Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)
Vol.19. No.2, Agustus 2020, pp. 74-84
P-ISSN : 1978-6603
E-ISSN : 2615-3475

Perancangan Sistem Pendeteksi Berita Hoax Menggunakan Algoritma Levenshtein Distance Berbasis Php

Nurhayati, Aprilianda Pasaribu
Teknik Informatika, STMIK Kaputama

Article Info
Article history:
Received Jun 12th, 2020
Revised Aug 8th, 2020
Accepted Aug 12th, 2020

Keyword
Hoax, Levenshtein Distance, TF-IDF, Prepocessing Text, Detection System

ABSTRACT
Di era 4.0 dimana Internet menjadi bagian penting dalam kehidupan saat ini, informasi dapat dengan mudah di akses kapanpun dan dimanapun. Namun tidak seluruh informasi yang disebarkan melalui internet berupa fakta. Data yang dipaparkan oleh Kementrian Komunikasi dan Informatika berdsarkan survey yang dilakukan pada tahaun 2018 menyebut sebanyak 800.000 situs di Indonesia terindikasi penyebar berita non-fakta atau hoax. Akibat yang ditimbulkan berita hoax sangat berbahaya karena menyerang pikiran alam bawah sadar manusia, sehingga sangat dibutuhkan sistem yang dapat mendeteksi berita hoax. Dalam penelitian ini digunakan database yang berisi dokumen berita hoax. Algoritma yang diterapkan adalah algoritma TF-IDF untuk mengukur bobot suatu kata dalam dokumen hoax dan dikombinasikan dengan algoritma Levenshtein 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 (prepocessing text) dilanjutkan dengan tahap perhitungan TF-IDF dankemudian tahap perhitungan jarak minimum antar kata menggunakan algoritmaLevenshtein Distance. Hasil batas 0,1 pada 40 dokumen yang sudah terklasifikasi sebagai data uji memiliki nilai Precision, Recall dan Accuracy yang tinggi, yaitu Precision1; Recall0,71;dan Accuracy80%.

First Author
Nama : Nurhayati
Program Studi : Teknik Informatika
STMIK Kaputama
Email: nurhayati_azura@yahoo.co.id

1. PENDAHULUAN

Di era industri 4.0, internet menjadi bagian penting dalam kehidupan manusia. Internet kini terhubung ke segala segi kehidupan manusia atau dikenal dengan istilah internet of things sehinggahal ini memudahkan manusia untuk mengakses kebutuhan mereka kapan pun dan dimanapun tanpa mengenal batas jarak dan waktu. Kini untuk mengetahui berita terbaru cukup dengan mengakses internet. Segala informasi menyebar dengan mudah dan cepat dan juga dengan begitu mudahnya menjadi viral, walaupun keaslian berita itu belum dapat dipertanggungjawabkan alias hoax. Beberapa informasi hoax disebabkan oleh perseorangan dan beberapadisebabkan oleh organisasi yang mengkhususkan dirinya dalam bidang pembuatanberita dan informasi hoax kemudian menyebarkannya pada masyarakat luas.Pemerintah Indonesia telahmengatur sanksi bagi pelaku berita hoax dalamPasal 28 ayat 1 Undang-Undang No. 11 Tahun 2008 tentangInformasi danTransaksi Elektronik atau Undang-Undang ITE. Dalam pasaltersebut dituliskan bahwa "Setiap orang yang dengan sengaja dan atau tanpa hakmenyebarkan berita bohong dan menyesatkan, ancamannya bisa terkena pidanamaksimal enam tahun dan denda maksimal Rp 1 miliar". Namun hal ini tidak juga menghentikan aksi pelaku berita hoax. Seperti yang telah dilansir oleh situs web CNN Indonesia bahwa data yangdipaparkan oleh Kementerian Komunikasi dan Informatika menyebut adasebanyak 800 ribu situs di Indonesia yang terindikasi sebagai penyebar beritapalsu dan ujaran kebencian (hate speech). Istilah Hoax biasanya disebut sebagai"virus

2. TEORITIS

2.1 Hoax

hoaks merupakan berita palsu yang mengandung informasi yang sengaja menyesatkan orang dan memiliki agenda politik tertentu. Hoaks adalah berita yang misleading alias menyesatkan, informasi dalam berita hoaks juga tidak memiliki landasan faktual, namun disajikan seolah-oleh sebagai serangkaian fakta[2,5]. Setiap tahunnya perkembangan berita hoaks meningkat fantastis. Seperti dilansir dari situs website Kementerian.

