

**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 *LEVENSHTAIN 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 sebenarnya.

Surabaya, 03 Agustus 2018

Yang membuat pernyataan,


Frista Gifti Weddiningrum
NIM: H76214015

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh :

NAMA : FRISTA GIFTI WEDDININGRUM

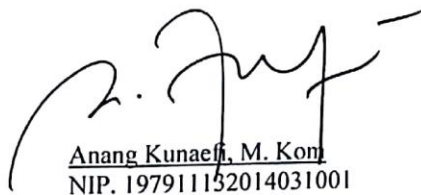
NIM : H76214015

JUDUL : DETEKSI KONTEN *HOAX* BERBAHASA INDONESIA PADA
MEDIA SOSIAL MENGGUNAKAN METODE *LEVENSHTAIN*
DISTANCE

Ini telah diperiksa dan disetujui untuk diujikan.

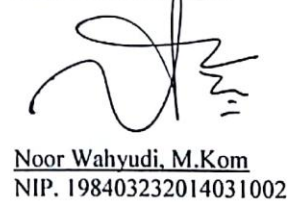
Surabaya, 23 Juli 2018

Dosen Pembimbing I



Anang Kunaefi, M. Kom
NIP. 197911132014031001

Dosen Pembimbing II



Noor Wahyudi, M. Kom
NIP. 198403232014031002

LEMBAR PENGESAHAN

Skripsi oleh Frista Gifti Weddiningrum ini telah dipertahankan

Di depan tim Penguji Skripsi

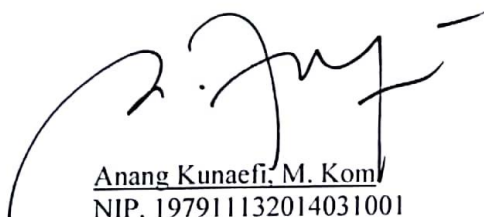
Surabaya, 25 Juli 2018

Mengesahkan Fakultas Sains dan Teknologi

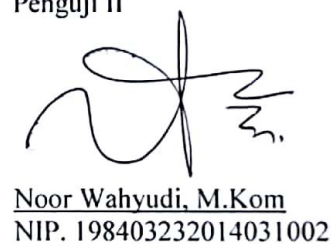
Universitas Islam Negeri Sunan Ampel Surabaya

Susunan Dewan Penguji

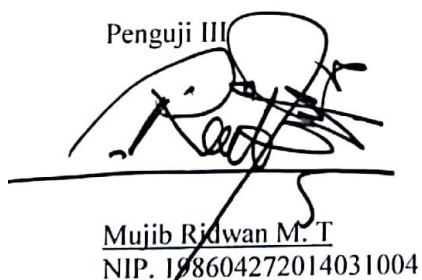
Penguji I


Anang Kunaefi, M. Kom
NIP. 197911132014031001

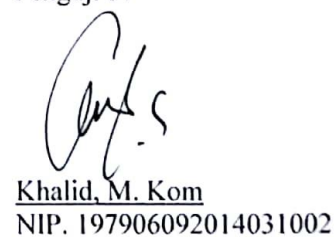
Penguji II


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
Universitas Islam Negeri Sunan Ampel Surabaya



Dr. Eni Purwati, M.Ag
NIP. 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

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :

☒ Skripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)

yang berjudul :

DETEKSI KONTEN HOAX BERBAHASA INDONESIA PADA MEDIA SOSIAL

MENGUNAKAN METODE LEVENSHTAIN DISTANCE

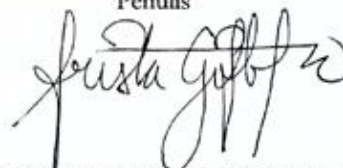
beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara *fulltext* untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 8 Agustus 2018.

Penulis



(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 memperlancar 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 metode *Levenshtein Distance* (LD) untuk mengukur jarak antar kata dalam dokumen. Penerapan Metode *Levenshtein Distance* dalam Sistem Deteksi *Hoax* memiliki beberapa tahap yang dimulai dengan tahap pra-pemrosesan kata, dilanjutkan dengan tahap perhitungan Tf-Idf dan kemudian 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*

