSETUJUAN PEMBIMBING	ii
LEMB	AR PENGESAHAN	iii
LEMB	AR PERNYATAAN PERSETUJUAN PUBLIKASI	iv
ABSTE	RAK	v
DAFT	AR ISI	vii
DAFTA	AR GAMBAR	ix
DAFT	AR TABEL	X
BAB I	PENDAHULUAN	1
1.1	Latar Belakang	1
1.2	Perumusan Masalah	3
1.3	Batasan Masalah	3
1.4	Tujuan Penelitian	3
1.5	Manfaat Penelitian	
1.6	Sistematika Penulisan Skripsi	4
BAB II	I TINJAUAN PUSTAKA	5
2.1	Tinjauan Penelitian Terdahulu	5
2.2	Teori yang Digunakan	8
2.2	2.1 Natural Language Processing	8
2.2	2.2 TF-IDF	9
2.2	2.3 Levenshtein Distance	10
2.2	2.4 Media Sosial	12
2.2	2.5 Bahasa Pemrograman Python	
2.2	2.6 Pengukuran Performa	13
2.3	Integrasi Keilmuan	
BAB II	II METODOLOGI PENELITIAN	17
3.1	Identifikasi Masalah dan Studi Literatur	17
3.2	Pengumpulan Data	19
3.3	Pra-pemrosesan Data Target	22

3.4 Pembuatan Sistem Deteksi <i>Hoax</i>	27
3.5 Pengujian dan Analisis Hasil Sistem	29
3.6 Penulisan Laporan	30
BAB IV HASIL DAN PEMBAHASAN	32
4.1 Proses Pra-pemrosesan Data Target	32
4.1.1 Pra-pemrosesan Data Induk	32
4.1.2 Proses Perhitungan Komponen Tf-Idf	35
4.2 Proses Pembuatan Sistem Deteksi <i>Hoax</i>	40
4.2.1 Spesifikasi Sistem	40
4.2.2 Fungsi Input Berita dan Pra-pemrosesan Kata	40
4.2.3 Fungsi Perhitungan Jarak dan Penentuan Nilai Tf-Idf	41
4.2.4 Fungsi Perhitungan Akhir	46
4.2.5 Tampilan AntarmukaSistem	
4.3 Proses Pengujian Sistem	49
4.3.1 Pengujian Ske <mark>nar</mark> io 1	49
4.3.2 Pengujian Ske <mark>nar</mark> io <mark>2</mark>	55
4.4 Pembahasan	63
BAB V PENUTUP	66
5.1 Kesimpulan	
5.2 Saran	67
DAFTAR PUSTAKA	68

# **DAFTAR GAMBAR**

Gambar 3.1 Diagram Alir Tahapan Penelitian	17
Gambar 3.2Halaman Utama turnbackhoax.id	19
Gambar 3.3 Halaman Berita <i>Hoax</i> turnbackhoax.id	20
Gambar 3.4 Tampilan Isi Salah Satu Halaman Berita	20
Gambar 3.5 Tampilan Penyimpanan Seluruh Berita <i>Hoax</i>	21
Gambar 3.6 Tampilan Penyimpanan Data Uji Skenario 1	22
Gambar 3.7 Tampilan Data Uji Skenario 2	22
Gambar 3.8 Diagram Alir Pra-pemrosesan Teks	24
Gambar 3.9 Diagram Alir Perhitungan Komponen Tf-Ifd	27
Gambar 3.10 Diagam Alir Sist <mark>em</mark> Deteksi Ho <mark>ax</mark>	28
Gambar 4.1 Tampilan Dokum <mark>en</mark> Data Target	39
Gambar 4.2 Tampilan Antar <mark>mu</mark> ka Sis <mark>te</mark> m <mark>Det</mark> eksi <i>Hoax</i>	48
Gambar 4.3 Tampilan Peng <mark>ece</mark> ka <mark>n Berit</mark> a <i>Non-Ho<mark>ax</mark></i>	48
Gambar 4.4 Tampilan Penge <mark>cekan Berit</mark> a <i>Hoax</i>	49
Gambar 4.5 Tampilan Grafik 22 Data Uji	50
Gambar 4.6 Tampilan Grafik 40 Data Uji	56

# **DAFTAR TABEL**

Tabel 2.1 Penelitian Sebelumnya
Tabel 2.2 Precision and Recall
Tabel 3.1 Precision and Recall Berdasarkan Data Pengujian
Tabel 4.2 Hasil Pengujian 22 Berita
Tabel 4.3 Klasifikasi Komponen Perhitungan Precision, Recall & Accuracy
Skenario 1 Batas 0,0013
Tabel 4.4 Nilai dari Komponen Perhitungan Dengan Batas 0,0013 51
Tabel 4.5 Klasifikasi Komponen Perhitungan Precision, Recall & Accuracy
Skenario 1 Batas 0,0014
Tabel 4.6 Nilai dari Komponen <mark>P</mark> er <mark>h</mark> itungan <mark>Deng</mark> an Batas 0,0014 53
Tabel 4.7 Klasifikasi Kom <mark>pon</mark> en <mark>Per</mark> hit <mark>ung</mark> an <i>Precision, Recall &amp; Accuracy</i>
Skenario 1 Batas 0,001554
Tabel 4.8 Nilai dari Kompo <mark>nen</mark> P <mark>erhitungan</mark> Deng <mark>an</mark> Batas 0,0015 54
Tabel 4.9 Hasil Pengujian 4 <mark>0 Berita</mark> 55
Tabel 4.10 Klasifikasi Komponen Perhitungan Precision, Recall & Accuracy
Skenario 2 Batas 0,0013
Tabel 4.11 Nilai dari Komponen Perhitungan Dengan Batas 0,0013 58
Tabel 4.12 Klasifikasi Komponen Perhitungan Precision, Recall & Accuracy
Skenario 2 Batas 0,0014
Tabel 4.13 Nilai dari Komponen Perhitungan Dengan Batas 0,0014 60
Tabel 4.14 Klasifikasi Komponen Perhitungan Precision, Recall & Accuracy
Skenario 2 Batas 0,0015
Tabel 4.15 Nilai dari Komponen Perhitungan Dengan Batas 0,0015

#### **BABI**

## **PENDAHULUAN**

# 1.1 Latar Belakang

Media sosial merupakan wadah yang sangat mendukung untuk mempererat komunikasi antara sesama manusia. Jarak dan waktu tidak menjadi penghalang untuk saling berkomunikasi. Tidak hanya sebagai media komunikasi, media sosial juga berperan sebagai media penyebaran informasi. Informasi yang tersebar melalui media sosial akandengan cepat dikonsumsi oleh setiap akun yang dimiliki masyarakat.

Saling berbagi informasi dengan sesama merupakan hal yang positif, namun tidak seluruh informasi yang disebarkan melalui media sosial berupa fakta. Telah terjadi berbagaimacam kasus penyebaran berita yang bukan merupakan fakta atau sering disebut *hoax*. Sedangkan *hoax*adalah informasi berbahaya yang menyesatkan persepsi manusia dengan menyebarkan informasi yang salah namun dianggap sebagai kebenaran(Rasywir & Purwarianti, 2015). Informasi sesat dari *hoax* dapat menyebabkan kerusakan finansial dan menyakiti setiap pengguna individu dan lebih buruk dari itu *hoax* memiliki kemampuan untuk mengumpulkan informasi dan memungkinkan untuk meyakinkan penerima menghadiri acara-acara yang tidak pernah ada (Ishak, Chen, & Yong, 2012).

Beberapa informasi *hoax* disebabkan oleh perseorangan dan beberapa disebabkan oleh organisasi yang mengkhususkan dirinya dalam bidang pembuatan berita dan informasi *hoax* kemudian menyebarkannya pada masyarakat luas. Seperti yang telah dilansir oleh situs web CNN Indonesia bahwa data yang dipaparkan oleh Kementerian Komunikasi dan Informatika menyebut ada sebanyak 800 ribu situs di Indonesia yang terindikasi sebagai penyebar berita palsu dan ujaran kebencian (*hate speech*). Sedangkan Pemerintah Indonesia telah mengatur dalam Pasal 28 ayat 1 Undang-Undang No. 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik atau Undang-Undang ITE. Dalam pasal tersebut dituliskan bahwa "Setiap orang yang dengan sengaja dan atau tanpa hak

menyebarkan berita bohong dan menyesatkan, ancamannya bisa terkena pidana maksimal enam tahun dan denda maksimal Rp 1 miliar".

Di dalam Kitab Al Quran tertulis bahwa jika menerima suatu berita ataupun informasi Allah berfirman untuk lebih dahulu memeriksa dengan teliti sebelum kemudian dipahami dan disebarkan kepada orang lain agar tidak menimbulkan musibah dan membuat diri sendiri menyesal. Firman Allah tertulis di dalam Surat Al Hujurat ayat 6, seperti yang ditulis dan diartikan sebagai berikut ini:

Artinya: "Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu." (QS. Al Hujurat: 6).

Dampak yang dihasilkan oleh berita hoax merupakan dampak yang tidak bisa disadari secara langsung, karena berita hoax akan langsung menyerang pemikiran pembacanya. Berita hoax akan dicerna dan jika tidak berhati-hati akan mempengaruhi cara berpikir pembacanya. Istilah Hoax biasanya disebut sebagai "virus pikiran", hal ini dikarenakan kemampuannya untuk mereplikasi diri, mengadaptasi, memutasi dan bertahan di dalam pikiran manusia (Vuković, Pripužić, & Belani, 2009).

Semakin maraknya tren *hoax* yang meracuni pemberitaan terutama pada media sosial, maka bermunculan pula pemikiran-pemikiran untuk melakukan tindak pencegahan terhadap penyebaran berita *hoax*. Telah banyak bermunculan tips-tips untuk menghindari terjebak dalam berita *hoax*, dan banyak juga media sosial yang menyajikan layanan tambahan untuk mengadukan konten yang diduga mengandung unsur *hoax* dan SARA. Untuk pengembangan teknologi penangkal *hoax* ini pun terdapat beberapa yang bermunculan, namun sejauh ini teknologi penangkal *hoax* yaitu yang diterapkan dalam sistem pendeteksi *hoax* masih jarang ditemukan. Beberapa sistem menggunakan kecerdasan artifisial untuk

menentukan sebuah berita mengandung unsur *hoax* atau tidak, dan ada pula yang menggunakan algoritma-algoritma pembanding teks.

Dalam sistem pendeteksi *hoax* digunakan cara pengolahan yang di dalamnya juga memiliki beberapa tahapan untuk mengolah setiap kata, memisahkannya dan membandingkannya dengan kata-kata yang sudah ada sebelumnya. Dalam penelitian ini akan digunakan perhitunganTf-Idf untuk mengukur bobot suatu kata dalam dokumen *hoax* yang digunakan sebagai perbendaharaan dan metode*Levenshtein Distance* (LD) untuk mengukur banyaknya perbedaan yang dimiliki dalam setiap dokumen yang sedang diproses, sehingga pada hasil akhirnya akan didapati persentase dari kemungkinan berita tersebut mengandung *hoax*.

Berdasarkan latar belakang maraknya penyebaran berita *hoax* di atas, maka akan dibuat sebuah penelitian dengan judul "Deteksi Konten *Hoax* pada Media Sosial Menggunakan Metode TF IDF dan*LevenshteinDistance*".

#### 1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka dapat diambil rumusan masalah yaitu, Bagaimana cara untuk menerapkan metode*Levenshtein Distance* dalam Sistem Deteksi *Hoax*Berbahasa Indonesian?

#### 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini dapat disebutkan sebagai berikut:

- Data uji coba didapatkan melaluisitus web turnbackhoax.id yangtelah memvalidasi dan mengelompokkan konten *hoax* di media sosial sejak tahun 2015.
- 2. Penelitian ini menggunakan data uji berupa data percakapan media sosial dan berita berbahasa Indonesia.

## 1.4 Tujuan Penelitian

Menerapkan Metode*Levenshtein Distance* dalam sebuah sistem deteksi *hoax*berbahasa Indonesia.

#### 1.5 Manfaat Penelitian

- 1. Bagi praktisi, untuk mengetahui pemodelan dari sebuah sistem yang dapat mendeteksi nilai *hoax*sebuah berita melalui data-data yang diolah menggunakan Metode*Levenshtein Distance*.
- 2. Bagi akademisi, untuk memberikan referensi untuk pengembangan sistem yang sama ataupun sebagai referensi untuk kebutuhan bisnis maupun rancang bangun aplikasi yang memanfaatkan Metode*Levenshtein Distance*.

# 1.6 Sistematika Penulisan Skripsi

#### 1. BAB I PENDAHULUAN

Dalam bab ini membahas latar belakang penelitian, perumusan masalah penelitian, batasan masalah penelitian, tujuan penelitian, manfaat penelitian dan sistematika penulisan laporan penelitian.

#### 2. BAB II TINJAUAN PUSTAKA

Dalam bab ini membahas tentang peneltian yang pernah dilakukan sebelumnya dan teori-teori yang digunakan di dalam metodologi penelitian.

## 3. BAB III METODOLOGI PENELITIAN

Dalam bab ini membahas tentang metodologi penelitian yaitu tahap-tahapan yang dilakukan dalam penelitian dari tahap awal studi literatur hingga tahap akhir dokumentasi dan penulisan laporan penelitian.

## 4. BAB IV HASIL DAN PEMBAHASAN

Dalam bab ini memaparkan hasil-hasil dari seluruh tahapan penelitian, dari tahap analisis, desain, implementasi desain, hasil testing dan implementasinya.

#### 5. BAB V PENUTUP

Dalam bab ini berisi kesimpulan dan saran pengembangan sistem.

# BAB II TINJAUAN PUSTAKA

# 2.1 Tinjauan Penelitian Terdahulu

Melalui hasil dari studi literatur maka telah ditemukan beberapa rujukan utama dari penelitian ini. Dalam rujukan utama terdapat dua buah penelitian yang memiliki studi kasus yang sama yaitu pendeteksian konten *hoax* dalam *email*. Namun kedua penelitian tersebut memiliki perbedaan dari segi metode, penelitian yang pertama menggunakan metode *Neural Network* dan penelitian kedua menggunakan metode *Levenshtein Distance*. Untuk perbandingan lebih lengkap dari penelitian-penelitian sebelumnya dapat dilihat dalam Tabel 2.1 menjabarkanperbandingan penelitian sebelumnya yang menjadi dasar dari penelitian ini.