2.2 Algoritma

Merupakan kumpulan perintah yang saling berkaiatan untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir[3]. Dalam penyusunannya diperlukan urutan serta logika agar algoritma yang dihasilkan sesuai dengan yang diharapkan. Algoritma merupakan bagian yang terpenting dan tidak dapat dipisahkan dari pemograman. Oleh karena itu, sebelum membuat suatu program aplikasi, hal pertama yang harus kita pahami adalah algoritma atau prosedur pemecahannya. Hal ini bertujuan agar program yang telah dibuat dapat sesuai dengan yang diharapkan. Algoritma merupakan hasil pemikiran konseptual. Agar dapat dimengerti oleh komputer, algoritma harus ditranslasikan ke dalam notasi bahasa pemograman.

2.3 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. NLP merupakan cabang ilmu kecerdasan buatan yang dikhususkan untuk mengolah pemrosesan linguistik. Bahasa alami manusia 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.

2.3.1. Pembobotan Kata (*Term Weighting* TF-IDF)

Hal yang perlu diperhatikan dalam pencarian informasi dari koleksi dokumen yang heterogen adalah pembobotan term[4]. *Term* dapat berupa kata, frase atau unit hasil *indexing* lainnya dalam suatu dokumen yang dapat digunakan untuk mengetahui konteks dari dokumen tersebut, maka untuk setiap kata tersebut diberikan indikator, yaitu *term weight[6]*. Rumus umum untuk *Term Weighting* TF-IDF adalah penggabungan dari formula perhitungan raw TF dengan formula IDF dengan cara mengalikan nilai TF dengan nilai IDF:

$$tfidf_{t,d} = tf_{t,d} \cdot idf_t \quad \text{.....(2.1)}$$

Keterangan :

$tfidf_{t,d}$ = bobot *term*
$tf_{t,d}$ = *term frequency* kata t pada dokumen d
idf_t = *inverse document frequency* kata t

Berapapun besarnya nilai $tf_{t,d}$, apabila **D = dft**, maka akan didapatkan hasil 0 (nol), dikarenakan hasil dari log 1, untuk perhitungan **IDF**. Untuk itu dapat ditambahkan nilai 1 pada sisi **IDF**, sehingga perhitungan bobotnya menjadi sebagai berikut:

$$tfidf_{t,d} = tf_{t,d} \cdot idf_t + 1 \quad \text{.....(2.2)}$$

2.3.2. *Inverse Document Frequency* (IDF)

IDF (*Inverse Document Frequency*) merupakan sebuah perhitungan dari bagaimana term didistribusikan secara luas pada koleksi dokumen yang bersangkutan. IDF menunjukkan hubungan ketersediaan sebuah *term* dalam seluruh dokumen. Semakin sedikit jumlah dokumen yang mengandung term yang dimaksud, maka nilai IDF semakin besar. Rumus IDF adalah :

$$idf_t = \log \frac{D}{df_t} \quad \text{.....(2.4)}$$

Keterangan :
idf_t = *Inverse Document Frequency*
D = Jumlah Keseluruhan Dokumen
df_t = Jumlah dokumen yang memuat *term* t