DAFTAR ISI

LEMBAR PERNYATAAN KEASLIAN	i
LEMBAR PERSETUJUAN PEMBIMBING	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI	iv
ABSTRAK	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR 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.....	4
1.6 Sistematika Penulisan Skripsi	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Tinjauan Penelitian Terdahulu	5
2.2 Teori yang Digunakan	8
2.2.1 <i>Natural Language Processing</i>	8
2.2.2 TF-IDF	9
2.2.3 <i>Levenshtein Distance</i>	10
2.2.4 Media Sosial.....	12
2.2.5 Bahasa Pemrograman Python	13
2.2.6 Pengukuran Performa.....	13
2.3 Integrasi Keilmuan	15
BAB III METODOLOGI PENELITIAN	17
3.1 Identifikasi Masalah dan Studi Literatur	17
3.2 Pengumpulan Data	19
3.3 Pra-pemrosesan Data Target.....	22

DAFTAR TABEL

Tabel 2.1 Penelitian Sebelumnya.....	5
Tabel 2.2 <i>Precision and Recall</i>	14
Tabel 3.1 <i>Precision and Recall</i> Berdasarkan Data Pengujian.....	30
Tabel 4.2 Hasil Pengujian 22 Berita	50
Tabel 4.3 Klasifikasi Komponen Perhitungan <i>Precision, Recall & Accuracy</i> Skenario 1 Batas 0,0013.....	51
Tabel 4.4 Nilai dari Komponen Perhitungan Dengan Batas 0,0013	51
Tabel 4.5 Klasifikasi Komponen Perhitungan <i>Precision, Recall & Accuracy</i> Skenario 1 Batas 0,0014.....	52
Tabel 4.6 Nilai dari Komponen Perhitungan Dengan Batas 0,0014.....	53
Tabel 4.7 Klasifikasi Komponen Perhitungan <i>Precision, Recall & Accuracy</i> Skenario 1 Batas 0,0015.....	54
Tabel 4.8 Nilai dari Komponen Perhitungan Dengan Batas 0,0015	54
Tabel 4.9 Hasil Pengujian 40 Berita	55
Tabel 4.10 Klasifikasi Komponen Perhitungan <i>Precision, Recall & Accuracy</i> Skenario 2 Batas 0,0013.....	57
Tabel 4.11 Nilai dari Komponen Perhitungan Dengan Batas 0,0013	58
Tabel 4.12 Klasifikasi Komponen Perhitungan <i>Precision, Recall & Accuracy</i> Skenario 2 Batas 0,0014.....	59
Tabel 4.13 Nilai dari Komponen Perhitungan Dengan Batas 0,0014	60
Tabel 4.14 Klasifikasi Komponen Perhitungan <i>Precision, Recall & Accuracy</i> Skenario 2 Batas 0,0015.....	61
Tabel 4.15 Nilai dari Komponen Perhitungan Dengan Batas 0,0015	61

BAB I

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 akan dengan 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

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:

فَعَلُّمُ نَادِمِينَ

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, Pripuzić, & Belani, 2009).

[illegible]

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 perhitungan Tf - 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*.

1.2 Perumusan Masalah

1.3 Batasan Masalah

1. 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.

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

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 menjabarkan perbandingan penelitian sebelumnya yang menjadi dasar dari penelitian ini.

Tabel 2.1 Penelitian Sebelumnya

Judul	Peneliti	Metode	Tahun	Data Set	Kelebihan (+) dan Kekurangan (-)
<i>An Intelligent Automatic Hoax Detection-Sistem</i>	Marin Vuković, Krešimir Pripužić, dan Hrvoje Belani	<ul style="list-style-type: none"> - Self-Organizing Map - Feed-Forward Neural Network 	2009	Email berbahasa Inggris	(+) Sistem dapat membedakan dan mengklasifikasi <i>email hoax</i> baru dengan membandingkan polanya dengan pola tersimpan yang sama.
					(-) Jika terdapat <i>email</i> dengan pola yang baru, sistem belum dapat membedakannya karena pola tersebut tidak ada dalam