Tabel 2.1Penelitian Sebelumnya

Judul	Peneliti		Metode	Tahun	Data Set	Kelebihan (+) dan
						Kekurangan (-)
An Intelligent	Marin	-	Self-	2009	Email	(+) Sistem dapat
Automatic	Vuković,		Organizing		berbahasa	membedakan dan
Hoax	Krešimir		Map		Inggris	mengklasifikasi
	Pripužić, dan	7	Feed-			<i>email hoax</i> baru
Detection-	Hrvoje Belani		Forward			dengan
Sistem			Neural			membandingkan
			Network			polanya dengan pola
						tersimpan yang
						sama.
						(-) Jika terdapat
						email dengan pola
						yang baru, sistem
						belum dapat
						membedakannya
						karena pola tersebut
						tidak ada dalam

Judul	Peneliti	Metode	Tahun	Data Set	Kelebihan (+) dan
Judui	renenu	Metode	Tanun	Data Set	Kekurangan (-)
					Penyimpanan
					sehingga sistem akan
					menganggap email
					dengan pola baru
					sebagai <i>email</i> yang
					tidak mengandung
					hoax.
Distance-	Adzlan Ishak,	Levenshtein	2012	Email	(+) Dalam pengujian,
based Hoax	Y. Y. Chen,	Distance		berbahasa	sistem menghasilkan
Detection	dan Suet-			Inggris	nilai positive
Sistem	Peng Yong				predictive value
Sistem		4.0			sebesar 0,96.
		8 No. 1			(-) Sistem belum bisa
			74		mengidentifikasi
					email yang orisinil.
					Seluruh <i>email</i> akan
				,	diukur tingkat
				4	kandungan <i>hoax</i> nya.
Sistem Temu	Munjiah Nur	Tf-Idf, LCS	2013	Dokumen	(+) Memliki nilai
Kembali	Saadah,			Teks	recall sebesar
Dokumen	Rigga Widar				96,84% karena
Teks dengan	Atmagi, Dyah		//		sistem mampu
	S. Rahayu,				bekerja secara efektif
Pembobotan	Agus Zainal				dalam
Tf-Idf Dan	Arifin				mengembalikan
LCS					sejumlah dokumen.
					Dalam penelitian ini
					disebutkan bahwa
					metode bobot urutan
					kata dan Tf-Idf
					cukup signifikan jika
					dibandingkan dengan
					bobot yang diukur
					dengan metode Tasi,
					dkk.

Judul	Peneliti	Metode	Tahun	Data Set	Kelebihan (+) dan
Juun	1 chenti	Microde Tanun Data S		Data Sct	Kekurangan (-)
					(-) memiliki nilai
					presisi sebesar
					30,36%.
Eksperimen	Errissya	Naïve Bayes,	2015	Artikel	Dalam penelitian ini,
pada Sistem	Rasywir, Ayu	Support Vector		Berbahasa	peneliti melakukan
Klasifikasi	Purwarianti	Machine,		Indonesia	percobaan
Berita Hoax		Algoritma C4.5			menggunakan lebih dari satu metode
Berbahasa		C4.3			pada setiap proses
Indonesia					dalam sistem
Berbasis	/	(°			klasifikasi <i>hoax</i> .
					Hasil penelitian
Pembelajaran		4 N 2			tersebut berupa
Mesin					metode dan
					algoritma apa saja
					yang paling akurat
				,	untuk
				4	mengklasifikasikan
					berita <i>hoax</i> .
					Dalam proses
				4	prapemrosesan teks, peneliti menemukan
			/ ~		prapemrosesan tanpa
					stemming yang
					mencapai hasil
					terbaik.
					Dalam proses
					ekstraksi fitur, fitur
					unigram memiliki
					akurasi terbaik.
					Dan untuk algoritma
					klasifikasi, algoritma
					Naïve Bayes
					menunjukkan hasil
					akurasi terbaik yaitu

Judul	Peneliti	Metode	Tahun	Data Set	Kelebihan (+)	) dan
Judui	renenu	Metode	Tanun		Kekurangar	ı (-)
					mencapai	nilai
					90,00136%.	

# 2.2 Teori yang Digunakan

### 2.2.1 Natural Language Processing

Natural Language Processing (NLP) atau Pemrosesan Bahasa Alami adalah sebuah otomatisasi proses untuk mengkaji interaksi antara komputer dan bahasa alami manusia yang digunakan dalam kehidupan sehari-hari, karena bahasa alami manusia beraneka ragam sehingga dalam penerapan Natural Language Processing sering menemui permasalahan dalam ambiguitas kata ataupun kata dengan makna ganda (Priansya, 2017). NLP merupakan cabang ilmu kecerdasan buatan yang dikhususkan untuk mengolah linguistik.Bahasa manusia pemrosesan alami memiliki kebergaman dan aturan tata bahasa yang berbeda-beda, sehingga komputer perlu untuk memproses bahasa yang biasa digunakan sehari-hari oleh manusia sehingga dapat memahami maksud dari manusia pengguna sistem.

Dalam penerapannya, untuk membuat sebuah sistem yang dapat melakukan Pemrosesan Bahasa Alami terlebih dahulu melalui *Text Preprocessing* atau tahap sebelum memproses teks. *Text Preprocessing memiliki* beberapa tahapan yaitu (Katariya & Chaudhari, 2015):

#### 1. Analisis Leksikal Teks

Proses mengubah sebuah teks atau kalimat menjadi per kata, yang bertujuan untuk mengidentifikasi kata-kata dalam sebuah teks.

## 2. Penghapusan *Stopwords*

*Stopwords* merupakan kata umum yang sering digunakan dalam sebuah teks dan biasanya tidak berguna jika digunakan untuk tujuan pencarian, salah satu contohnya adalah kata penghubung, misalnya:

dan, atau, tetapi. Penghapusan *stopwords* memiliki manfaat penting yaitu untuk mengurangi ukuran indeks yang digunakan nantinya.

# 3. Stemming

Proses pemisahan kata yang mengandung awalan atau akhiran sehingga menghasilkan kata dasar. Hal ini berguna unutk meningkatkan kinerja pengambilan kata karena akan menguragi varian kata yang sama dalam konsep umum. Selain itu proses *stemming* juga berguna untuk mengurangi ukuran struktur pengindeksan karena jumlah istilah indeks yang berbeda menjadi berkurang.

#### 2.2.2 **TF-IDF**

Term Frequency Inverse Document Frequency atau biasa disebut dengan TF-IDF merupakan algoritma yang digunakan untuk mengukur bobot (weight) setiap kata pada sebuah dokumen atau bahkan sekumpulan dokumen-dokumen, bobot tersebut akan merepresentasikan pentingnya sebuah kata dalam dokumen, semakin besar nilai bobot maka semakin penting peran kata tersebut dalam membentuk sebuah dokumen.

Pendekatan TF-IDF menyajikan teks dengan ruang vektor yang disetiap fitur dalam teks sesuai dengan satu kata (Zhang, Gong, & Wang, 2005). TF (*Term Frequency*) akan menghitung frekuensi kemunculan sebuah kata dan dibandingkan jumlah seluruh kata yang ada di dalam dokumen, berikut persamaan yang digunakan untuk menghitung TF(Saadah, Atmagi, Rahayu, & Arifin, 2013).

$$tf(i) = \frac{freq(t_1)}{\sum freq(t)}$$
 (1)

Keterangan:

tf(i) : nilai *Term Frequency* sebuah kata dalam sebuah dokumen.

freq  $(t_i)$  : frekuensi kemunculan sebuah kata dalam sebuah dokumen.

 $\sum freq(t)$ : jumlah keseluruhan kata dalam dokumen.

Sementara IDF (*Inverse Document Frequency*) menghitung logaritma dari jumlah seluruh dokumen dan dibandingkan dengan jumlah dokumen dimana dalam dokumen tersebut kata (*t*) yang dimaksud muncul. Berikut persamaan yang digunakan untuk menghitung IDF(Saadah et al., 2013).

$$idf(i) = \log \frac{|D|}{|\{d: t_i \in d\}|}$$
 (2)

Keterangan:

idf(i) : nilai *Inverse Document Frequency* sebuah kata di seluruh isi dokumen.

: jumlah seluruh dokumen.

 $|(d: t_i \in d)|$  : jumlah dokumen yang mengandung kata (t).

Dengan kedua persamaan tersebut maka dapat ditentukan nilai bobot (w) sebuah kata dalam sekumpulan dokumen, dengan menghitung perkalian dari kedua persamaan sebelumnya. Berikut persamaan untuk menentukan nilai bobot (w) sebuah kata(Saadah et al., 2013).

$$weigth (tf-idf)_i = tf(i) \times idf(i)$$
(3)

#### 2.2.3 Levenshtein Distance

Levenshtein Distance adalah sebuah matriks untuk mengukur angka perbedaan antara 2 string, jarak antara string diukur berdasarkan angka penambahan karakter, penghapusan karakter ataupun penggantian karakter yang diperlukan untuk mengubah string sumber menjadi string target (Ishak dkk., 2012). Berikut matriks dari Levenshtein Distance(Afriansyah& Puspitaningrum, 2015):

$$leva, b(i,j) = \begin{cases} \max(i,j) & if \min(i,j) = 0 \\ leva, b(i-1,j) + 1 \\ leva, b(i,j-1) + 1 \\ leva, b(i-1,j-1) + 1(ai \neq bj) \end{cases}$$
(4)

Keterangan:

lev a,b adalah matriks levensteindistance;

i adalah baris matriks;

j adalah kolom matriks.

Dalam metode ini memiliki aturan penilaian yang akan dijelaskan dalam contoh sebagai berikut:

- Jika *string* sumber (a) adalah "hitung" dan *string* target (b) juga terisi dengan kata "hitung", maka nilai lev a,b = 0. Sehingga dalam proses tersebut tidak terjadi perubahan apapun dalam dua kata yang diukur jaraknya, karena kedua kata tersebut sama satu sama lain.
- Jika *string* sumber (a) adalah "hitung" dan *string* target (b) adalah "hutang", maka nilai lev a,b = 2, karena dalam prosesnya terjadi dua penggantian karakter huruf yaitu dari "i" menjadi "u" dan dari "u" menjadi "a". Proses penggantian tersebut dibutuhkan untuk mengubah *string* yang asli menjadi *string* gabungan.
- Kedua hasil di atas ditemukan melalui perhitungan di dalam matriks dari setiap karakter *string* yang dibandingkan menggunakan tiga persamaan di dalam nilai minimal.

Selanjutnya setelah didapatkan hasil dari matriks levenshtein di atas, maka dilanjutkan dengan perhitungan seberapa besar nilai kesamaan antara string yang dibandingkan menggunakan rumus berikut(Afriansyah & Puspitaningrum, 2015):

$$Similarity = \left\{1 - \frac{editdistance}{maxLengt\ h(stra, strb)}\right\}$$
 (5)

# Keterangan:

edit distance adalah hasil dari perbandingan yang telah dilakukan atau Levenstein Distance.

*maxLength* adalah jumlah string dari kata yang terpanjang antara str*a*dan str*b*.

stra adalah panjang string pertama.

strb adalah panjang string kedua

Similarity adalah nilai kesamaan antara kedua string.

Sehingga dapat ditarik kesimpulan bahwa semakin besar nilai Similarity yang dihasilkan maka semakin besar kesamaan yang dimiliki oleh dua dokumen yang dibandingkan.

#### 2.2.4 Media Sosial

Media sosial merupakan sebuah wadah dalam dunia maya yang digunakan untuk saling berkomunikasi ataupun saling berbagi informasi dengan sesama penggunanya.Sementara Jejaring sosial adalah struktur sosial yang terdiri dari beragam individu ataupun kelompok organisasi yang dihubungkan karena memiliki kesamaan sosialitas, visi, ide dan lain sebagainya (Priansya, 2017).

Setiap jejaring sosial memiliki kekhususan masing-masing dalam hal fitur untuk menghubungkan anggota-anggotanya. Terdapat beberapa yang menggunakan fitur bergambar sebagai fitur utama dan ada pula yang menggunakan fitur ruang percakapan (*chatting*) sebagai fitur utama, dan ada pula yang menyajikan lini masa untuk membantu menyebarkan informasi ataupun berita kepada pengguna jejaring sosial yang lain.

Masyarakat Anti Fitnah Indonesia (MAFINDO) adalah organisasi perkumpulan resmi yang didirikan pada tanggal 19 November 2016, Akta Notaris Nomor 1 Tanggal 19 November 2016 yang dibuat oleh ISMA

JANUARTI, SH., M.KN,SK (Surat Keputusan) Pendirian Perkumpulan NOMOR AHU-0078919.AH.01.07.TAHUN 2016 Menteri Hukum dan Hak Asasi Manusia.

Organisasi ini mengumpulkan konten berita melalui forum yang mereka miliki di jejaring sosial Facebook yaitu Forum Anti Fitnah Hasut dan *Hoax* (FAFHH). Konten berita tersebut akan diarsip dan didiskusikan di dalam forum, kemudian hasil diskusi tersebut akan dibagikan melalui situs resmi yang dimiliki MAFINDO yaitu turnbackhoax.id.

## 2.2.5 Bahasa Pemrograman Python

Bahasa Pemrograman Python termasuk ke dalam katehori *highlevel language* atau bahasa pemrograman yang mendekati bahasa manusia.Berbeda dengan *lowlevel language*, *highlevel language* tidak dapat dijalankan secaralangsung oleh mesin sehingga perlu diproses terlebih dahulu agar dapat dijalankan (Downey, 2012).

Karena Bahasa Python merupakan jenis highlevel language maka terdapat beberapa keuntungan di dalamnya yaitu dalam penulisan program yang tidak akan memakan banyak waktu, mudah untuk dibaca dan lebih mudah untuk dibenarkan. Kemudian Bahasa Python ini juga dapat dijalankan diberbagai macam komputer.

# 2.2.6 Pengukuran Performa

Ketika sebuah sistem telah berhasil dirancang sebagaimana mestinya dan sudah diimplementasikan yang kemudian menghasilkan nilai seperti yang diinginkan, maka tahapan selanjutnya adalah pengukuran performa. Pengukuran performa dilakukan untuk menguji keakuratan, keefektifan dan efisiensi sistem yang dibangun. Terdapat sekumpulan rumus yang dapat digunakan sebagai media pengukuran yang sesuai dengan penelitian yang sedang dilakukan yaitu *Precision, Recall* dan *Accuracy*.

Precision and Recall adalah matriks perhitungan yang digunkan untuk mengukur kefektifitasan pengambilan informasi(Manning, Raghavan, & Schütze, 2008).

• *Precision* (P) adalah pecahan dari dokumen dan diambil yang relevan.

$$Precision = \frac{\#(dokumen\ boax\ terklasifikasi\ boax)}{\#(jumla\ b\ dokumen\ terklasifikasi\ boax)}$$

• Recall (R) adalah bagian dari dokumen yang relevan yang diambil..

Gagasan tersebut dapat diperjelas melalui Tabel 2.2 berikut:

Tabel 2.2*Precision and Recall* (Sumber: Manning, Raghavan, & Schütze, 2008)

	Relevan	Tidak Relevan
Diambil	true positi <mark>ve (</mark> tp)	false positive (fp)
Tidak diambil	fa <mark>lse ne</mark> gative (fn)	true negative (tn)

Berdasarkan Tabel 2.2, dapat dituliskan rumus sebagai berikut untuk menghitung akurasi sebuah sistem menggunakan perhitungan *Precision and Recall*(Manning et al., 2008):

$$P = tp / (tp + fp) \tag{6}$$

$$R = tp / (tp + fn) \tag{7}$$

Dimisalkan jika terdapat 10 buah dokumen berita yang akan diuji dalam sistem deteksi *hoax* dan telah diklasifikasi sebelumnya menjadi 5 dokumen memiliki konten *hoax* dan 5 dokumen merupakan berita orisinil. Kemudian 10 dokumen tersebut diuji dalam sistem dan sistem memberikan hasil bahwa terdapat 7 dokumen yang terdeteksi sebagai konten *hoax* yaitu 4 berita berkonten *hoax* dan 3 berita berkonten orisinil. Maka dapat disebutkan bahwa 4 berita berkonten *hoax* yang diambil merupakan nilai *true positive* (tp), 3 berita berkonten orisinil yang diambil merupakan nilai *false positive* (fp), 1 berita berkonten hoax yang tidak diambil merupakan

nilai *false negative* (fn) dan 3 berita berkonten orisinil sisanya yang tidak diambil merupakan nilai *true negative* (tn).