2.4. Algoritma *Levenshtein Distance*

Levenshtein Distance adalah sebuah algoritma yang menggunakan matriks untuk mengukur angka perbedaan antara 2 string[3]. Jarak antara string diukur berdasarkan angka penambahan karakter (*insertion)*,

penghapusan karakter *(deletion)*ataupun penggantian karakter (*substitusion*) yang diperlukan untuk mengubah string sumber menjadi string target (Ishak dkk., 2012). Berikut rumus matriks dari *Levenshtein Distance*:

$$\text{Iev a,b(i,j)} = \begin{cases} \max(i,j)\ if\ \min(i,j) = 0 \\ min \begin{cases} lev\ a,b(1-j)+1 \\ lev,a,b\ (1,j-1)+1 \\ lev\ a,b(i-1,j-1)+1(ai\ \neq bj) \end{cases} \end{cases}$$

Keterangan:
lev a,b : matriks *levensteindistance*
i : baris matriks
j : kolom matriks.

Dalam metode ini memiliki aturan penilaian yang akan dijelaskan dalam contoh sebagai berikut:
1. 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 terhubung sama satu sama lain[6].
2. 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.
3. 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:

$$Similarity = \left\{1 - \frac{edit\ distance}{maxLength\ h\ (stra,strb)}\right\} \text{.....(2.6)}$$

Keterangan:
edit distance : hasil dari *Levenstein Distance.*
maxLength : jumlah *string* dari kata yang terpanjang antara *stra* dan *strb*.
stra : panjang *string* pertama.
strb : panjang *string* kedua.
Similarity : 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.

2.5. 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.

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

$$Precision = \frac{\#(\text{dokumen hoax terklasifikasi hoax})}{\#(\text{jumlah dokumen terklasifikasi hoax})} \text{.....(2.7)}$$

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

$$Recall = \frac{\#(\text{dokumen hoax terklasifikasi hoax})}{\#(\text{jumlah dokumen hoax yang diuji})} \text{.....(2.8)}$$

Penjelelasan mengenai *Precission* dan *Recall* dijelaskan melalui tabel berikut :

Tabel II.1 *Precission* dan *Recall*

	Relevan	Tidak Relevan
Diambil	*True Positive* (tp)	*False Positive* (fp)
Tidak Diambil	*False Negative* (fn)	*True Negative* (tn)

Berdasarkan Tabel diatas, dapat dituliskan rumus sebagai berikut untuk menghitung akurasi sebuah sistem menggunakan perhitungan *Precision and Recall*

P = tp / (tp + fp)
R = tp / (tp + fn) ..... (2.9)

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 :

$$ac = \frac{\sum match}{\sum tp} * 100\% \quad \text{.....(2.10)}$$

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

2.6. PHP (*Hypertext Preprocessor*)

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*

3. ANALISA DAN HASIL

3.1. Tokenizing

contoh kasus penulis menggunakan potongan berita *hoax* yang penulis kutip dari website turnbackhoax.id[7]. Berikut sample berita *hoax* yang digunakan :

Seorang bayi telah dilahirkan di Israel yang mukanya mirip seperti Dajjal,Allah (s.w.t) telah berkata di dalam Al-Qur'an bahwa seorang bayi yang kelihatan seperti Dajjal akan dilahirkan di Israel dan itu adalah petanda hari perhitungan. Berikut adalah perbandingan sebelum dan sesudah tahap *tokenizing* adalah sebagai berikut :

Tabel III.1 Proses Sebelum dan Sesudah *Tokenizing*

Sebelum *Tokenizing*	**Sesudah *Tokenizing***
Seorang bayi telah dilahirkan di Israel yang mukanya mirip seperti Dajjal,Allah (s.w.t) telah berkata di dalam Al-Qur'an bahwa seorang bayi yang kelihatan seperti Dajjal akan dilahirkan di Israel dan itu adalah petanda hari perhitungan.	seorang bayi telah dilahirkan di israel yang mukanya mirip seperti dajjal allah swt telah berkata di dalam alquran bahwa seorang bayi yang kelihatan seperti dajjal akan dilahirkan di israel dan itu adalah pertanda hari perhitungan