Judul	Peneliti	Metode	Tahun	Data Set	Kelebihan (+) dan Kekurangan (-)
					Penyimpanan sehingga sistem akan menganggap <i>email</i> dengan pola baru sebagai <i>email</i> yang tidak mengandung <i>hoax</i> .
<i>Distance-based Hoax Detection Sistem</i>	Adzlan Ishak, Y. Y. Chen, dan Suet-Peng Yong	<i>Levenshtein Distance</i>	2012	<i>Email</i> berbahasa Inggris	(+) Dalam pengujian, sistem menghasilkan nilai <i>positive predictive value</i> sebesar 0,96.
					(-) Sistem belum bisa mengidentifikasi <i>email</i> yang orisinal. Seluruh <i>email</i> akan diukur tingkat kandungan <i>hoax</i> nya.
Sistem Temu Kembali Dokumen Teks dengan Pembobotan Tf-Idf Dan LCS	Munjiah Nur Saadah, Rigga Widar Atmagi, Dyah S. Rahayu, Agus Zainal Arifin	<i>Tf-Idf, LCS</i>	2013	Dokumen Teks	(+) Memiliki nilai recall sebesar 96,84% karena sistem mampu bekerja secara efektif dalam mengembalikan 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 Kekurangan (-)
					(-) memiliki nilai presisi sebesar 30,36%.
Eksperimen pada Sistem Klasifikasi Berita Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin	Errissya Rasywir, Ayu Purwarianti	<i>Naïve Bayes, Support Vector Machine, Algoritma C4.5</i>	2015	Artikel Berbahasa Indonesia	<p>Dalam penelitian ini, peneliti melakukan percobaan menggunakan lebih dari satu metode pada setiap proses dalam sistem klasifikasi <i>hoax</i>. Hasil penelitian tersebut berupa metode dan algoritma apa saja yang paling akurat untuk mengklasifikasikan berita <i>hoax</i>.</p> <p>Dalam proses prapemrosesan teks, peneliti menemukan prapemrosesan tanpa <i>stemming</i> yang mencapai hasil terbaik.</p> <p>Dalam proses ekstraksi fitur, fitur unigram memiliki akurasi terbaik.</p> <p>Dan untuk algoritma klasifikasi, algoritma <i>Naïve Bayes</i> menunjukkan hasil akurasi terbaik yaitu</p>

3. Stemming

2.2.2 TF-IDF

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).

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

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

9

$\sum freq(t)$: 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).

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

Keterangan:

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

|D| : 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).

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

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):

Keterangan:

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

maxLength adalah jumlah string dari kata yang terpanjang antara stradan strb.

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

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).

Keterangan:

$\Sigma match$: jumlah deteksi yang benar

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 pembagiannya adalah total dari seluruh data yang digunakan dalam pengujian.

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 khusus *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.

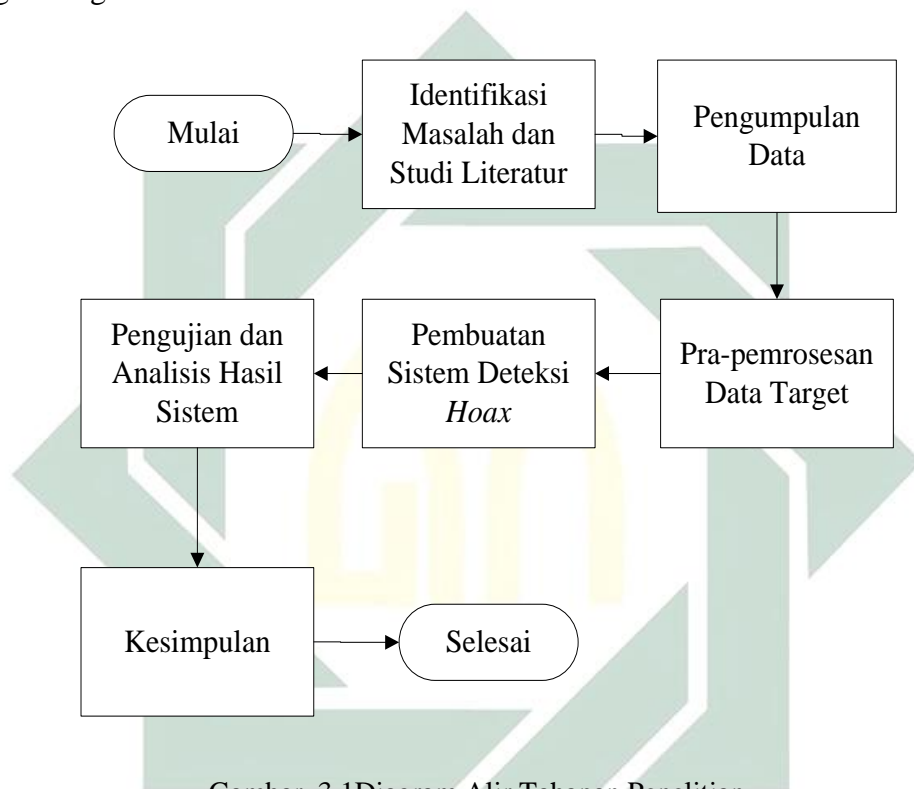
[illegible]

1. 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.