Selain *Precision and Recall*, dalam perhitungan performa sistem juga diperlukan adanya perhitungan Akurasi sistem, untuk memastikan seberapa akurat sistem tersebut dapat digunakan dalam mendeteksi konten *hoax* pada berita. Tingkat akurasi sebuah sistem dapat dihitung menggunakan persamaan berikut (Syafitri, 2010).

$$ac = \frac{\sum matc \, h}{\sum tp} \times 100\% \tag{8}$$

Keterangan:

ac : tingkat akurasi (%)

 $\sum match$ : jumlahdeteksi yang benar

 $\sum tp$  : jumlah data yang diuji

Jumlah deteksi benar adalah jumlah banyaknya data uji yang telah diuji dan sesuai dengan pengelompokannya, nilai tersebut didapatkan dari penjumlahan antara nilai *true positive* dan nilai *true negative*. Kemudian pembaginya adalah total dari seluruh data yang digunakan dalam pengujian.

# 2.3 Integrasi Keilmuan

Seperti yang telah dijelaskan pada latar belakang di Bab I tentang *Hoax* yang merupakan suatu informasi bukan fakta namun disebarluaskan dan mempengaruhi pembacanya sehingga dipercaya sebagai sebuah fakta. Dalam hal ini khasus *hoax* merupakan sebuah penipuan publik, karena penyebarannya tidak hanya pada satu dua orang saja, dengan semakin majunya teknologi penyebaran informasi palsu ini dapat dengan mudah tersebar pada setiap orang disetiap kalangan.

Menurut Dr. H. Suis, M.Fil.I. yang merupakan seorang Kepala Program Studi Jurusan Dhirosah Islamiyah Strata-2 yang juga seorang pakar keislaman mengatakan bahwa berdasarkan hadis Rasulullah SAW, kebohongan dapat dikelompokkan menjadi tiga berdasarkan akibat yang disebabkan yaitu:

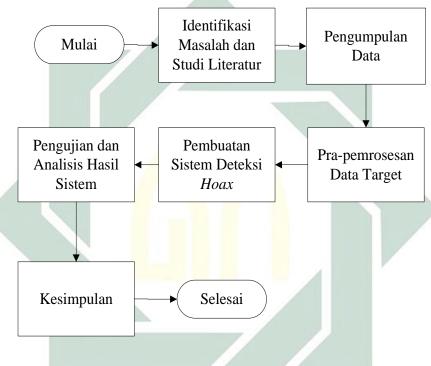
- Kebohongan yang tidak dipermasalahkan, merupakan kebohongan yang tidak membawa dosa lain setelah kebohongan tersebut. Dosanya sama dengan dosa perbuatan maksiat.
- 2. Kebohongan yang membawa dosa lain namun tidak sampai menyebabkan kehancuran.
- 3. Kebohongan yang membawa kehancuran, merupakan kebohongan yang jika dilakukan akan membawa dosa yang tidak terputus meskipun yang berbuat kebohongan sudah meninggal dunia.

Berita *hoax* merupakan berita bohong yang dapat menyebabkan penerimanya mempercayai isi dari berita atau informasi tersebut, dan kemudian menyebarluaskan sehingga terjadi kebohongan-kebohongan lain dan pada akhirnya mengakibatkan fitnah. Dalam firman Allah yang tertuang dalam surat Al-Hujurat ayat 6 disampaikan bahwa jika kita menerima sebuah berita hendaknya dipastikan dulu keasliannya sebelum kita menyebarkannya pada orang-orang yang belum mengetahuinya.

Berdasarkan hal-hal tentang berita dan kebohongan yang disebutkan di atas, maka dapat ditarik sebuah kesimpulan hubungan antara islam dengan penelitian yang dilakukan adalah hasil dari penelitian ini akan mempermudah setiap orang yang menggunakannya untuk memastikan seberapa dekat berita yang mereka terima mengandung unsur kebohongan/hoax, sehingga dapat mencegah perbuatan bohong *jariyah* (kebohongan yang tidak terputus) lainnya. Karena sebuah kebohohongan kecil dapat menjadi besar jika kebohongan tersebut dipercaya dan menyebabkan kehancuran jika kebohongan tersebut dilaksanakan.

# BAB III METODOLOGI PENELITIAN

Pada Gambar 3.1 merupakan bentuk diagram alir dari tahapan penelitian yang sedang dilaksanakan.



Gambar 3.1Diagram Alir Tahapan Penelitian

# 3.1 Identifikasi Masalah dan Studi Literatur

Tahap ini merupakan tahapan pertama dari serangkaian tahapan penelitian yang dilakukan. Tahap identifikasi masalah adalah tahapan dimana peneliti melakukan pendalaman data lebih lanjut mengenai studi kasus yang ada. Literatur-literatur yang digunakan sebagian besar berbahasa inggris dan berasal dari luar Indonesia, hal ini dikarenakan literatur yang membahas tentang studi kasus dan metode penyelesaian yang sama masih terlalu minim jika dalam Bahasa Indonesia. Penelitian ini dilakukan dengan menggunakan beberapa penelitian yang dijadikan rujukan, sehingga dalam pelaksanaannya akan sesuai dengan

metodologi yang pernah dilakukan dengan beberapa tambahan yang diperoleh dari penggabungan metode dari penelitian-penelitian tersebut.

Penelitian akan diterapkan dalam sebuah sistem yang disebut dengan sistem deteksi *hoax*. Di dalam sistem tersebut telah diterapkan metodologi pengolahan teks yang diawali dari pra-pemrosesan teks , dimana teks dokumen murni yang dimasukkan akan diolah dan disaring sehingga menghasilkan kata-kata yang lebih efektif untuk diolah pada proses selanjutnya. Pra-pemrosesan teks ini juga diterapkan dalam dua penelitian yang dijadikan rujukan yaitu penelitian dengan judul *Distance-based Hoax Detection Sistem*(Ishak et al., 2012) dan *An Intelligent Automatic Hoax Detection-Sistem*(Vuković et al., 2009).

Dalam proses pengolahan kata, ketika kata-kata *hoax* telah disaring memasuki tahap pengukuran jarak dan pengukuran bobot, hal ini dilakukan bertujuan untuk mmeningkatkan keakuratan sistem dalam membedakan konten *hoax* dan *non-hoax*. Proses pengukuran bobot kata (*weigth*) menggunakan algoritma Tf-Idf yang dirujuk dari penelitian dengan judul Sistem Temu Kembali Dokumen Teks dengan Pembobotan Tf-Idf dan LCS(Saadah et al., 2013), penelitian tersebut menyatakan bahwa dalam penggunaan perhitungan Tf-Idf akan menghasilkan nilai yang cukup signifikan dengan metode perhitungan bobot yang lain. Selain itu penelitian tersebut juga menghasilkan nilai *recall* yang besar.

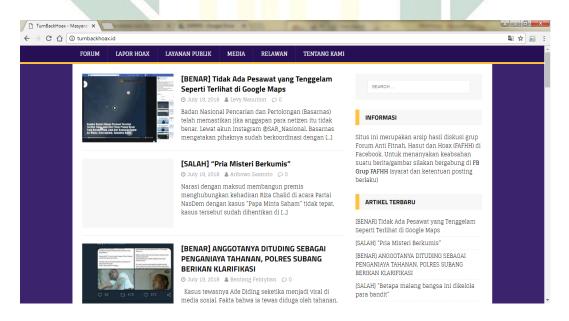
Selanjutnya adalah proses perhitungan jarak antar kata. Dalam penelitian ini telah ditentukan untuk menggunakan perhitungan jarak antar kata dibandingkan dengan algoritma artificial intelegent dikarenakan berdasarkan penelitian yang dirujuk dengan judul Distance-based Hoax Detection Sistem yang di dalamnya menggunakan metode perhitungan jarak antar kata, telah menghasilkan nilai positive predictive value sebesar 0,96 dibandingkan dengan penelitian yang juga dijadikan rujukan dengan judul An Intelligent Automatic Hoax Detection-Sistem yang di dalamnya menggunakan artificial intelegent untuk menemukan konten hoax, namun menghasilkan nilai 73,86% kebenaran dokumen hoax terklasifikasi. Sehingga dalam penelitian ini menggunakan metode Levenshtein Distance yaitu

metode perhitungan jarak antar kata yang digunakan oleh penelitian sebelumnya memiliki nilai *positive predictive value* 0,96.

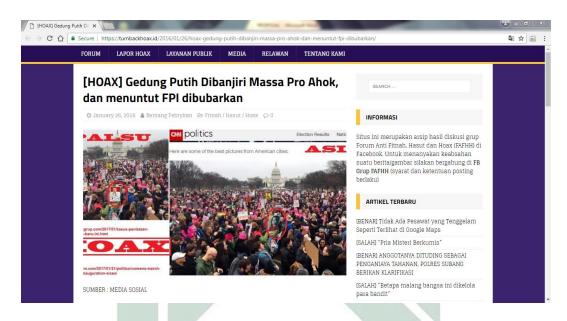
# 3.2 Pengumpulan Data

Dalam penelitian ini, data yang digunakan adalah data berita *hoax*, baik itu dari sosial media ataupun dari situs *web* yang khusus menyajikan berita untuk khalayak umum.turnbackhoax.id adalah sebuah situs *web* yang menyediakan datadata tersebut, terlebih situs ini mengkhususkan diri untuk menyediakan berita *hoax* dan *non-hoax* yang sudah diklasifikasikan berdasarkan hasil diskusi dan penelusuran fakta yang dilakukan oleh anggotanya maupun informasi-informasi yang didapatkan dari *non-*anggota yang membagikan fakta atau hanya sekedar untuk mengklarifikasi sebuah berita.

Berikut adalah Gambar 3.2 yang menyajikan tampilan dari halaman awal situs web turnbackhoax.id dan Gambar 3.3 yang menyajikan tampilan salah satu halaman berita terklasifikasi hoax.

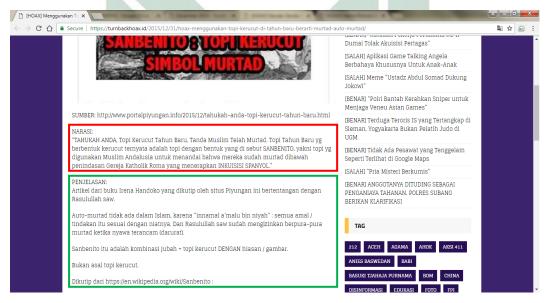


Gambar 3.2Halaman Utama turnbackhoax.id (Sumber: turnbackhoax.id)



Gambar 3.3Halaman Berita *Hoax* turnbackhoax.id (Sumber: turnbackhoax.id)

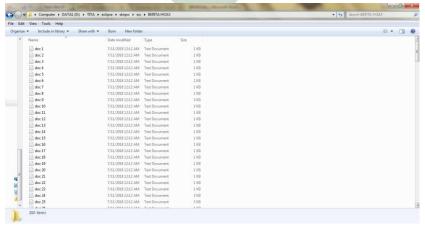
Bagian yang diambil dan disimpan dari halaman berita *hoax* adalah bagian yang mengandung narasi penyebaran berita *hoax*. Situs web turnbackhoax.id telah membagi bagian narasi berita *hoax* dan bagian penjelasan yang menjelaskna berita tersebut *hoax* di dalam satu halaman berita, berikut Gambar 3.4 menampilkan isi dari salah satu halaman berita situs turnbackhoax.id.



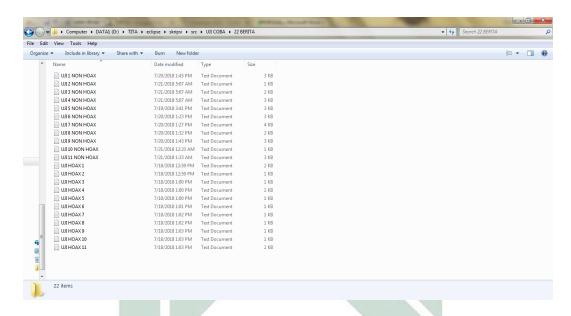
Gambar 3.4 Tampilan Isi Salah Satu Halaman Berita (Sumber: turnbackhoax.id)

Dapat dilihat pada Gambar 3.4 yang menampilkan isi halaman berita *hoax*, dalam gambar terdapat bagian narasi yang ditandai dengan persegi panjang berwarna merah, bagian tersebut merupakan narasi asli dari berita *hoax* yang tersebar sebelumnya dan bagian itu lah yang diambil dan disimpan dalam dokumen berita *hoax*. Sementara untuk bagian di dalam persegi panjang berwarna hijau merupakan bagian penjelasan justifikasi *hoax* yang merupakan hasil diskusi dan penelusuran informasi yang dilakukan oleh anggota Masyarakat Anti Fitnah Indonesia.

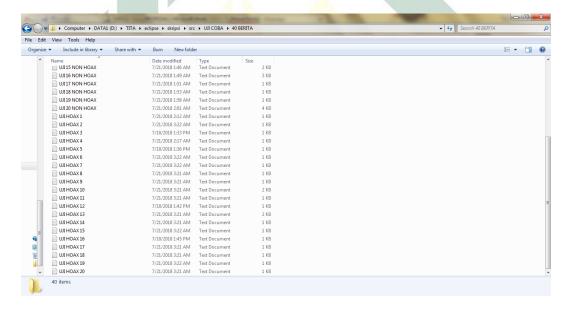
Penelitian ini menggunakan data berita yang sudah diklarifikasi jenisnya antara hoax dan non-hoax dari situs web turnbackhoax.id, dengan pembagian 70% untuk data latih dan 30% untuk data uji untuk kedua skenario. Sehingga untuk skenario pertama dapat dihasilkan 50 berita sebagai data latih dan 22 berita sebagai data uji and skenario kedua menghasilkan 100 berita sebagai data latih dan 40 berita sebagai data uji. Seluruh data latih dari kedua skenario adalah dokumen berita yang sudah diklasifikasikan sebagai berita hoax. Untuk data uji skenario pertama dibagi menjadi 2 yaitu 11 berita dengan konten hoax dan 11 berita dengan konten non-hoax dan untuk skenario kedua dibagi menjadi 20 berita dengan konten hoax dan 20 berita dengan konten non-hoax. Berikut tampilan penyimpanan seluruh dokumen berita dalam Gambar 3.5, tampilan penyimpanan data uji skenario 1 pada Gambar 3.6 dan tampilan data uji skenario 2 pada Gambar 3.7.