3.2. Stopword Removal atau *Filtering*

Tabel III.2 Proses Sebelum dan Sesudah *Stop Removal/Filtering* Teks

Sebelum *Filtering*	**Sesudah *Filtering***
seorang bayi telah dilahirkan israel yang mukanya mirip seperti dajjal allah swt telah berkata di dalam alquran bahwa seorang bayi yang kelihatan seperti dajjal akan dilahirkan di israel dan itu adalah petanda hari perhitungan.	seorang bayi -dilahirkan israel - mukanya mirip –dajjal allah swt berkata - - alquran seorang bayi - kelihatan - dajjal- dilahirkan -israel --- petanda hari perhitungan.

3.3. *Stemming* Nazief & Andriani

Tabel III.3 Proses Sebelum dan Sesudah *Stemming*

Sebelum *Stemming*	Sesudah *Stemming*
bayi	bayi
dilahirkan	lahir
israel	israel
mukanya	muka
mirip	mirip
dajjal	dajjal
allah	allah
swt	swt
berkata	kata
alquran	alquran
seorang	orang
bayi	bayi
kelihatan	lihat
dajjal	dajjal
dilahirkan	lahir
israel	israel
petanda	tanda
hari	hari
perhitungan	hitung

3.4. ***Sorting***

Sorting teks digunakan untuk mengurutkan kata hasil dari *stemming* secara *ascending* atau menaik, sehingga pencocokan *string* dokumen dilakukan pada sata yang sudah terurut.

Tabel III.4 Proses Sebelum dan Sesudah *Sorting*

Sebelum *Sorting*	Sesudah *Sorting*
orang	Allah
bayi	alquran
lahir	bayi
israel	bayi
muka	dajjal
mirip	dajjal
dajjal	hari
allah	hitung
swt	israel
kata	israel
alquran	kata
orang	lahir
bayi	lahir
lihat	lihat
dajjal	mirip
lahir	muka
israel	orang
tanda	orang
hari	swt
hitung	tanda

3.5. Pembobotan Kata Algoritma TF-IDF

Setelah tahap teks *preprocessing* selesai,tahap selanjutnya adalah pembobotan kata (*term*). Dalam pembobotan kata (*term*) ini,setiap kata yang telah melewati proses *preprocessing*, akan di-*parsing* (diuraikan) terlebih dahuludan disimpan dalam *database*, kemudian dihitung jumlah kemunculan setiap katanya. Setelah berita yang diinput dilakukan *prepoessing* dan menjadi data kata hoax, langkah selanjutnya adalah dengan membandingkan data tesebut dengan data latih. Kemudian dilakukan perhitungan untukmengetahui bobot perkata dengan menghitungjumlah *term frequency* dokumen (tf) terlebihdahulu, kemudian menghitung nilai jumlahdokumen yang memiliki *term* (df), danselanjutnya menghitung nilai idf. Setelah nilai TF dan IDF sudah didapat, maka langkah terakhiradalah menentukan bobot kata denganmengalikan TF dan IDF. Hasil dari prosesperhitungan ini disimpan dalam *database* danakan dilanjutkan dengan tahap berikutnyauntuk dilakukan perhitungan menggunakan algoritma *Levenshten Distance* yang merupakan tahap akhir proses.

Tabel III. 5 Koleksi Dokumen *Hoax*

Dokumen 1 (d1)	Badan Meteorologi dan Geofisia menyatakan bahwa akan terjadi kemarau panjang yang akan melanda dunia. Diperkirakan kemarau panjang tersebut akan dimulai tahun 2019 hingga 2022. Cadangan air dunia saat ini hanya tersisa 3% saja. Lalu apa artinya informasi ini bagi kita?Artinya adalah