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 kebohohonan kecil dapat menjadi besar jika kebohongan tersebut dipercaya dan menyebabkan kehancuran jika kebohongan tersebut dilaksanakan.

BAB III

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



Gambar 3.1 Diagram 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

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).

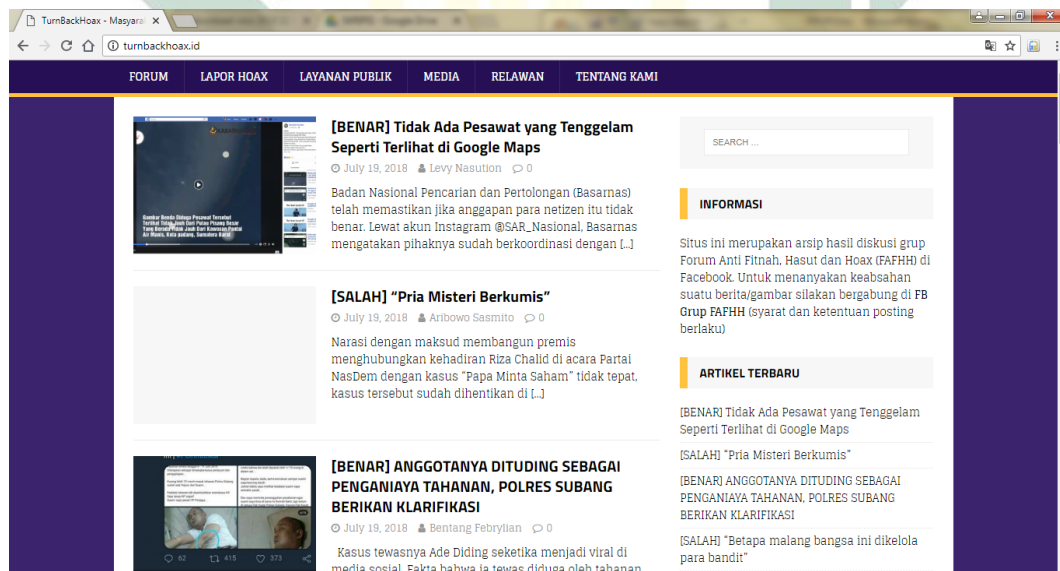
Selanjutnya adalah proses perhitungan jarak antar kata. Dalam penelitian ini telah ditentukan untuk menggunakan perhitungan jarak antar kata dibandingkan dengan algoritma *artificial intelegent* dikarenakan berdasarkan penelitian yang dirujuk dengan judul *Distance-based Hoax Detection Sistem* yang di 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

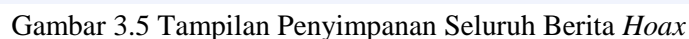
Dalam penelitian ini, data yang digunakan adalah data berita *hoax*, baik itu dari sosial media ataupun dari situs *web* yang khusus menyajikan berita untuk khalayak umum. turnbackhoax.id adalah sebuah situs *web* yang menyediakan data-data tersebut, terlebih situs ini mengkhususkan diri untuk menyediakan 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.