Gambar 3.5 Tampilan Penyimpanan Seluruh Berita *Hoax* 



Gambar 3.6 Tampilan Penyimpanan Data Uji Skenario 1



Gambar 3.7 Tampilan Data Uji Skenario 2

## 3.3 Pra-pemrosesan Data Target

Data yang sudah didapatkan melalui proses penelusuran dalam situs web turnbackhoax.id akan diolah dan disimpan ke dalam penyimpanan dokumen dengan format .txt untuk dijadikan data yang nantinya akan diproses kembali dalam perhitungan Tf-Idf. Penggunaan Tf-Idf selain untuk memberikan bobot setiap kata dalam dokumen juga digunakan sebagai penyeleksi fitur atau kata yang memegang informasi yang sangat mempengaruhi sebuah dokumen.Penyeleksian

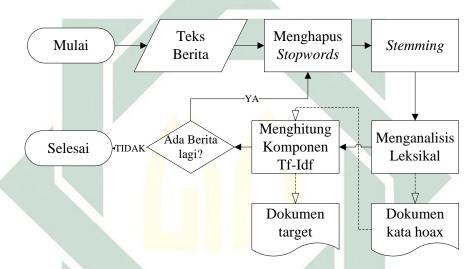
menggunakan metode Tf-Idf didapatkan dari kata yang memiliki nilai Tf-Idf atau nilai bobot yang besar dibandingkan dengan bobot kata lain. Semakin besar nilai bobot Tf-Idf maka semakin penting kata tersebut dalam sebuah dokumen.

Selain Tf-Idf, terdapat berbagai macam metode untuk melakukan penyeleksian kata, seperti penelitian yang dijadikan rujukan oleh penelitian ini yang berjudul Eksperimen pada Sistem Klasifikasi Berita Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin(Rasywir & Purwarianti, 2015), dalam penelitian tersebut membandingkan beberapa metode untuk menemukanyang tepat untuk mendeteksi berita *hoax*. Salah satu yang dibandingkan adalah metode untuk menyeleksi kata. Metode penyeleksian kata ini adalah metode tambahan untuk meningkatkan keakuratan pendeteksian *hoax*. Tidak hanya Tf-Idf, berdasarkan penelitian sebelumnya, metode untuk penyeleksian kata ini ada berbagai macam yaitu *Information Gain* (IG), *Mutual Information* (MI), *Chi-Square* (CS).

Pra-pemrosesan data seacara umum memiliki tiga proses utama yaitu Analisis Leksikal, Penghapusan *Stopwords*, dan yang terakhir adalah *Stemming*. Dalam penelitian ini memiliki kerangka yang berbeda dengan pra-pemrosesan data secara umum. Pra-pemrosesan data dalam penelitian ini tetap memiliki langkah yang sama namun dengan urutan yang berbeda, yaitu diawali dengan seleksi kata yang terlebih dahulu memasuki tahap Penghapusan *Stopwords*, setelah diolah di dalam *Stopwords*, seleksi kata selanjutnya dilakukan oleh *Stemming* dan berakhir di tahap Analisis Leksikal. Hal tersebut dilakukan karena bahasa pemrograman Python memiliki *Library Stopwords* dan *Stemming* yang dapat menyeleksi kata dari seluruh isi berita tanpa melalui tahap Analisis Leksikal terlebih dahulu. Tahapan Pra-pemrosesan ini akan menghasilkan kata-kata berita*hoax* terseleksi dan disimpan dalam dokumen dengan format .txt.

Selain mengubah urutan seleksi kata, pra-pemrosesan data pada penelitian ini juga ditambahkan tahap perhitungan komponen pembentuk rumus Tf-Idf. Hal tersebut dilakukan karena data yang dihasilkan oleh tahap ini nantinya akan

disimpan ke dalam dokumen dengan format .txt bersama dengan kata-kata dari berita *hoax* yang sudah melalui tahap pra-pemrosesan utama, penggabungan data kata *hoax* dan data komponen Tf-Idf dilakukan untuk mengurangi kesalahan pencocokan nilai Tf-Idf dari masing-masing kata. Dokumen tersebut nantinya akan digunakan sebagai datatarget dalam matriks perhitungan *Levenshtein Distance*, dengan hanya memilih dan menyeleksi data kata *hoax* saja.Tahap-tahap tersebut di atas dapat dijabarkan dalam Diagram Alir di Gambar 3.8 di bawah ini.



Gambar 3.8Diagram Alir Pra-pemrosesan Teks

# a. Penghapusan Stopwords

Dalam tahap ini akan dilakukan penghapusan kata-kata yang umum digunakan dan biasanya tidak memiliki pengaruh jika digunakan dalam proses pencarian, seperti contohnya adalah kata penghubung (dan, tetapi, karena, atau, dan lain sebagainya) ataupun kata awalan(di, ke, dari, dan lain sebagainya). Berikut contoh penerapannya.

Kalimat utuh:

0 1 2 3 4 5

Kalimat → Politisi tersebut dikabarkan telah terjerat kasus

6

korupsi

Hasil ketika sudah melalui proses penghapusan stopwords:

0 1 2 3

Kalimat → Politisi dikabarkan terjerat kasus korupsi

Tahap ini diperlukan selain karena kata-kata tersebut tidak memiliki pengaruh yang besar pada proses utama, penghapusan *stopwords* juga berguna untuk meminimalisir ukuran *String* yang digunakan ketika program dijalankan.

# b. Stemming

Dalam proses *stemming* ini sebuah kata akan diproses kembali untuk menghapus awalan dan akhiran (*suffix*, *prefix*) pada sebuah kata sehingga kata tersebut menjadi kata dasar. Seperti yang dicontohkan di bawah ini.

Kata yang sudah melalui proses penghapusan stopwords:

0 1 2 3 4

Kalimat → Politisi dikabarkan terjerat kasus korupsi

Kata yang telah melalui proses *stemming*:

0 1 2 3 4

Kalimat → Politisi kabar jerat kasus korupsi

Dapat dilihat perubahannya adalah dalam indeks 1 mengalami penghapusan awalan dan akhiran dari "dikabarkan" menjadi "kabar". Kemudian dalam indeks 2 mengalami penghapusan awalan dari "terjerat" menjadi "jerat".Hal ini diperlukan untuk mengurangi ukuran dari *String*dan untuk menjadikan pencarian lebih akurat.

#### c. Analisis Leksikal Teks

Proses analisis leksikal memisahkan kata per kata dari hasil proses Stemming dan ditempatkan kedalam indeks-indeks. Hasil dari proses analisis leksikal juga dapat disebut sebagai hasil akhir dari proses prapemrosesan kata utama dan disimpan perbaris di dalam dokumen dengan format .txt sebagai *Library* Kata *Hoax*. Proses pemisahan kata dalam analisis leksikal dapat dilihat pada contoh berikut:

0 1 2 3 4

Kalimat hasil proses Stemming → Politisi kabar jerat kasus korupsi

Teks yang sudah diolah dalam proses analisis leksikal:

0 1 2 3 4

[indeks] → [Politisi] [kabar] [jerat] [kasus] [korupsi]

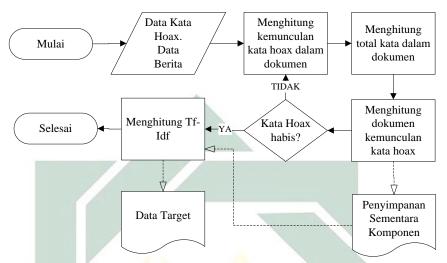
## d. Perhitungan Komponen Tf-Idf

Proses Perhitungan Komponen Tf-Idf merupakan proses dimana sistem akan menghasilkan angka-angka yang membentuk perhitungan Tf-Idf berdasarkan kata yang tersimpan dalam dokumen kata *hoax* dan dibandingkan dengan seluruh dokumen berita. Seperti yang sudah dijelaskan dalam teori Tf-Idf di BAB II, Tf-Idf merupakan penggabungan dari dua rumus melalui perkalian yaitu rumus Tf dan rumus Idf yang menghasilkan bobot sebuah kata dalam dokumen.

Dalam proses ini, sistemakan menghitung banyaknya kata yang muncul dalam dokumen  $(t_i)$  dan total seluruh kata dalam dokumen  $(\sum t)$  yang merupakan komponen pembentuk rumus Tf. Proses ini juga menghitung banyaknya dokumen dengan kata yang dicari nilai bobotnya (d:  $t_i \in d$ ) yang merupakan komponen perhitungan Idf.

Setelah mendapatkan komponen-komponen tersebut sistem menyimpannyake dalam dokumen perhitungan sementara Tf-Idf dengan format .csv. Dokumen tersebut nantinya akan diolah lagi secara manual menggunakan Aplikasi Microsoft Excel sehingga menghasilkan nilai Tf-Idf dan disimpan bersama dengan kata yang ada di dalam dokumen kata *hoax*. Kata-kata *hoax* disimpan bersama hasil perhitungan Tf-Idf dalam dokumen dengan format .txt yang nantinya akan digunakan sebagai data target di

dalam system deteksi *hoax*. Untuk lebih jelasnya proses perhitungan komponen Tf-Idf dapat di lihat dalam Diagram Alir Gambar 3.9 berikut:



Gambar 3.9Diagram Alir Perhitungan Komponen Tf-Ifd

#### 3.4 Pembuatan Sistem Deteksi *Hoax*

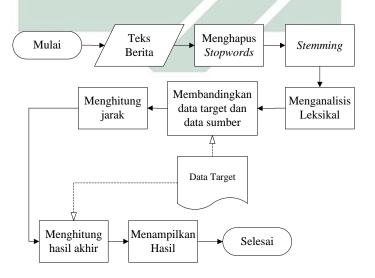
Dalam tahap ini akan dibuat sebuah sistem deteksi *hoax*. Setelah melalui tahap pra-pemrosesan data target yang telah dijabarkan pada sub bab sebelumnya, maka tahap selanjutnya adalah membuat sebuah proses untuk melakukan pengecekan atau pendeteksian konten berita yang akan diuji dan dibandingkan dengan kata-kata yang sudah disimpan melalui tahap pra-pemrosesan data target. Dalam alur sistem juga dilakukan pra-pemrosesan teks untuk teks berita baru yang dimasukkan, namun untuk penyimpanan data teks yang dimasukkan ini akan disimpan ke dalam penyimpanan yang berbeda dengan data target. Data baru ini akan disimpan ke dalam *list*Python, dikarenakan data masukan ini tidak digunakan dalam jangka waktu yang lama seperti data target, sehingga digunakan penyimpanan sementara yaitu di dalam *list*Python

Sistem ini menggunakan hasil perhitungan Tf-Idf untuk menghitung bobot kata danperhitungan Levenshtein Distance untuk menghitung jarak antar kata yang dibandingkan. Hasil perhitungan Tf-Idf didapatkan melalui data target. Untuk hasil perhitungan jarak kata didapatkan dengan membandingan dua kata yaitu kata pertama di dapatkan melalui berita yang dimasukan dalam sistem dan selanjutnya

disebut sebagai kata sumber, dan kata kedua didapatkan dari data target yang selanjutnya disebut kata target.

Perbandingan dua kata tersebut menggunakan perhitungan *Levenshtein Distance*, dengan menghitung banyaknya usaha yang dilakukan untuk mengubah sebuah kata sumber menjadi kata target, usaha yang dimaksud berbentuk penghapusan huruf, penambahan huruf, dan subtitusi huruf. Jika dimisalkan dalam kata sumber terdapat 2 huruf yang dihapus dan 1 huruf diubah menjadi huruf lain sehingga menjadi sama dengan kata target, maka didapatkan nilai perhitungan *Levenshtein Distance*-nya adalah 3, karena terdapat 3 usaha untuk mengubah kata sumber menjadi kata target. Ketika jarak kedua kata sudah ditemukan, maka sistem akan menghitung nilai kesamaannya menggunakan rumus *similarity*. Nilai kesamaan (similarity) kata tersebut kemudian akan dikalikan dengan nilai bobot (Tf-Idf) dan didapatkan hasil akhir dari sebuah kata sumber yang diproses tersebut. Hasil paling akhir dari sistem ini merupakan hasil dari perhitungan ratarata seluruh hasil akhir setiap kata sumber yang dibandingkan.

Sistem dibangun menggunakan bahasa pemrograman Python dan berbasis *desktop*. Alur dari sistem pengecekan akan digambarkan dalam diagram alir pada Gambar 3.10 berikut.



Gambar 3.10 Diagam Alir Sistem Deteksi Hoax

#### 3.5 Pengujian dan Analisis Hasil Sistem

Setelah sistem sudah terbentuk, maka selanjutnya dilakukan pengujian pada sistem. Pengujian dilakukan dengan dua skenario, skenario pertama menggunakan data berita yang sudah diambil sebanyak 22berita yang terdiri dari 11 berita hoaxdengan rentang April 2016 –Mei 2016 dan 11 non-hoaxdengan rentang waktu Oktober 2015 – Deseember 2016 berita yang dijadikan data uji merupakan berita yang sudah diketahui pengklasifikasiannya. Skenario kedua menggunakan data berita yang sudah diambil sebanyak 40 berita yang terdiri dari 20 berita hoaxdengan rentang Oktober 2016 – November 2016 dan 20 non-hoax dengan rentang April 2016 – Januari 2017. Kemudian teks berita tersebut akan diolah di dalam sistem dan menghasilkan nilai akhir pendeteksian. Semakin besar nilai yang dihasilkanmaka semakin besar kemungkinan teks berita tersebut mengandung unsur hoax. Sebaliknya, semakin kecil nilai yang dihasilkan maka semakin kecil kemungkinan teks berita tersebut mengandung unsur hoax.

Ketika proses pengujian selesai, selanjutnya adalah masuk ke tahap analisis sistem. Sebelum itu, seluruh nilai yang dihsailkan oleh proses pengujian akan diubah dalam bentuk tabel untuk mengetahui batas-batas yang dapat digunakan sebagai penentu klasifikasi *hoax* sebuah berita.

Dalam tahap analisis, sistem akan dinilai menggunakan pengukuran keakuratan sistem menggunakan *Precision and Recall* dan Pengukuran Akurasi. Dalam sistem akan dilihat ketepatan dan akurasinya melalui perhitungan tersebut.Berdasarkan rumus yang telah ditetapkan dalam BAB II, maka rumus dapat lebih diperjelas dengan memasukkan unsur perhitungan berdasarkan datadata yang dihasilkan dalam pengujian.Berikut penjelasan lebih lanjut jika dalam rumus dimasukkan unsur yang perlu dihitung berdasarkan data pengujian.