	kemunculan Dajjal telah sangat dekat. Dan munculnya Imam Mahdi telah berada di tengah-tengah kita tanpa kita sadari. Ini berarti apa yang disabdakan Rasulullah telah terbukti.
Dokumen 2 (d2)	Seorang pria di Bojonegoro meninggal akibat menggunakan Hp.Dia mendengar musik pakai Handset, saat Hpnya di Charge dan ketiduran.Arus Listrik masuk melalui telinga sampai ke sekujur tubuhnya. Kulitnya meletup letup sampai membentuk lubang2 di seluruh tubuh nya…Sekali lagii, jangan pernah gunakan Hp saat sedang di charge sepenting apapun keadaannya karena ini bisa mengancam keselamatan.
Dokumen 3 (d3)	Anak perempuan mertua saya (umur 31 thn) baru meninggal semalam disebabkan oleh leukimia. Almarhum semasa hidupnya meneliti utk gelar Master in botanical di kampus USM mengkaji sejenis tumbuhan ini. Rekan satu tim research beliau sdh meninggal setahun yg lalu mengidap penyakit leukimia juga. Almarhum dpt bertahan hingga semalam. Memang hsl penelitiannya telah disahkan oleh pihak kampus USM dan pihak Kementrian Kesehatan bhw leukimia itu dpt ditimbulkan dari tumbuhan tsb. Jadi cerita D Hizzad Bole penyebab kanker darah (leukimia) ternyata terbukti. Di Cina sdh cukup lama diketahui tentang bahaya tanaman ini, tapi hanya diberitakan suratkabar Cina saja. Sedangkan ditempat kita tanaman ini dirawat & dijadikan tanaman hias di rumah.. PERHATIAN & WASPADA.Jika ada tanaman ini dirumah, silahkan secepatnya musnah dengan membakarnya sebelum tanaman berbunga. Karena dari bunganya menyebabkan kanker darah (leukimia)
Dokumen 4 (d4)	Teman2 yb, sekedar menginformasikan saja, karena fenomena Equinox yang akan mempengaruhi Malaysia, Singapura dan Indonesia di 5 hr ke depan. Disarankan utk tinggal di dalam rumah atau diruang kerja terutama dari jam 12:00-15:00 setiap hari. Suhu akan berfluktuasi sampai 40 derajat Celcius. Hal ini dapat dengan mudah menyebabkan dehidrasi dan matahari stroke. (Fenomena ini adalah karena matahari diposisikan tepat di atas garis khatulistiwa pd tgl 20 Maret. Harap menjaga kesehatan diri agar tdk dehidrasi. Setiap orang setidaknya mengkonsumsi sekitar 3 liter cairan setiap hari. Memonitor tekanan darah. Kemungkinan mendapatkan serangan panas.Mandi air dingin sesering mungkin. Mengurangi daging, perbanyak buah2an & sayuran. Tempatkan lilin tidak terpakai di luar rumah. Jika lilin bisa meleleh, berarti udara dalam tingkat yang cukup berbahaya.Jika bisa selalu menempatkan ember dgn air setengah penuh

	di ruangan, ruang tamu & di setiap kamar untuk menjaga suhu tetap lembab. Pengalaman pertama di Malaysia dan Singapura. Heat stroke yang tidak memiliki gejala indikasi. Setelah pingsan, yang serius berbahaya seperti kegagalan organ dalam.Hari menjadi lebih hangat mungkin lebih dari 2 minggu ke depan. Bisa sampai 9 derajat lebih tinggi dari biasanya ! Terima kasih.
Dokumen 5 (d5)	Susanti, warga Sulawesi Selatan didenda 700rb karena telah melahirkan di rumah. Ia yang tidak memiliki uang lebih memilih melahirkan dirumah dengan jasa dukun beranak. Namun pihak puskesmas setempat mengatakan bahwa setiap kelahiran harus dilakukan di puskesmas, dimana uangnya untuk menggaji para staf. Peraturan aneh ini pun memaksa Susanti meminjam uang ke tetangga untuk membayar denda.