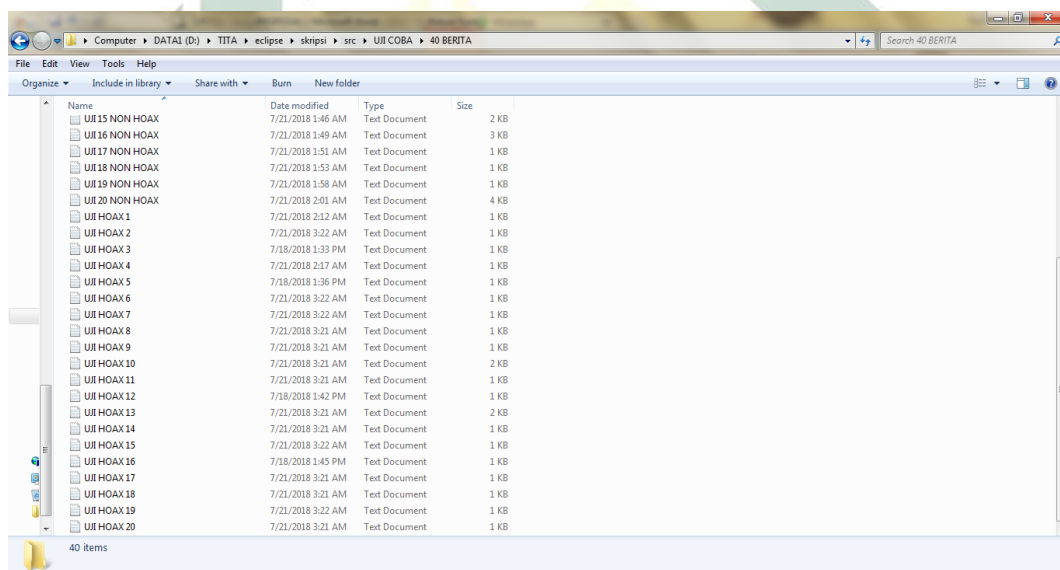
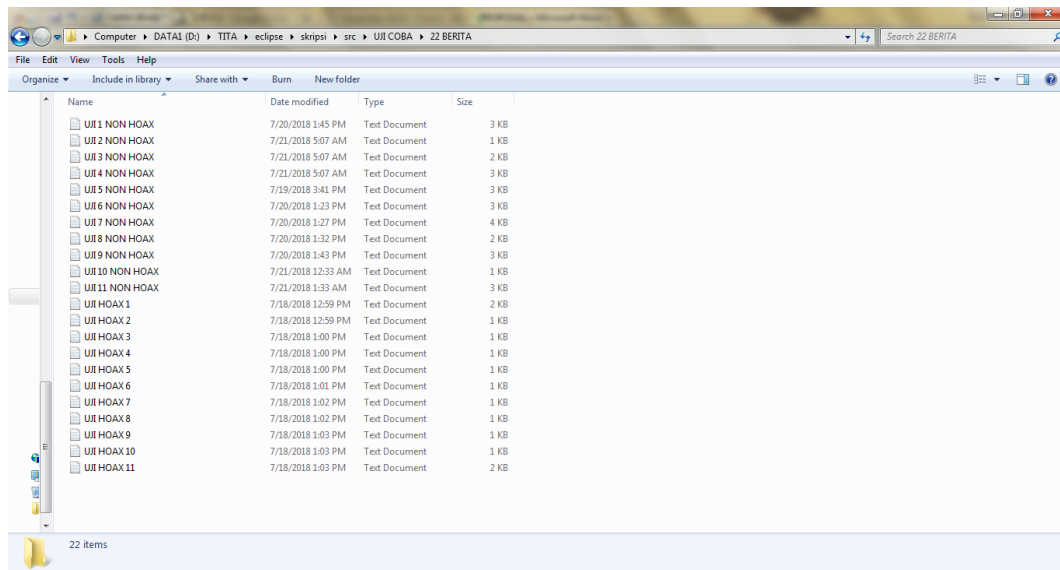
Berikut adalah Gambar 3.2 yang menyajikan tampilan dari halaman awal situs *web* turnbackhoax.id dan Gambar 3.3 yang menyajikan tampilan salah satu halaman berita terklasifikasi *hoax*.



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

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 3.7 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

but di atas dapat dijabarkan dalam Diagram Alir di Gambar 3.8 di bawah

```

graph TD
    Start([Mulai]) --> Text[/Teks Berita/]
    Text --> Remove[Menghapus Stopwords]
    Remove --> Stemming[Stemming]
    Stemming --> Analyze[Menganalisis Leksikal]
    Analyze --> HoaxDoc[Dokumen kata hoax]
    Analyze --> Calculate[Menghitung Komponen Tf-Idf]
    Calculate --> TargetDoc[Dokumen target]
    Calculate --> Decision{Ada Berita lagi?}
    Decision -- YA --> Remove
    Decision -- TIDAK --> End([Selesai])
  
```

a. Penghapusan *Stopwords*

Kalimat utuh:

24

Kalimat hasil proses Stemming → Politisi kabar jerat kasus korupsi

0 1 2 3 4
[indeks] → [Politisi] [kabar] [jerat] [kasus] [korupsi]

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, sistem akan 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.