*Precision* (P) mengukur ketepatan sistem melakukan klasifikasi jenis dokumen *hoax* dan *non-hoax*.

$$Precision = \frac{\#(dokumen\ hoax\ terklasifikasi\ hoax)}{\#(jumla\ h\ dokumen\ terklasifikasi\ hoax)}$$

• *Recall* (R) mengukur ketepatan sistem menghasilkan nilai-nilai yang relevan sehingga dokumen dapat terklasifikasi.

$$Recall = \frac{\#(dokumen\ hoax\ terklasifikasi\ hoax)}{\#(jumla\ h\ dokumen\ hoax\ yang\ diuji)}$$

Gagasan tersebut dapat diperjelas melalui Tabel 3.1 berikut:

Tabel 3.1 Precision and Recall Berdasarkan Data Pengujian

Aktual Prediksi	Dokumen <i>Hoax</i>	Dokumen Non-Hoax
Terklasifikasi  Hoax	true positive (tp)	false positive (fp)
Terklasifikasi	false negative (fn)	true negative (tn)
Non-Hoax	AL AL	

Setelah ditemukan banyak jumlah masing-masing komponen dari Tabel 3.1 dari hasil data yang diuji, maka selanjutnya adalah menghitung *Precision, Recall* dan *Accuracy*-nya dalam rumus berikut.

$$Precision = tp / (tp + fp)$$
 (6)

$$Recall = tp / (tp + fn)$$
 (7)

Kemudian berikut adalah rumus akurasinya:

$$ac = \frac{\sum matc \ h \ (tp+tn)}{\sum tp} \times 100\%(8)$$

Keterangan:

ac : tingkat akurasi (%)

∑match : jumlahdokumen hoax terklasifikasi hoax dan non-hoax

terklasifikasi non-hoax

 $\sum tp$  : jumlah data yang diuji

## 3.6 Penulisan Laporan

Pada tahap terakhir ini akan dilakukan penulisan langkah kerja penelitian dari awal proses hingga akhir proses pengujian dan analisis sistem. Penulisan keseluruhan laporan ini mengikuti format yang sudah ada dan menampilkan

proses-proses penelitian yang benar-benar dilakukan oleh peneliti. Laporan penelitian ini diharapkan untuk dapat menjadi acuan untuk pengembangan lebih lanjut dari sistem yang sama maupun sebagai referensi untuk pembuatan sistem yang berbeda.



#### **BAB IV**

### HASIL DAN PEMBAHASAN

Berdasarkan penelitian yang telah dilakukan selama rentang waktu yang telah ditentukan, maka berikut adalah penjelasan dari proses penelitian dan hasil dari proses tersebut.

## 4.1 Proses Pra-pemrosesan Data Target

Seperti yang sudah dijelaskan di dalam BAB III, pra-pemrosesan data target akan melalui 2 tahap. Tahap pertama merupakan pra-pemrosesan data induk yang terdiri dari Penghapusan *Stopwords*, *Stemmming* dan Analisis Leksikal, dan tahap kedua merupakan tahap Perhitungan Komponen Tf-Idf. Kedua tahapan tersebut dijelaskan dalam sub-sub bab berikut ini.

## 4.1.1 Pra-pemrosesan Data Induk

Pra-pemrosesan data induk dilakukan untuk menyeleksi data berita yang akan dijadikan sebagai data latih. Karena dalam penelitian ini menggunakan dua skenario percobaan data, maka berita yang disaring juga memiliki dua kelompok berita. Dengan skenario 1, jumlah berita yang disaring sebanyak 50 berita dan dengan skenario 2, jumlah berita yang disaring sebanyak 100 berita. Kedua skenario tersebut akan melalui proses yang sama dalam Pra-pemrosesan Data Induk yaitu Penghapusan Stopwords, Stemming dan Analisis Leksikal. Berikut merupakan code bahasa pemrograman Python berbasis desktop dari Pra-pemrosesan Data Induk.

```
Deklarasi:
i = integer
numb = integer
maxDocNumb = integer
j = String
openDoc = String
stopwords = String
stemming = String
readDoc = String
lexicalAnalysis[] = String
```

```
Deskripsi:
for i <maxDocNumb
numb← i + 1
openDoc← (Directory: doc(numb))
readDoc← openDoc
for j ←readDoc[]
stopwords← j
stemming← stopwords
lexicalAnalysis[] ← stemming
```

Dalam code di atas dapat dilihat terdapat iterasifor i < maxdocNumb, iterasi tersebut akan terus berulang berdasarkan banyaknya nilai yang diberikan di dalam maxdocNumb Fungsi dari iterasi tersebut adalah untuk mengulang suluruh proses pra-pemrosesan data induk berdasarkan banyaknya berita yang akan diproses. Jika akan menyimpan berita hoax yang akan diproses, maka nilai di dalam maxdocNumb menjadi nilai maksimal dari banyaknya dokumen yang akan diproses.

Kemudian di dalam iterasi, terdapat codeopenDoc ← (Directory: doc(numb)) yang berfungsi untuk membuka file dalam direktori yang menyimpan berita hoax. Fungsi dari numb dalam doc(numb) adalah numb akan berulang dan membuka dokumen dengan contoh [doc 1, doc 2, doc 3,....., doc n] n sama dengan nilai maxDocNumbyang dimasukkan.Setelah membuka dokumen, maka codereadDoc ← openDocdigunakan untuk membacadan menampung dalam bentuk list/array per baris dari dokumen berita yang dibuka.

Seluruh isi dokumen yang sudah di tampung dalam *list*akan diiterasi berdasarkan panjang *list* yang menampung berita. Di dalam *code* iterasi for  $j \leftarrow readDoc$  terdapat proses pra-pemrosesan data dari baris kalimat yang sudah disimpan di dalam*list*. Jadi setiap kalimat yang terdapat di dalam *list*akan di saring dalam proses ini. Penyaringan pertama dengan Penghapusan *Stopwords*: stopwords  $\leftarrow j$ , kemudian hasil dari stopwordsdiporses dalam *Stemming*: stemming  $\leftarrow$  stopwords dan yang

terakhir memisahkan kalimat hasil dari proses *stemming* menjadi perkata dalam proses Analisis Leksikal:lexicalAnalysis[]  $\leftarrow$  stemming. Dalam proses Analisis Leksikal, kata *hoax* yang telah tersaring akan ditampung di dalam *list*.

Terdapat proses tambahan setelah dilakukan penyaringan kata dalam pra-pemrosesan kata, yaitu proses penyeleksian kesamaan kata. Jika didalam lexicalAnalysis[] terdapat kata yang sama maka kata yang sama tidak akan disimpan lagi. Proses ini dilakukan oleh iterasi bersyarat, berikut adalah *code* dari proses penyeleksian kesamaan kata dan penyimpanannya dalam dokumen yang akanmenyimpan data kata *hoax*.

```
Deklarasi:
k = integer
maxRangeArray = integer
temp = String
lexicalAnalysis[] = String
uniqueArray[] = String
openDoc = String
writeDoc = String
Deskripsi:
for k <maxRangeArray
str \leftarrow lexical Analysis [k]
for x ← str
if x !=uniqueArray
thenuniqueArray[] ← x
openDoc← (Directory: hoaxDoc.txt)
for m <maxRangeArray</pre>
writeDoc← uniqueArray[m]
openDoc← writeDoc
```

Dalam *code* di atas terdapat dua iterasi, iterasi yang pertama digunakan untuk menampung isi dari setiap indeks ke dalam temp. Kemudian iterasi kedua digunakan untuk menyeleksi kata yang sama dengan syarat yang disebutkan di dalam if, jika kata baru yang akan diseleksi tidak terdapat pada uniqueArray, maka kata tersebut akan tersimpandidalamnya.

Seperti dalam *code* sebelumnya, dimana terdapat fungsi untuk membuka file dan membacanya. Dalam *code*openDoc ← (Directory: hoaxDoc.txt) memiliki fungsi untuk membuka file dokumen di dalam direktori. Selanjutnya iterasi for m < maxRangeArrayakan melakukan perulangan sebanyak panjang dari indeks uniqueArray. Di dalam perulangan tersebut juga memiliki fungsi untuk menulis seluruh isi indeks uniqueArray ke dalam file dokumen yang telah dibuka sebelumnya. Dokumen yang menyimpan kata-kata yang telah disaring tersebut akan dijadikan sebagai penyimpanan kata *hoax* sementara dan disimpan di dalam hoaxDoc.txtyang kemudian akan diproses lebih lanjut dalam proses Perhitungan Komponen Tf-Idf.

## 4.1.2 Proses Perhitungan Komponen Tf-Idf

Perhitungaan komponen Tf-Idf dilakukan untuk mendapatkan nilai dari setiap kata yang telah tersimpan dalam dokumen kata *hoax* terhadap seluruh dokumen berita yang belum terseleksi. Dalam proses ini akan menghasilkan dokumen data target yang berisi seluruh kata *hoax* yang telah tersimpan sebelumnya dan hasil Tf-Idf dari setiap kata tersebut.

Dalam proses perhitungan komponen Tf-Idf akan memiliki beberapa iterasi yang didalamnya memiliki fungsi masing-masing. Berikut adalah gambaran umum dari keseluruhan proses yang dilakukan dalam iterasi utama.

```
Deklarasi:
  openDoc = String
  readDoc = String
  h,m = String
  lecxicalAnalysis[] = String

Deskripsi:
  openDoc 		(Directory: hoaxDoc.txt)
  readDoc 		(openDoc
  for h 		 readDoc
  lexicalAnalysis[] 		 h
  for m 		 lexicalAnalysis
  countWordTerm(m)
  countDocWords()
```

Dalam *code* tersebut terdapat dua fungsi untuk membuka dokumen yang berbeda. openDoc ← (Directory: hoaxDoc.txt) memiliki fungsi untuk membuka dokumen hoaxDoc.txtyang berisi data kata hoax hasil dari proses sebelumnya, kemudian *code*readDoc ← openDoc memiliki fungsi untuk membaca setiap baris dari isi dokumenhoaxDoc.txt dan menampungnya di dalam lexicalAnalysis[]yang sekaligus memecah urutan kata yang menjadi satu string menjadi per-String di dalam indeks lexicalAnalysis[].

Setelah seluruh kata berhasil ditampung di dalam array lexicalAnalysis[], maka proses selanjutnya adalah masuk ke dalam fungsi countWordTerm(). Dalam fungsi countWordTerm() akan dijalankan proses untuk menghitung komponen dari rumus Tf dan Idf, yaitu menghitung banyaknya kata yang muncul dalam sebuah dokumen (ti) untuk Tf dan menghitung banyaknya dokumen yang memiliki kata tersebut untuk Idf. Berikut adalah *code* dari proses tersebut.

```
Deklarasi:
i = integer
numb = integer
count, countDoc = integer
maxWordNumb = integer
m,j,k = String
openDocToRead, openDocToSave = String
readDoc, insertDoc = String
word = String
stopwords = String
stemming = String
lexicalAnalysis[] = String
Deskripsi:
countWordTerm(word)
openDocToSave← (Directory: doc.csv)
insertDoc→ openDocToSave
for m ←word
for i < maxWordNumb</pre>
numb \leftarrow i + 1
openDocToRead← (Directory: doc(numb))
readDoc← openDocToRead
for j ← readDoc
```

Di dalam*code*di atas terdapat iterasi for m ← word yang memiliki fungsi untuk mengulang proses dalam iterasi tersebut sebanyak jumlah kata *hoax*yang ditampung dalam maxWordNumb. Dalam iterasi ini juga terdapat proses yang sama dengan pra-pemrosesan sebelumnya, hal ini dilakukan untuk membandingkan dan menghitung keberadaan setiap kata *hoax* dalam setiap kata di dalam dokumen-dokumen berita. Dokumen berita tersimpan di dalam *list*lexicalAnalysis.

Kemudian kata *hoax* yang sudah diproses akan dibandingkan dalam iterasi for k ← lexicalAnalysisyang di dalamnya memiliki fungsi untuk penyeleksian oleh if m == k. Jika kata *hoax* (m) terdapat di dalam *list* berita (k) maka proses akan menambahkan 1 pada perhitungan kata (count += 1). Kemudian hasil perhitungan tersebut akan disimpan dalam dokumen csv yang telah dibuka di awal proses (openDocToSave ← (Directory: doc.csv)).

Perhitungan selanjutnya adalah perhitungan jumlah dokumen yang memiiki kata *hoax*(m). Jika kata *hoax* (m) terdapat di dalam *list* berita (k) maka nilai hitung akan lebih besar dari 0, sehingga proses akan

menambahkan 1 pada perhitungan dokumen (countDoc += 1)dalam perhitungan ini akan menghasilkan nilai n<sub>i</sub> yang merupakan komponen perhitungan Idf, jika perhitungan kata sama dengan nol, maka kata tersebut tidak terdapat di dalam dokumen, kemudian disimpan dalam dokumen csv yang sama dengan komponen Tf. Selanjutnya nilai hitung akan dikembalikan menjadi 0, untuk menghitung dari awal pada setiap iterasi kata yang dibandingkan.

Proses perhitungan selanjutnya adalah menghitung banyaknya kata yang terdapat dalam sebuah dokumen, dalam proses countDocWords() ini akan menghasilkan nilai  $\sum t$  yang merupakan komponen perhitungan Tf. Berikut adalah code dari perhitungan banyak kata dalam dokumen.

```
Deklarasi:
i = integer
numb, maxDocNumb = integer
amount = integer
lengthLexicalAnalysis = integer
j = String
openDocToRead, openDocToSave = String
readDoc, insertDoc = String
lexicalAnalysis[] = String
Deskripsi:
countDocWords()
for i < maxDocNumb</pre>
numb \leftarrow i + 1
openDocToRead← (Directory: doc(numb))
readDoc← openDocToRead
for j ←readDoc
lexicalAnalysis[] ← j
amount \leftarrow amount + lengthLexicalAnalysis
openDocToSave← (Directory: doc.csv)
insertDoc←amount
openDocToSave← insertDoc
amount← 0
```

Dalam *code* tersebut terdapat fungsi untuk membuka dokumen openDocToRead ← (Directory: doc(numb)), yang kemudian akan diolah di dalam iterasi for j ← readDoc yang memiliki fungsi untuk mengulang perhitungan di dalamnya. Proses di dalam iterasi tersebut

menghitung kata yang terdapat di dalam sebuah dokumen (amount ← amount + lengthLexicalAnalysis), kemudian hasil dari proses tersebut disimpan di dalam dokumen (insertDoc ← amount).

Setelah seluruh komponen perhitungan Tf-Idf terkumpul seluruhnya, maka tahap selanjutnya adalah perhitungan nilai Tf-Idf yang dilakukan menggunakan Microsoft Excel dengan data yang ada di dalam dokumen hitung tfidf.csv. Tujuan dari penggunaan dokumen dengan format csv adalah untuk memudahkan melakukan perhitungan Tf-Idf, bahasa pemrograman Python hanya mendukung pembacaan dan penulisan dalam bentuk *plain text*, csv merupakan salah satu dari jenis *plain text* yang dapat mengolah data angka jika dibuka melalui Microsoft Excel.

Hasil dari perhitungan Tf-Idf yang dilakukan di dalam *file* csv yang dibuka melalui Microsoft Excel tersebut adalah nilai rata-rata Tf-Idf setiap dokumen dari masing-masing kata *hoax*. Kemudian setiap kata *hoax* dan hasil Tf-Idf nya disimpan ke dalam dokumen yang digunakan sebagai data target di dalam Sistem Deteksi *Hoax*. Jumlah seluruh kata yang terdapat pada data target skenario 1 adalah sebanyak 1355 kata hoax dari 50 dokumen berita dan untuk skenario 2 sebanyak 2380 kata hoax dari 100 dokumen berita. Berikut Gambar 4.1 merupakan tampilan dari dokumen data target.

```
The Lat Trimut Your Hope Age of the Control of the
```

Gambar 4.1 Tampilan Dokumen Data Target

#### 4.2 Proses Pembuatan Sistem Deteksi *Hoax*

Sistem Deteksi *Hoax* merupakan program yang akan menjalankan pendeteksian terhadap berita yang dimasukkan oleh pengguna ke dalam sistem untuk dicek apakah berita tersebut termasuk berita *hoax* atau *non-hoax*.Di dalam program ini terdapat perhitungan jarak menggunakan *Levenshtein Distance* atau juga dikenal sebagai *Edit Distance*.