Dokumen data latih diatas dilakukan proses *Text Preprocessing.* Sehingga menjadi :

Tabel III.6 Hasil *Preprocessing*Data Latih

d1	d2	d3	d4	d5
ada	akibat	almarhum	ada	anak
air	ancam	anak	air	aneh
arti	apa	bahaya	air	atur
badan	arus	bakar	alami	bahwa
bagi	bisa	baru	atas	bayar
bukti	bojonegoro	beliau	bahaya	denda
cadang	charge	belum	banyak	denda
dajjal	charge	berita	bisa	dukun
dekat	dengar	bole	buah	gaji
dunia	guna	botanical	cair	harus
geofisika	guna	kampus	celcius	jasa
hingga	handset	bukti	cukup	kata
imam	hp	cepat	hari	lahir
informasi	hp	cerita	dapat	lahir
jadi	jangan	cina	darah	lahir
kemarau	keadaan	darah	mungk	lebih
kira	bujur	gelar	in	milik
landa	kulit	hias	dehidr	paksa
mahdi	lalu	hidup	asi	pihak
meteorologi	letup	hingga		puskes
mulai	listrik	hizzad		mas
muncul	lubang	hsl		pilih
nyata	masuk	idap		
panjang	musik	jadi		
rasulullah	orang	jenis		
saat	pakai	kabar		
sabda	penting	kaji		

Setelah dilakukan pra-pemrosesan tekspada masing-masing dokumen dalam koleksi, maka selanjutnya dilakukan perhitungan *term weightening* TF-IDF, sebagai berikut :

Tabel III.7 Hasil Perhitungan TF-IDF

Q	tf					df	$\frac{D}{df}$	IDF (Log(D/df))	IDF +1	W(TF.IDF) = tf * (IDF+1)				
	d1	d2	d3	d4	d5					d1	d2	d3	d4	d5
allah	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alquran	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bayi	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dajjal	1/35 = 0,0285	0	0	0	0	1	5	0,7	1,7	0,04845	0	0	0	0
hari	0	0	0	3/113 = 0,0265	0	3	1,6	0,2	1,2	0	0	0	0,0318	0
hitung	0	0	0	0	0	0	0	0	0	0	0	0	0	0
israel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
kata	0	0	0	0	1/34 = 0,03	1	5	0,7	1,7	0	0	0	0	0,051
lahir	0	0	0	0	3/34 = 0,0882	3	1,6	0,2	1,2	0	0	0	0	0,10584
lihat	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mirip	0	0	0	0	0	0	0	0	0	0	0	0	0	0
muka	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Jumlah *string* d1 = 35, d2 = 34, d3=70, d4=113, d5=34

Jika kata *hoax*terdapat dalam *list* berita maka nilai hitung akan lebih besar dari 0, sehingga proses menambahkan 1 pada perhitungan dokumen.

3.6. **Perhitungan *Levenshtein Distance***

untuk menghitung jarak antara 2 kata (*string*) yang dibandingkan dan juga untuk mengukur kesamaan antara 2 kata yaitu kata sumber (a) dan kata target (b). Kata sumber merupakan input berita yang dimasukkan user (data uji), sedangkan kata target merupakan database kata hoks (data latih). penyeleksian panjang kedua *string* terlebih dahulu. Jika salah satu atau kedua *string* merupakan *string* kosong, jalannya algoritma ini berhenti dan memberikan hasil *edit distance* bernilai 0 (nol). Apabila *string* yang dibandingkan panjangnya tidak bernilai 0, maka dilakukan perhitungan. Misal *string* a adalah "LAHIR" dan *string* b adalah "ORANG", Jika dilihat secara sekilas, kedua *string* tersebut memiliki jarak 5. Berarti untuk mengubah *string* "LAHIR" menjadi "ORANG" diperlukan 5 operasi, yaitu : Dengan menggunakan representasi matriks dapat dilihat pada tabel dibawah.