26

```

graph TD
    Start([Mulai]) --> Input[/Data Kata Hoax.  
Data Berita/]
    Input --> Process1[Menghitung kemunculan kata hoax dalam dokumen]
    Process1 --> Process2[Menghitung total kata dalam dokumen]
    Process2 --> Process3[Menghitung dokumen kemunculan kata hoax]
    Process3 --> Decision{Kata Hoax habis?}
    Decision -- YA --> Process4[Menghitung Tf-Idf]
    Decision -- TIDAK --> Process3
    Process4 --> End([Selesai])
    Process4 -.-> Output[/Data Target/]
    Process3 -.-> Storage[/Penyimpanan Sementara Komponen/]
    
```

3.4 Pembuatan Sistem Deteksi *Hoax*

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

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 substitusi 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 rata-rata seluruh hasil akhir setiap kata sumber yang dibandingkan.

```

graph TD
    Mulai([Mulai]) --> TeksBerita[/Teks Berita/]
    TeksBerita --> MenghapusStopwords[Menghapus Stopwords]
    MenghapusStopwords --> Stemming[Stemming]
    Stemming --> MenganalisisLeksikal[Menganalisis Leksikal]
    MenganalisisLeksikal --> Membandingkan[Membandingkan data target dan data sumber]
    DataTarget[/Data Target/] -.-> Membandingkan
    Membandingkan --> MenghitungJarak[Menghitung jarak]
    MenghitungJarak --> MenghitungHasilAkhir[Menghitung hasil akhir]
    MenghitungHasilAkhir -.-> MenghitungJarak
    MenghitungHasilAkhir --> MenampilkanHasil[Menampilkan Hasil]
    MenampilkanHasil --> Selesai([Selesai])
  
```

28

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 22 berita yang terdiri dari 11 berita *hoax* dengan rentang April 2016 – Mei 2016 dan 11 *non-hoax* dengan rentang waktu Oktober 2015 – Desember 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 – Januari 2017. Kemudian teks berita tersebut akan diolah di dalam sistem dan menghasilkan nilai akhir pendeteksian. Semakin besar nilai yang dihasilkan maka 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 data-data 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*.

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

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

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

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

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.

$$\text{Precision} = \text{tp} / (\text{tp} + \text{fp}) \quad (6)$$

$$\text{Recall} = \text{tp} / (\text{tp} + \text{fn}) \quad (7)$$

Kemudian berikut adalah rumus akurasinya:

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

Keterangan:

ac : tingkat akurasi (%)

$$\Sigma match : \text{jumlah dokumen } hoax \text{ terklasifikasi } hoax \text{ dan } non-hoax \\ \text{terklasifikasi } non-hoax$$
$$\sum tp$$
 : 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

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*, *Stemming* 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
```


dalam proses Analisis Leksikal: `lexicalAnalysis[] ← stemming`. Dalam proses Analisis Leksikal, kata *hoax* yang telah tersaring akan ditampung di dalam *list*.

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 akan menyimpan data kata *hoax*.

```

Deklarasi:
k = integer
maxRangeArray = integer
temp = String
lexicalAnalysis[] = String
uniqueArray[] = String
openDoc = String
writeDoc = String

Deskripsi:
for k <maxRangeArray
str←lexicalAnalysis[k]
for x ← str
if x !=uniqueArray
thenuniqueArray[] ← x

openDoc← (Directory: hoaxDoc.txt)
for m <maxRangeArray
writeDoc← uniqueArray[m]
openDoc← writeDoc

```

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.

4.1.2 Proses Perhitungan Komponen Tf-Idf

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.

[illegible]

proses perhitungan selanjutnya adalah menghitung berapa kali kata tersebut dapat dalam sebuah dokumen, dalam proses counting menghasilkan nilai $\sum t$ yang merupakan komponen penting dari TFIDF adalah *code* dari perhitungan banyak kata dalam dokumen dibandingkan.

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
  numb ← i + 1
  openDocToRead ← (Directory: doc(numb))
  readDoc ← openDocToRead
  for j ← readDoc
    lexicalAnalysis[] ← j
  amount ← amount + lengthLexicalAnalysis
  openDocToSave ← (Directory: doc.csv)
  insertDoc ← amount
  openDocToSave ← insertDoc
  amount ← 0

```

`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

4.2.1 Spesifikasi Sistem

Spesifikasi komputer yang digunakan untuk membuat sistem *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 bisa dilakukan adalah Python versi 2.7 ke bawah, untuk penggunaan versi 3 ke atas, tidak direkomendasikan bagi pengguna komputer dengan RAM 2GB.

Library-library Python yang digunakan untuk membuat sistem deteksi *hoax* adalah *LibraryNLTK*, *Library Sastrawi*, dan *Library* *Stopwords* dan *Stemming* Berbahasa Indonesia, sementara *Library* *distance* digunakan untuk memanggil fungsi perhitungannya. *Library* *Stopwords* dan *Stemming* Berbahasa Indonesia, sementara *Library* *distance* digunakan untuk memanggil fungsi perhitungannya.