#### 4.2.1 Spesifikasi Sistem

Spesifikasi komputer yang digunakan untuk membuat sistem deteksi hoax adalah dengan RAM minimal 2GB, didampingi dengan prosesor minimal Intel Core i.3. Jika program dijalankan menggunakan komputer dengan RAM 2GB, maka penginstalan bahasa pemrograman Python yang bisa dilakukan adalah Python versi 2.7 ke bawah, untuk penggunaan Python versi 3 ke atas, tidak direkomendasikan bagi pengguna komputer dengan RAM 2GB.

Library-library Python yang digunnakan untuk membuat sistem deteksi hoax adalah LibraryNLTK, Library Sastrawi, dan Library Tkinter. LibraryNLTK digunakan untuk memanggil fungsi perhitungan edit distance, Library Sastrawi digunakan untuk memanggil fungsi Penghapusan Stopwords dan Stemming Berbahasa Indonesia, sementara Library Tkinter digunakan untuk membuat desain antarmuka pengguna.

Untuk penyimpanan dapat menggunakan basis data yang support jika digunakan dengan bahasa pemrograman Python, ataupun penyimpanan dapat dilakukan did lam dokumen, seperti yang sudah diperlihatkan dalam penelitian ini.

### 4.2.2 Fungsi Input Berita dan Pra-pemrosesan Kata

Tahap pertama yang dilakukan oleh sistem deteksi *hoax* adalah mendapatkan masukan berupa berita yang dilakukan oleh pengguna dan mengolahnya di dalam pra-pemrosesan kata.Berikut adalah code dari proses tersebut.

```
Deklarasi:
i = integer
lengthLexicalAnalysis = integer
news = String
stopwords = String
stemming = String
temp = String
lexicalAnalysis[] = String
uniqueArray[] = String
Deskripsi:
news←|user input|
stopwords← news
stemming← stopwords
lexicalAnalysis[] 
   stemming
for i <lengthLexicalAnalysis</pre>
temp←lexicalAnalysis[i]
for x ←temp
if x !=uniqueArray
uniqueArray← x
```

Ketika program di jalankan pengguna terlebih dahulu memasukkan berita yang akan dicek oleh sistem, jika pengguna sudah memasukkan berita ke dalam perintah tersebut maka *code*news  $\leftarrow$  |user input| akan mengambil nilai berita tersebut sebagai *String* yang disimpan dalam *String*news. Selanjutnya *String*news akan masuk ke dalam pra-pemrosesan kata dengan langkah yang sama seperti pra-pemrosesan sebelumnya, yaitu Penghapusan *Stopwords*: stopwords  $\leftarrow$  news, *Stemming*: stemming  $\leftarrow$  stopwords dan Analisis Leksikal: lexicalAnalysis[].

Kemudian hasil dari proses pra-pemrosesan kata akan disimpan di dalam *String* temp dan dilanjutkan dengan penyeleksian kata untuk mengurangi redundansi kata dan di simpan di dalam *array*uniqueArray.

## 4.2.3 Fungsi Perhitungan Jarak dan Penentuan Nilai Tf-Idf

Setelah semua proses pra-pemrosesan kata selesai sehingga berita yang dimasukkan oleh pengguna sudah melalui proses penyaringan dan menghasilkan kata-kata spesifik tanpa perulangan. Proses selanjutnya yang dilakukan oleh sistem adalah proses perhitungan jarak (*Levenshtein* 

*Distance*) dengan membandingkan kata masukan pengguna yang selanjutnya disebut sebagai kata sumber dengan kata target yang berasal dari data target, kemudian masuk ke dalam proses penentuan nilai Tf-Idf dari setiap kata sumber.

Sebelum masuk kedalam perhitungan utama, proses terlebih dahulu membuka file dokumen data target kemudian menemukan nilai jarak minimal dari perbandingan kata sumber dan kata target. Berikut *code*dari proses tersebut.

```
Deklarasi:
value = integer
b = float
minValue = float
openDocToRead = String
readDoc = String
k,a = String
uniqueArray [] = String
lexicalAnalysis[] = String
saveSourceWords [] = String
LD(source, target) = Levenstein Distance Matrix
Deskripsi:
openDocToRead← (Directory: hoaxDoc.txt)
readDoc← openDocToRead
minValue← 99
for i ← uniqueArray
 saveSourceWords[] ← i
for k ← readDoc
       lexicalAnalysis[] \leftarrow k
 a,b ←lexicalAnalysis[]
 value \leftarrow LD(i, a)
 ifvalue<minValue
       thenminValue←value
```

Iterasi for i ← uniqueArray merupakan iterasi utama yang digunakan dalam proses di atas adalah iterasi yang mengulang seluruh proses sebanyak kata sumber yang ditampung dalam listuniqueArray. Kemudian kata sumber disimpan di dalam listsaveSourceWords[] ← i. Proses selanjutnya ada di dalam iterasi pertama yang terdapat di dalam iterasi utama. Iterasi tersebut memiliki fungsi untuk mengulang seluruh

proses di dalamnya berdasarkan banyaknya kata yang terdapat di dalam readDoc.

Kemudian masuk ke dalam proses hitung jarak antara kata sumber (i) dan kata target (a). Di awali dengan mengambil karakter menggunakan lexicalAnalysis[] menjadi dua karakter. Karakter pertama disimpan dalam Stringa mengambil kata target dan karakter kedua disimpan dalam Stringb mengambil nilai Tf-Idf dari kata target (a).Kemudian menghitung jarak kata sumber dan kata target (value = LD(i, a)) dan selanjutnya sistem mencari nilai minimal dari hasil jarak kedua kata denganif value < minValue.Jika hasil perhitungan jarak lebih kecil dari nilai minimal yang sudah ditetapkan, maka nilai minimal akan memiliki nilai yang sama dengan hasil perhitungan jarak (minValue = value). Nilai yang dihasilkan dari proses di atas merupakan nilai paling minimal dari jarak kedua kata untuk selanjutnya diolah dalam proses perhitungan panjang String dari dua kata yang dibandingkan, berikut code dari proses tersebut.

```
Deklarasi:
value = integer
minValue = float
b = float
targetLength, sourceLength, maxLength = float
j,a = String
readDoc = String
lexicalAnalysis[] = String
LD(source, target) = Levenshtein Distance Matrix
Deskripsi:
for j ←readDoc
lexicalAnalysis[] ← j
a,b ←lexicalAnalysis[]
ifvalue == minValue
thentargetLength\leftarrowlength(a) - 1
      sourceLength ← length (i)
iftargetLength>maxLength
thenmaxLength targetLength
ifmaxLength<sourceLength
thenmaxLength ← sourceLength
```

Dalam proses tersebut terdapat iterasi for j ← readDoc yang berfungsi untuk mengulang seluruh proses di dalamnya sebanyak panjang StringreadDoc. Dalam iterasi tersebut memiliki proses awal yang sama dengan iterasi sebelumnya yaitu mencari nilai jarak dari dua kata yang dibandingkan, namun nilai jarak pada proses ini digunakan untuk di bandingkan dengan nilai minimal yang merupakan hasil dari proses sebelumnya. Di dalam if value == minValue menyatakan bahwa, jika hasil jarak dari kata target (a) sama dengan nilai minimal, maka proses akan mengambil kata target (a) untuk dihitung panjang karakternya (targetLength ← length(a) - 1)dan dibandingkan dengan panjang karakter dari kata sumber (sourceLength ← length(i)).

Masih di dalam perbandingan sebelumnya terdapat persyaratan if targetLength > maxLength, jika panjang karakter kata target lebih besar dari panjang maksimal yang sudah ditetapkan, maka nilai panjang maksimal sama dengan nilai panjang karakter target. Kemudian di dalam persyaratan tersebut juga terdapat persyaratan if maxLength < sourceLength, jika nilai panjang maksimal kurang dari panjang karakter kata sumber maka nilai panjang maksimal sama dengan panjang karakter kata sumber. Dari proses di atas akan di hasilkan panjang karakter paling besar dari dua kata yang dibandingkan dalam perhitungan jarak. Setelah mendapatkan nilai panjang karakter maksimal, selanjutnya proses akan menghitung nilai-nilai yang dibutuhkan untuk membentuk hasil akhir. Berikut code dari perhitungan-perhitungan yang saling berhubungan untuk menentukan hasil akhir dari sistem deteksi hoax.

```
Deklarasi:
b = float
similarity = float
minValue = float
maxLength = float
tfidfMix = float
count, countAverage = float
averageFirstValue = float
saveFirstValue[] = float
saveFirstAverage[] = float
```

```
i = String
saveSourceWords[] = String

Deskripsi:
similarity  1 - (minValue/maxLength)
tfidfMix  similarity * b
count  count +tfidfMix
saveFirstValue[]  count
if i !=saveSourceWords
thencount  0
averageFirstValue  count/lengthSaveFirstValue
saveFirstAverage[]  averageFirstValue

countAverage  countAverage + saveFirstAverage[lastIndex]
delete saveFirstValue[]
count  0
```

Code di atas diawali dengan perhitungan similarity dari kedua kata yang dibandingkan (similarity ← 1 - (minValue/maxLength)).Kemudian hasil similarity digabungkan dengan nilai Tf-Idf (b) dari kata target (a) dalam perkalian (tfidfMix ← similarity \* b). Hasil perkalian similarity dan Tf-Idf akan dijumlahkan dengan hasil perkalian yang sama namun dengan nilai similarity dan Tf-Idf kata target (a) yang berbeda. Kemudian kata target (a) disimpan di dalam listsaveFirstValue[], hal ini digunakan untuk menghitung banyaknya kata target yang memiliki nilai jarak minimal yang sama terhadap satu kata sumber (i).

Kemudian terdapat persyaratan if i != saveSourceWords digunakan untuk menyaring jika kata sumber (i) tidak terdapat di dalam data kata target maka proses akan otomatis memberikan nilai nol pada seluruh perhitungan, sehingga hasil akhir dari kata sumber (i) yang tidak ada dalam daftar kata target adalah nol.

Di luar dari persyaratan di atas, perhitungan selanjutnya adalah menghitung rata-rata dari keseluruhan hasil perkalian *similarity* dan Tf-Idf (averageFirstValue ← count/lengthSaveFirstValue) kata target (a) yang memiliki jarak minimal terhadap kata sumber (i). Hasil rata-rata

tersebut disimpan ke dalam *list*saveFirstAverage[]. Setelah mendapatkan rata-rata perhitungan, maka proses selanjutnya adalah menghitung rata-rata dari seluruh hasil akhir yang disimpan di dalam *list*simpan rerata.

Dalam *code* terdapat fungsi delete saveFirstValue[] digunakan untuk menghapus seluruh isi di dalam *list*saveFirstValue[] sehingga pada perulangan selanjutnya nilai yang disimpan akan kembali pada indeks nol. Begitu juga dengan variabel hitung yang dikembalikan menjadi nol, sehingga perhitungan yang disimpan dalam variabel hitung pada perulangan sekarang tidak mempengaruhi perhitungan pada perulangan selanjutnya.

### 4.2.4 Fungsi Perhitungan Akhir

Setelah seluruh perulangan utama atau perulangan seluruh kata sumber telah dilakukan maka, poroses selanjutnya adalah perhitungan paling akhir dari seluruh proses sistem deteksi *hoax*, berikut *souce code* perhitungannya.

```
Deklarasi:
lastValue = float
countAverage = float
limitValue = float
lengthSaveSourceWords = String

Deskripsi:
lastValue ← countAverage / lengthSaveSourceWords
if lastValue >= limitValue
then Output ← "Berita berkonten HOAX"
else if lastValue < limitValue
then Output ← "Berita merupakan FAKTA"
```

Rumus lastValue ← countAverage / lengthSaveSourceWords merupakan rumus untuk mengitung hasil paling akhir dari proses sistem deteksi *hoax*. Perhitungannya adalah dengan cara membagi hasil penjumlahan seluruh rata-rata hasil akhir setiap kata sumber (countAverage) dengan jumlah kata sumber yang di bandingkan dalam sistem (lengthSaveSourceWords). Perhitungan tersebut juga dapat disebut dengan rata-rata dari seluruh hasil akhir setiap kata sumber.

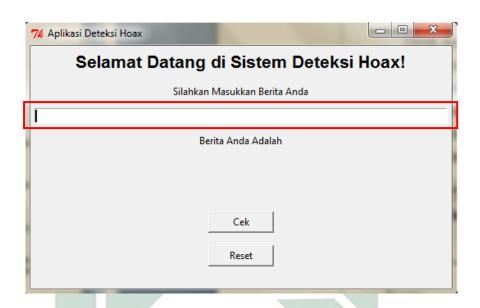
Berdasarkan hasil akhir dari proses pendeteksian, maka ditentukan batas pengklasifikasian (limitvalue) untuk menentukan berita yang dicek merupakan berita *hoax* atau *non-hoax*. Batas tersebut ditentukan berdasarkan hasil pengujian sistem dan disesuaikan dengan banyaknya data yang digunakan.Kemudian sistem mencetak hasilnya berdasarkan batas yang telah ditentukan. Jika hasil pendeteksian di atas atau sama dengan batas klasifikasi maka berita tersebut termasuk *hoax* dan jikadi bawah batas, maka berita terklasifikasi *non-hoax* atau Fakta.

## 4.2.5 Tampilan AntarmukaSistem

Tampilan Antarmuka Sistem Deteksi *Hoax* dirancang menggunakan Graphic User Interface (GUI) standar Python Tkinter.Python memiliki berbagai macam GUI yang dapat digunakan untuk membuat tampilan antarmuka sistem, terdapat beberapa yang berbayar dan ada pula yang tidak berbayar, salah satu GUI yang tidak berbayar dan mudah digunakan adalah Tkinter.

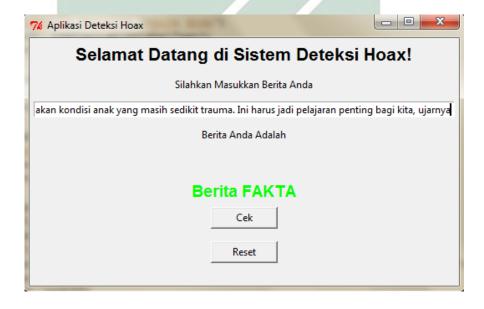
Tkinter merupakan GUI yang dapat langsung digunakan tanpa perlu melakukan peng-install-an terlebih dahulu, karena GUI Tkinter telah ter-install bersamaan ketika melakukan peng-install-an Python.Untuk menggunakan modul Tkinter, terlebih dahulu melakukan pemanggilan modul dengan menggunakan from Tkinter import \* di awal modul program.Dengan begitu seluruh modul GUI yang dimiliki Tkinter telah dipanggil dan dapat digunakan secara bebas dengan digabungkan dengan modul sistem.

Berikut pada Gambar 4.2 merupakan tampilan antarmuka sederhana dari Sistem Deteksi Hoax.