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.

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.

40

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*.

44

```
i = String
saveSourceWords[] = String

Deskripsi:
similarity ← 1 - (minValue/maxLength)
tfidfMix ← similarity * b
count ← count + tfidfMix
saveFirstValue[] ← count
if i != saveSourceWords
then count ← 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 ($\text{similarity} \leftarrow 1 - (\text{minValue}/\text{maxLength})$). Kemudian hasil *similarity* digabungkan dengan nilai Tf-Idf (b) dari kata target (a) dalam perkalian ($\text{tfidfMix} \leftarrow \text{similarity} * b$). 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 ($\text{averageFirstValue} \leftarrow \text{count}/\text{lengthSaveFirstValue}$) kata target (a) yang memiliki jarak minimal terhadap kata sumber (i). Hasil rata-rata

tersebut disimpan ke dalam `listsaveFirstAverage[]`. Setelah mendapatkan rata-rata perhitungan, maka proses selanjutnya adalah menghitung rata-rata dari seluruh hasil akhir yang disimpan di dalam `listsimpan_rerata`.

Dalam *code* terdapat fungsi `delete saveFirstValue[]` digunakan untuk menghapus seluruh isi di dalam `listsaveFirstValue[]` 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.

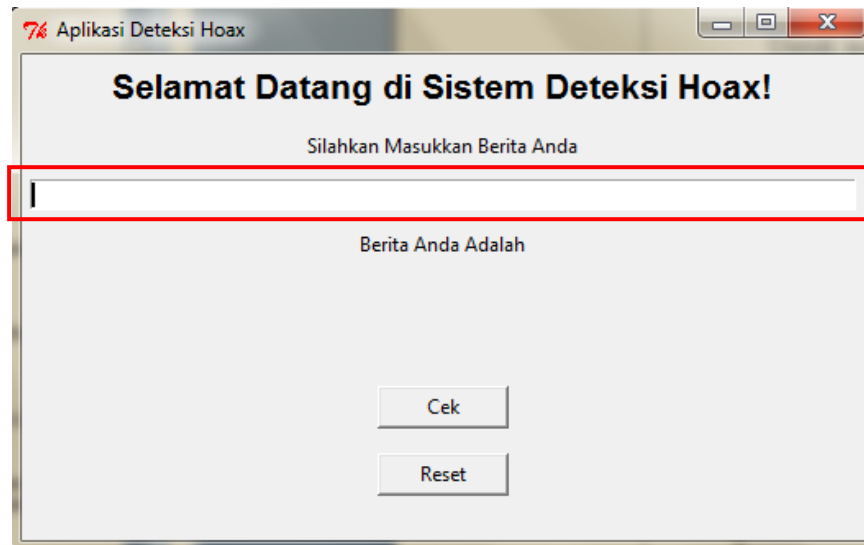
```