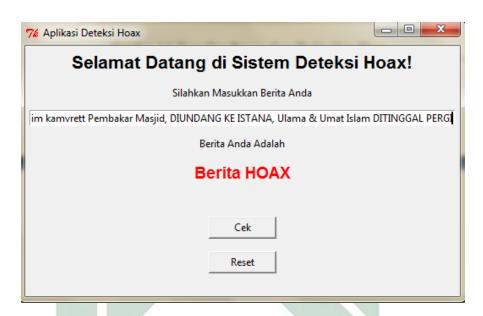


Gambar 4.2 Tampilan Antarmuka Sistem Deteksi Hoax

Untuk memasukkan berita jika hendak melakukan pengecekan, maka salin teks berita dari sumber dan temple pada teks area yang diberi tanda persegi panjang merah. Kemudian klik tombol 'Cek' untuk melakukan pengecekan dan mendapatkan hasil klasifikasi berita yang dimasukkan. Berikut tampilan antarmuka ketika sistem selesai melakukan pengecekan, Gambar 4.3 untuk pengecekan dengan hasil konten *non-hoax* dan Gambar 4.4 untuk pengecekan dengan hasil konten *hoax*.



Gambar 4.3 Tampilan Pengecekan Berita Non-Hoax



Gambar 4.4 Tampilan Pengecekan Berita *Hoax* 

Kemudian untuk menghapus berita yang telah dicek menggunakan tombol 'Reset', maka berita lama akan terhapus dan kondisi sistem kembali seperti semula.

#### 4.3 Proses Pengujian Sistem

Seperti yang dijelaskan dalam BAB III tentang pengujian sistem, yaitu tentang pengujian yang dilakukan dengan dua skenario, skenario pertama menggunakan data berita yang sudah diambil sebanyak 22 berita yang terdiri dari 11 berita *hoax*dengan rentang April 2016 – Mei 2016 dan 11 *non-hoax*dengan rentang waktu Oktober 2015 – Deseember 2016 berita yang dijadikan data uji merupakan berita yang sudah diketahui pengklasifikasiannya. Skenario kedua menggunakan data berita yang sudah diambil sebanyak 40 berita yang terdiri dari 20 berita *hoax* dengan rentang Oktober 2016 – November 2016 dan 20 *non-hoax* dengan rentang April 2016 – Februari 2017. Berikut hasil pengujian yang dijabarkan dalam sub-sub bab berikut.

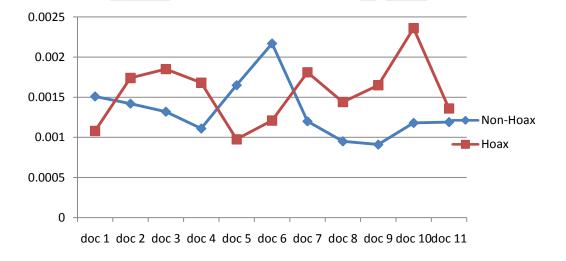
#### 4.3.1 Pengujian Skenario 1

Pengujian Skenario 1 menggunakan data uji sebanyak 22 berita, yang dibagi menjadi 11 berita *non-hoax* dan 11 berita *hoax*. Berikut adalah Tabel 4.1 yang menunjukkan data hasil pengujian dari 22 berita.

Tabel 4.2 Hasil Pengujian 22 Berita

Berita Non-Hoax	Hasil	Berita Hoax	Hasil
doc 1	0.00151	doc 1	0.00108
doc 2	0.00142	doc 2	0.00174
doc 3	0.00132	doc 3	0.00185
doc 4	0.00111	doc 4	0.00168
doc 5	0.00165	doc 5	0.000976
doc 6	0.00217	doc 6	0.00121
doc 7	0.0012	doc 7	0.00181
doc 8	0.00095	doc 8	0.00144
doc 9	0.00091	doc 9	0.00165
doc 10	0.00118	doc 10	0.00236
doc 11	0.00119	doc 11	0.00136

Berdasarkan Tabel di atas maka dapat dibuat diagram garis untuk mempermudah menentukan batas-batas untuk melakukan klsifikasi berita hoax dan non-hoax dalam sistem deteksi hoax. Berikut Gambar 4.5 merupakan tampilan grafik garis dari 22 data pengujian di atas.



Gambar 4.5 Tampilan Grafik 22 Data Uji

Berdasarkan grafik tersebut pada titik 0,0015 beberapa berita terklasifikasi sesuai jenisnya maka untuk selanjutnya dalam tahap Analisis

Hasil Sistem akan ditentukan tiga batas analisis digunakan untuk menentukan nilai *Precision, Recall & Accuracy*yaitu batas 0,0013; batas 0,0014; batas 0,0015 dari sistem deteksi *hoax*.

Berikut penjabaran dari klasifikasi dan nilai komponen perhitungan *Precision, Recall & Accuracy*dari 22 berita dari Tabel 4.2 yang menjabarkan klasifikasi dengan batas 0,0013, dan Tabel 4.3 merupakan nilai dari komponen perhitungan dengan batas 0,0013.

Tabel 4.3 Klasifikasi Komponen Perhitungan *Precision, Recall & Accuracy* Skenario 1 Batas 0,0013

Batas 0,0013			
Berita Non-Hoax	Hasil	Berita Hoax	Hasil
doc 1	0.00151	doc 1	0.00108
doc 2	0.00142	doc 2	0.00174
doc 3	0.00132	doc 3	0.00185
doc 4	0.00111	doc 4	0.00168
doc 5	0.00165	doc 5	0.000976
doc 6	0.00217	doc 6	0.00121
doc 7	0.0012	doc 7	0.00181
doc 8	0.00095	doc 8	0.00144
doc 9	0.00091	doc 9	0.00165
doc 10	0.00118	doc 10	0.00236
doc 11	0.00119	doc 11	0.00136

### Keterangan Tabel:

True Positive	False Negative
False Positive	True Negative

Tabel 4.4 Nilai dari Komponen Perhitungan Dengan Batas 0,0013

	batas 0,0013
True Positive (tp)	8
False Positive (fp)	5
False Negative (fn)	3

	batas 0,0013
True Negative (tn)	6

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut.

Precision = tp / (tp + fp) Recall = tp / (tp + fn)
$$= 8 / (8 + 5) = 8 / (8 + 3)$$

$$= 0,62 = 0,72$$
Accuracy =  $\frac{\sum matc \ h(tp + tn)}{\sum tp} \times 100\%$ 

$$= \frac{8+6}{22} \times 100\%$$

$$= 64\%$$

Dari hasil perhitungan di atas maka didapatkan hasil pengujian skenario 1 dengan batas 0,0013 dengan nilai *Precision* 0,62; *Recall* 0,72; dan *Accuracy* 64%.

Berikut penjabaran dari klasifikasi dan nilai komponen perhitungan *Precision, Recall & Accuracy* dari 22 berita dari Tabel 4.4 yang menjabarkan klasifikasi dengan batas 0,0014, dan Tabel 4.5 merupakan nilai dari komponen perhitungan dengan batas 0,0014.

Tabel 4.5 Klasifikasi Komponen Perhitungan *Precision, Recall & Accuracy* Skenario 1 Batas 0,0014

Batas 0,0014			
Berita Non- Hoax	Hasil	Berita Hoax	Hasil
doc 1	0.00151	doc 1	0.00108
doc 2	0.00142	doc 2	0.00174
doc 3	0.00132	doc 3	0.00185
doc 4	0.00111	doc 4	0.00168

Berita Non- Hoax	Hasil	Berita Hoax	Hasil
doc 5	0.00165	doc 5	0.000976
doc 6	0.00217	doc 6	0.00121
doc 7	0.0012	doc 7	0.00181
doc 8	0.00095	doc 8	0.00144
doc 9	0.00091	doc 9	0.00165
doc 10	0.00118	doc 10	0.00236
doc 11	0.00119	doc 11	0.00136

## Keterangan Tabel:

True Positive	False Negative
False Positive	True Negative

Tabel 4.6 Nilai dari Komponen Perhitungan Dengan Batas 0,0014

	b <mark>at</mark> as 0,0014
True Positive	7
False Positive	4
False Negative	4
True Negative	7

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut.

Precision = tp / (tp + fp) Recall = tp / (tp + fn)
$$= 7 / (7 + 4) = 7 / (7 + 4)$$

$$= 0,64 = 0,64$$
Accuracy =  $\frac{\sum matc \ h(tp + tn)}{\sum tp} \times 100\%$ 

$$= \frac{7 + 7}{22} \times 100\%$$

Dari hasil perhitungan di atas maka didapatkan hasil pengujian skenario 1 dengan batas 0,0014 dengan nilai *Precision* 0,64; *Recall* 0,64; dan *Accuracy* 64%.

Berikut penjabaran dari klasifikasi dan nilai komponen perhitungan *Precision, Recall & Accuracy* dari 22 berita dari Tabel 4.6 yang menjabarkan klasifikasi dengan batas 0,0015, dan Tabel 4.7 merupakan nilai dari komponen perhitungan dengan batas 0,0015.

Tabel 4.7 Klasifikasi Komponen Perhitungan *Precision, Recall & Accuracy* Skenario 1 Batas 0,0015

Batas 0,0015			
Berita Non- Hoax	<mark>H</mark> asil	Berita Hoax	Hasil
doc 1	0.00151	doc 1	0.00108
doc 2	0.00142	doc 2	0.00174
doc 3	0.00132	doc 3	0.00185
doc 4	0.00111	doc 4	0.00168
doc 5	0.00165	doc 5	0.000976
doc 6	0.00217	doc 6	0.00121
doc 7	0.0012	doc 7	0.00181
doc 8	0.00095	doc 8	0.00144
doc 9	0.00091	doc 9	0.00165
doc 10	0.00118	doc 10	0.00236
doc 11	0.00119	doc 11	0.00136

## Keterangan Tabel:

True Positive	False Negative
False Positive	True Negative

Tabel 4.8 Nilai dari Komponen Perhitungan Dengan Batas 0,0015

	batas 0,0015
True Positive	6

	batas 0,0015
False Positive	3
False Negative	5
True Negative	8

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut.

Precision = tp / (tp + fp) Recall = tp / (tp + fn)
$$= 6 / (6 + 3) = 6 / (6 + 5)$$

$$= 0,67 = 0,54$$
Accuracy =  $\frac{\sum matc \ h(tp + tn)}{\sum tp} \times 100\%$ 

$$= \frac{6+8}{22} \times 100\%$$

$$= 64\%$$

Dari hasil perhitungan di atas maka didapatkan hasil pengujian skenario 1 dengan batas 0,0015 dengan nilai *Precision* 0,67; *Recall* 0,54; dan *Accuracy* 64%.

## 4.3.2 Pengujian Skenario 2

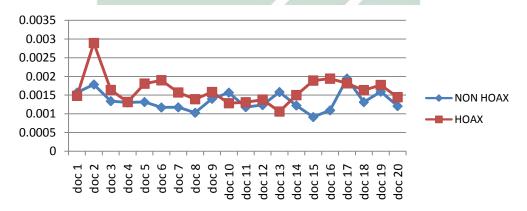
Pengujian Skenario 2 menggunakan data uji sebanyak 40 berita, yang dibagi menjadi 20 berita *non-hoax* dan 20 berita *hoax*. Berikut adalah Tabel 4.8 yang menunjukkan data hasil pengujian dari 40 berita.

Tabel 4.9 Hasil Pengujian 40 Berita

Berita Non- Hoax	Hasil	Berita Hoax	Hasil
doc 1	0.00156626	doc 1	0.001479
doc 2	0.00178153	doc 2	0.002889
doc 3	0.001337517	doc 3	0.001638

Berita Non- Hoax	Hasil	Berita Hoax	Hasil
doc 4	0.001302774	doc 4	0.001314
doc 5	0.001314324	doc 5	0.001805
doc 6	0.001168763	doc 6	0.001895
doc 7	0.001172135	doc 7	0.001567
doc 8	0.001028519	doc 8	0.001389
doc 9	0.001399543	doc 9	0.001581
doc 10	0.001565115	doc 10	0.001281
doc 11	0.001177235	doc 11	0.001309
doc 12	0.001232402	doc 12	0.001378
doc 13	0.001580199	doc 13	0.00106
doc 14	0.001221855	doc 14	0.001498
doc 15	0.00091315	doc 15	0.001887
doc 16	0.001092715	doc 16	0.001939
doc 17	0.001940282	doc 17	0.001811
doc 18	0.001311824	doc 18	0.00163
doc 19	0.001590698	doc 19	0.001768
doc 20	0.001206414	doc 20	0.001442

Berdasarkan Tabel di atas maka dapat dibuat diagram garis untuk mempermudah menentukan batas-batas untuk melakukan klsifikasi berita hoax dan non-hoax dalam sistem deteksi hoax. Berikut Gambar 4.6 merupakan tampilan grafik garis dari 40 data pengujian di atas.



Gambar 4.6 Tampilan Grafik 40 Data Uji

Berdasarkan grafik titik yang ditemukan sebagai batasan tidak jauh berbeda dengan yang didapatkan oleh pengujian skenario 1 yaitu pada titik 0,0015 beberapa berita terklasifikasi sesuai jenisnya maka untuk selanjutnya dalam tahap Analisis Hasil Sistem akan ditentukan tiga batas analisis yang sama dengan skenario 1 untuk menentukan nilai *Precision, Recall & Accuracy*yaitu batas 0,0013; batas 0,0014; batas 0,0015 dari sistem deteksi *hoax*.

Berikut penjabaran dari klasifikasi dan nilai komponen perhitungan *Precision, Recall & Accuracy* dari 40 berita dari Tabel 4.9 yang menjabarkan klasifikasi dengan batas 0,0013, dan Tabel 4.10 merupakan nilai dari komponen perhitungan dengan batas 0,0013.

Tabel 4.10 Klasifikasi Komponen Perhitungan *Precision, Recall & Accuracy*Skenario 2 Batas 0,0013

Batas 0,0013			
Berita Non- Hoax	Hasil	Berita Hoax	Hasil
doc 1	0.00156626	doc 1	0.00147948
doc 2	0.00178153	doc 2	0.002888707
doc 3	0.001337517	doc 3	0.001637899
doc 4	0.001302774	doc 4	0.001314457
doc 5	0.001314324	doc 5	0.001805396
doc 6	0.001168763	doc 6	0.001895446
doc 7	0.001172135	doc 7	0.001567057
doc 8	0.001028519	doc 8	0.001388853
doc 9	0.001399543	doc 9	0.001581351
doc 10	0.001565115	doc 10	0.00128136
doc 11	0.001177235	doc 11	0.00130875
doc 12	0.001232402	doc 12	0.00137792
doc 13	0.001580199	doc 13	0.0010603
doc 14	0.001221855	doc 14	0.001497904
doc 15	0.00091315	doc 15	0.001886784
doc 16	0.001092715	doc 16	0.001938564
doc 17	0.001940282	doc 17	0.001810898
doc 18	0.001311824	doc 18	0.001629758
doc 19	0.001590698	doc 19	0.001768372
doc 20	0.001206414	doc 20	0.001442111

## Keterangan Tabel:

True Positive	False Negative
False Positive	True Negative

Tabel 4.11 Nilai dari Komponen Perhitungan Dengan Batas 0,0013

	batas 0,0013
True Positive	18
False Positive	11
False Negative	2
True Negative	9

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut.

Precision = tp / (tp + fp) Recall = tp / (tp + fn)  
= 
$$18 / (18 + 11)$$
 =  $18 / (18 + 2)$   
=  $\mathbf{0,62}$  =  $\mathbf{0,9}$   
Accuracy =  $\frac{\sum matc \ h(tp + tn)}{\sum tp} \times 100\%$   
=  $\frac{18 + 9}{40} \times 100\%$   
=  $\mathbf{68\%}$ 

Dari hasil perhitungan di atas maka didapatkan hasil pengujian skenario 1 dengan batas 0,0013 dengan nilai *Precision* 0,62; *Recall* 0,9; dan *Accuracy* 68%.

Berikut penjabaran dari klasifikasi dan nilai komponen perhitungan *Precision, Recall & Accuracy* dari 40 berita dari Tabel 4.11 yang menjabarkan klasifikasi dengan batas 0,0014, dan Tabel 4.12 merupakan nilai dari komponen perhitungan dengan batas 0,0014.

Tabel 4.12 Klasifikasi Komponen Perhitungan *Precision, Recall & Accuracy*Skenario 2 Batas 0,0014

Batas 0,0014			
Berita Non- Hoax	Hasil	Berita Hoax	Hasil
doc 1	0.001566	doc 1	0.00147948
doc 2	0.001782	doc 2	0.002888707
doc 3	0.001338	doc 3	0.001637899
doc 4	0.001303	doc 4	0.001314457
doc 5	0.001314	doc 5	0.001805396
doc 6	0.001169	doc 6	0.001895446
doc 7	0.001172	doc 7	0.001567057
doc 8	0.001029	doc 8	0.001388853
doc 9	0.0014	doc 9	0.001581351
doc 10	0.001565	doc 10	0.00128136
doc 11	0.001177	doc 11	0.00130875
doc 12	0.001232	doc 12	0.00137792
doc 13	0.00158	doc 13	0.0010603
doc 14	0.001222	doc 14	0.001497904
doc 15	0.000913	doc 15	0.001886784
doc 16	0.001093	doc 16	0.001938564
doc 17	0.00194	doc 17	0.001810898
doc 18	0.001312	doc 18	0.001629758
doc 19	0.001591	doc 19	0.001768372
doc 20	0.001206	doc 20	0.001442111

## Keterangan Tabel:

True Positive	False Negative
False Positive	True Negative

Tabel 4.13 Nilai dari Komponen Perhitungan Dengan Batas 0,0014

	batas 0,0014
True Positive	14
False Positive	6
False Negative	6
True Negative	14

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut.

Precision = tp / (tp + fp) Recall = tp / (tp + fn)  
= 14 / (14 + 6) = 14 / (14 + 6)  
= 0,7 = 0,7  
Accuracy = 
$$\frac{\sum matc \ h(tp + tn)}{\sum tp} \times 100\%$$
  
=  $\frac{14 + 14}{40} \times 100\%$ 

Dari hasil perhitungan di atas maka didapatkan hasil pengujian skenario 1 dengan batas 0,0014 dengan nilai *Precision* 0,7; *Recall* 0,7; dan *Accuracy* 70%.

Berikut penjabaran dari klasifikasi dan nilai komponen perhitungan *Precision, Recall & Accuracy* dari 40 berita dari Tabel 4.13 yang menjabarkan klasifikasi dengan batas 0,0015, dan Tabel 4.14 merupakan nilai dari komponen perhitungan dengan batas 0,0015.

Tabel 4.14 Klasifikasi Komponen Perhitungan *Precision, Recall & Accuracy*Skenario 2 Batas 0,0015

Batas 0,0015			
Berita Non- Hoax	Hasil	Berita Hoax	Hasil
doc 1	0.001566	doc 1	0.001479
doc 2	0.001782	doc 2	0.002889
doc 3	0.001338	doc 3	0.001638
doc 4	0.001303	doc 4	0.001314
doc 5	0.001314	doc 5	0.001805
doc 7	0.001172	doc 7	0.001567
doc 8	0.001029	doc 8	0.001389
doc 9	0.0014	doc 9	0.001581
doc 10	0.001565	doc 10	0.001281
doc 11	0.001177	doc 11	0.001309
doc 12	0.001232	doc 12	0.001378
doc 13	0.00158	doc 13	0.00106
doc 14	0.001222	doc 14	0.001498
doc 15	0.000913	doc 15	0.001887
doc 16	0.001093	doc 16	0.001939
doc 17	0.00194	doc 17	0.001811
doc 18	0.001312	doc 18	0.00163
doc 19	0.001591	doc 19	0.001768
doc 20	0.001206	doc 20	0.001442

# Keterangan Tabel:

True Positive	False Negative
False Positive	True Negative

Tabel 4.15 Nilai dari Komponen Perhitungan Dengan Batas 0,0015

	batas 0,0015
True Positive	11
False Positive	6
False Negative	9
True Negative	14

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut.

Precision = 
$$tp / (tp + fp)$$
 Recall =  $tp / (tp + fn)$   
=  $11 / (11 + 6)$  =  $11 / (11 + 9)$   
=  $0,65$  =  $0,55$ 

Accuracy = 
$$\frac{\sum matc \ h(tp+tn)}{\sum tp} \times 100\%$$
$$= \frac{11+14}{40} \times 100\%$$
$$= 63\%$$

Dari hasil perhitungan di atas maka didapatkan hasil pengujian skenario 1 dengan batas 0,0014 dengan nilai *Precision* 0,65; *Recall* 0,55; dan *Accuracy* 63%.

Berdasarkan hasil pengujian dua skenario di atas, jika di bandingkan dari kekonsistenan nilai antara *Precision, Recall* dan *Accuracy*, maka batas 0,0014 pada skenario 2 merupakan batas yang konsisten untuk dijadikan pedoman pengklasifikasian dokumen berita, karena nilai dari *Precision, Recall* dan *Accuracy* sama satu sama lain yaitu 0,7 dan 70. Namun tidak hanya hasil dari *Precision, Recall* dan *Accuracy* yang dapat menentukan keakuratan sistem deteksi *hoax*, tetapi juga banyaknya data berita *hoax* yang dijadikan *library* akan semakin meningkatkan keakuratan sistem untuk mendeteksi sebuah berita, hal ini dapat dilihat dari skenario 2 yang memiliki data kata *hoax* lebih banyak dibandingkan dengan skenario 1, membuat hasil dari *Precision, Recall* dan *Accuracy* skenario 2 lebih tinggi dibandingkan dengan skenario 1.

#### 4.4 Pembahasan

Berdasarkan hasil pengujian sistem, didapatkan batas 0,0014 pada skenario 2 sebagai batas yang memiliki nilai yang konsisten antara *Precision, Recall* dan *Accuracy* nya. Hal tersebut dapat disebabkan karenadata yang digunakan pada skenario 2 lebih banyak jika dibandingkan dengan data pada skenario 1.Sehingga di dalam perhitungan Tf-Idf, jumlah kemunculan kata pada dokumen yang disimpan sangat mempengaruhi.Karena banyaknya kata, kemunculan kata dan banyaknya dokumen yang digunakan sangat mempengaruhi perhitungan bobot sebuah kata yang dibandingkan. Maka dari itu penambahan perhitungan diperlukan untuk memastikan sebesar apa frekuensi kemunculan sebuah kata dalam dokumen, dan seberapa besar pengaruhnya dalam dokumen tersebut.

Kemudian untuk nilai akurasi yang belum bisa mencapai angka prosentase atas mendekati sempurna, dimungkinkan karenalibrary berita hoaxmenyimpan berita dengan topik yang berbeda-beda, namun bukan sebuah berita yang terus menerus disebarkan, sehingga jika dibandingkan dengan berita lain yang juga memiliki topik yang berbeda-beda, pengaruhnya tidak akan sebesar jika berita tersebut merupakan berita yang terus menerus dijadikan topik hangat untuk menyebarkan kebohongan. Sehinngga hal ini juga sangat berpengaruh pada perhitungan bobot kata. Dalam dokumen penyimpanan berita hoax terdapat beberapa topik yang memang sering dijadikan sasaran untuk penyebaran berita hoax, namun berita tersebut merupakan berita yang sangat spesifik seputar dunia politik, sehingga hal tersebut sangat mempengaruhi keberadaan dokumen dengan berita hoax yang memang jarang menjadi sasaran pemberitaan dan jumlahnya tidak mendominasi. Semantara untuk data uji, tidak semua berita dalam data uji membahas tentang topik politik, sehingga terdapat beberpa nilai false negative dan false porsitive yang mempengaruhi hasil akhir Precision, Recall dan Accuracy.

## BAB V

#### **PENUTUP**

### 5.1 Kesimpulan

Berdasarkan hasil dari penelitian tentang Deteksi *Hoax* Pada Sosial Media Berbahasa Indonesia Menggunakan Metode *Levenshtein Distance*, maka dapat disimpulkan bahwa:

- 1. Terdapat langkah-langkah untuk menerapkan Metode Levenshtein Distance dalam Sistem Deteksi *Hoax* yaitu:
  - a. Pembuatan Dokumen Data Target yang di dalamnya terdapat kumpulan kata *hoax* yang sudah disederhanakan dalam Prapemrosesan Kata dan Penyeleksian kata dengan memberi bobot pada setiap kata menggunakan Tf-Idf.
  - b. Pembuatan Sistem Deteksi *Hoax* yang di dalamnya terdapat beberapa proses hingga menghasilkan nilai klasifikasi yaitu Prapemrosesan kata sumber, membandingkan kata sumber dan kata target, menghitung jarak (*Levenshtein Distance*), memberi bobot (Tf-idf), dan menghitung hasil akhir sekligus pengklasifikasian.
- 2. Penerapan metode *Levenstein Distance* yang dipadukan dengan Tf-Idf terbukti mampu membedakan antara berita *hoax* dan tidak, dengan tingkat akurasi yang cukup bagus.
- 3. Batas 0,0014 pada skenario 2 yang memiliki data latih sebanyak 100 berita terindikasi *hoax* dan 40 berita sebagai data uji dengan pembagian 20 berita *non-hoax* dan 20 berita *hoax*, memiliki nilai *Precision*, *Recall* dan *Accuracy* yang konsisten yaitu *Precision* 0,7; *Recall* 0,7 dan *Accuracy* 70%. Yang berarti semakin banyak kata *hoax* yang dijadikan data latih, maka semakin akurat sistem melakukan pendeteksian.

#### 5.2 Saran

Berdasarkan kesimpulan yang telah dijabarkan di atas, maka berikut adalah saran yang siberikan untuk peneltian yang akan datang:

- 1. Diharapkan pada penelitian selanjutnya menggunakan berita dengan rentang waktu terkini hingga batas lampau yang diperlukan untuk meningkatkan akurasi pengecekan berita-berita terbaru.
- 2. Untuk dapat meningkatkan nilai akurasi maka diperlukan beberapa hal yang diprediksi dapat meningkatkan nilai akurasi yaitu dengan selalu memperbarui library kata hoax beserta nilai bobotnya, mengelompokkan library kata hoax sesuai dengan topiknya dan memberikan pilihan pendeteksian berita berdasarkan topiknya juga. Dengan begitu diharapkan perhitungan jarak kata beserta pembobotannya akan lebih spesifik dan dapat meningkatkan perhitungan akurasinya.
- 3. Diharapkan pada penelitian selanjutnya dapat menerapkan metode lain seperti Self-Organizing Map, Feed-forward Neural Networ, Naïve Bayes, Support Vector Machine, Algoritma C4.5, dan lain sebagainya.

### **DAFTAR PUSTAKA**

- CNN Indonesia. "Ada 800 Ribu Situs Penyebar *Hoax* di Indonesia". 2016. [Online]. Tersedia: http://www.cnnindonesia.com. Diakses pada tanggal 6 Maret 2018 pukul 12.19 WIB.
- Downey, A. B. (2012). Think Python. Sebastopol, California: O'REILLY.
- Afriansyah, Z., & Puspitaningrum, D. (2015). MENGGUNAKAN ALGORITMA LEVENSHTEIN DISTANCE (Studi Kasus: DNA Kanker Hati Manusia), 3(2), 61–67.
- Forum Anti Fitnah, Hasut dan Hoax. Arsip Hasil Diskusi Grup Forum Anti Fitnah, Hasut dan Hoax. 2015. [Online]. Tersedia: http://turnbackhoax.id/. Diakses pada tangggal 12 Maret 2018 pukul 10.00 WIB.
- har07.Library Sastrawi Python. 2016. [Online]. Tersedia: https://github.com/har07/PySastrawi. Diakses pada tanggal 14 April 2018 pukul 13.00 WIB.
- Ishak, A., Chen, Y. Y., & Yong, S. P. (2012). Distance-based hoax detection system. 2012 International Conference on Computer and Information Science, ICCIS 2012 - A Conference of World Engineering, Science and Technology Congress, ESTCON 2012 - Conference Proceedings, 1, 215— 220. https://doi.org/10.1109/ICCISci.2012.6297242
- Katariya, N. P., & Chaudhari, M. S. (2015). Text Preprocessing for Text Mining Using Side Information, *3*, 3–7.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval Introduction. Computational Linguistics (Vol. 35). https://doi.org/10.1162/coli.2009.35.2.307
- Mulyanto, A. (2010). Analisis Edit Distance Menggunakan Algoritma Dynamic Programming. *Saintek*, 5(2), 1–10. Retrieved from

- http://repository.ung.ac.id/get/simlit\_res/1/281/Analisis-Edit-Distance-Menggunakan-Algoritma-Dynamic-Programming.pdf
- Rasywir, E., & Purwarianti, A. (2015). Eksperimen pada Sistem Klasifikasi Berita Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin. *Jurnal Cybermatika*, 3(2), 1–8.
- Saadah, M. N., Atmagi, R. W., Rahayu, D. S., & Arifin, A. Z. (2013). Sistem

  Temu Kembali Dokumen Teks dengan Pembobotan Tf-Idf Dan LCS. *Jurnal Ilmiah Teknologi Informasi (JUTI)*, 11(1), 17–20.

  https://doi.org/10.12962/j24068535.v11i1.a16
- Stezar Priansya. (2017). Social Media Text Normalization Using Word2vec.

  Levenshtein Distance, and Jaro-Winkler Distance. Institut Teknologi
  Sepuluh Nopember Surabaya, Final Project KS 141501.
- Syafitri, N., (2010). Perbandingan Metode K-Nearest Neighbor (KNN) dan Metode Nearest Cluster Classifier (NCC) dalam Pengklasifikasian Kualitas Batik Tulis. *J. Teknologi Informasi dan Pendidikan*, 2, 1, 42 53.
- Vuković, M., Pripužić, K., & Belani, H. (2009). An intelligent automatic hoax detection system. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5711 LNAI(PART 1), 318–325. https://doi.org/10.1007/978-3-642-04595-0\_39
- Zhang, Y., Gong, L., & Wang, Y. (2005). An improved TF-IDF approach for text classification. *Journal of Zhejiang University SCIENCE*, 6(1), 49–55. https://doi.org/10.1631/jzus.2005.A0049