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

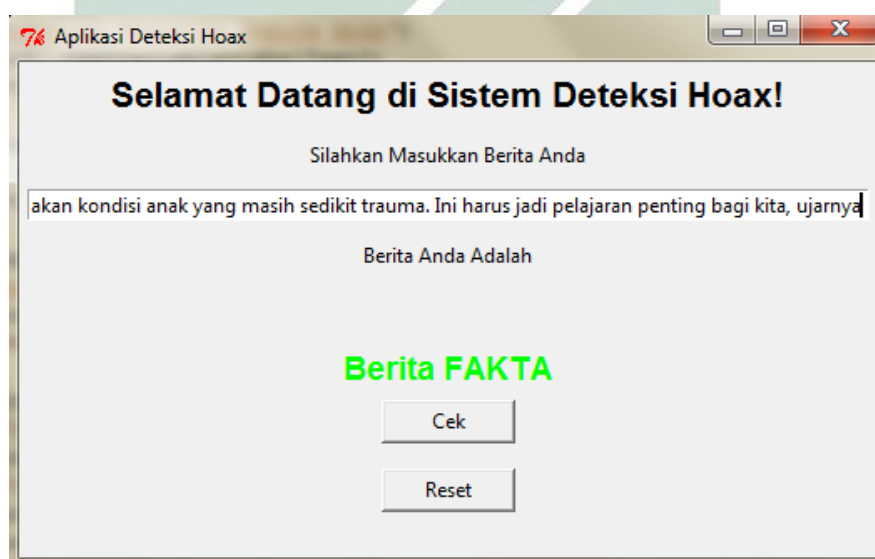
Deskripsi:
lastValue ← countAverage / lengthSaveSourceWords
if lastValue ≥ limitValue
then Output ← "Berita berkonten HOAX"
else if lastValue < limitValue
  then Output ← "Berita merupakan FAKTA"

```

Rumus $\text{lastValue} \leftarrow \text{countAverage} / \text{lengthSaveSourceWords}$ 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 ($\text{lengthSaveSourceWords}$). Perhitungan tersebut juga dapat disebut dengan rata-rata dari seluruh hasil akhir setiap kata sumber.



Untuk memasukkan berita jika hendak melakukan pengecekan, maka salin teks berita dari sumber dan tempel 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*.

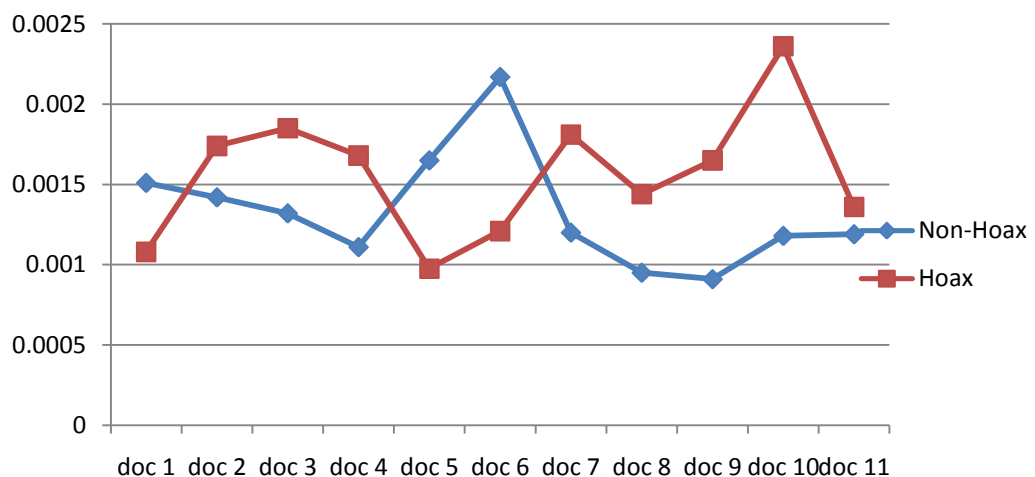


Gambar 4.3 Tampilan Pengecekan Berita *Non-Hoax*

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 klasifikasi 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

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.

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

<i>True Positive</i>	<i>False Negative</i>
<i>False Positive</i>	<i>True Negative</i>

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

= 64%

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	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:

<i>True Positive</i>	<i>False Negative</i>
<i>False Positive</i>	<i>True Negative</i>

Tabel 4.8 Nilai dari Komponen Perhitungan Dengan Batas 0,0015

	batas 0,0015
<i>True Positive</i>	6

	batas 0,0015
<i>False Positive</i>	3
<i>False Negative</i>	5
<i>True Negative</i>	8

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

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

$= 6 / (6 + 3)$ $= 6 / (6 + 5)$

$= \mathbf{0,67}$ $= \mathbf{0,54}$

Accuracy = $\frac{\sum match(tp+tn)}{\sum tp} \times 100\%$

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

$= \mathbf{64\%}$

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

Tabel 4.13 Nilai dari Komponen Perhitungan Dengan Batas 0,0014

	batas 0,0014
<i>True Positive</i>	14
<i>False Positive</i>	6
<i>False Negative</i>	6
<i>True Negative</i>	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

$$\begin{aligned}\text{Accuracy} &= \frac{\sum \text{match}(tp+tn)}{\sum tp} \times 100\% \\ &= \frac{14+14}{40} \times 100\% \\ &= \mathbf{70\%}\end{aligned}$$

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.

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

$$\begin{aligned}\text{Accuracy} &= \frac{\sum \text{match}(tp+tn)}{\sum tp} \times 100\% \\ &= \frac{11+14}{40} \times 100\% \\ &= \mathbf{63\%}\end{aligned}$$

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 dibandingkan 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.

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 Pra-pemrosesan 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 Pra-pemrosesan kata sumber, membandingkan kata sumber dan kata target, menghitung jarak (*Levenshtein Distance*), memberi bobot (Tf-idf), dan menghitung hasil akhir sekaligus 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

- 68

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>