Tabel III.8 Matriks Perhitungan Levenshtein Distance

L	A	H	I	R
O	R	A	N	G

Berdasarkan tabel diatas kita telah melakukan perbandingan antara string a "LAHIR' yang memilki panjang 5 dengan string b "ORANG" memilki panjang 5. Dari hasil perbandingan maka didapatilah nilai jarak perbedaan antara 2 string diatas adalah 5 yang diambil dari nilai matriks pada ujung kanan bawah matriks. Jika panjang *string* keduanya tidak nol, berarti setiap sekuen memilki sebuah karakter. Sehingga perhitungan yang dilakukan dengan mentransformasikan *string* a (kata sumber) menjadi b (kata target). Jika *string* a sama dengan *string* b, maka nilai cost sama dengan 0. Misalnya pada *String* a (i1) diawali karakter "D" dan karakter *String* b (j1) diawali dengan karakter "D", cost sama dengan 0 karena tidak ada perubahan apapun yang dilakukan oleh sistem. Jika *string* a berbeda dengan *string* b, maka nilai cost-nya 1 karena membutuhkan 1x operasi perubahan dari *string* a menjadi *string* a. Sehingga, nilai edit distance-nya dari pentransformasian sekuen pertama menjadi sekuen kedua ditambah 1. Pada matriks diatas, saat *string* b karakter "O" dan *string* a karakter "L" nilai cost sama dengan 1, karena membutuhkan 1 operasi yaitu *Subtitusi* atau merubah L menjadi O.Pada

		L	**A**	**H**	**I**	**R**
	0	**1**	**2**	**3**	**4**	**5**
O	**1**	1	2	3	4	5
R	**2**	2	2	3	4	5
A	**3**	3	3	3	4	5
N	**4**	4	4	4	4	5
G	**5**	5	5	5	5	**5**

perbandingan karakter selanjutnya, nilai cost pertama mempengaruhi nilai cost selanjutnya. Nilai cost terkecil merupakan nilai cost yang digunakan.

3.7. Perhitungan Nilai *Similiarity*

Setelah mendapatkan nilai *edit-distance,* langkah selanjutnya adalah menghitung nilai *similiarity*. Jika nilai similarity adalah 1, maka kedua *string* yang dibandingkan sama. Di lain hal, jika similarity 0, maka kedua *string* yang dibandingkan tidak sama.

$$Similarity = \left\{1 - \frac{edit\ distance}{maxLength\ (stra,strb)}\right\}$$

$$Similarity = \left\{1 - \frac{5}{maxLength\ (5,5)}\right\}$$

$$Similarity = \left\{1 - \frac{5}{5}\right\}$$

$$Similarity = \{1 - 1\}$$

$$Similarity = 0$$

Nilai kesamaan (*similarity*) kata tersebut dikalikan dengan nilai bobot (TF-IDF) dari kata target yang dibandingkan yaitu ‘ORANG’ dengan bobot kata 0,042 dan 0,01232. Perhitungan sebagai berikut :

Tfidfmix = $Similarity$ * bt
Tfidfmix = 0 * 0,042 = 0
Tfidfmix = 0 * 0,01232= 0

Selanjutnya perkalian *similiarity* dan TF-IDF akan dijumlahkan dengan hasil perkalian yang sama namun dengan nilai *similiarity*dan TF-IDF yang berbeda. Kemudian kata target akan disimpan di data target. Hal ini bertujuan untuk menghitung banyaknya kata target yang memiliki nilai jarak minimal yang sama terhadap satu kata sumber. Perhitungan selanjutnya adalah menghitung rata-rata dari keseluruhan hasil perkalian *similiarity* dan TF-IDF dan didapatilah nilai dalam %.

4. IMPLEMENTASI

Setelah sistem sudah terbentuk, maka selanjutnya dilakukan pengujian pada sistem. Teks berita yang sudah diuji akan menghasilkan nilai, jika nilai yang dihasilkan besar maka semakin besar kemungkinan teks berita tersebut mengandung unsur *hoax* dan sebaliknya. 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. 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.

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

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

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

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

Gagasan tersebut dapat diperjelas melalui tabel berikut.

Tabel III.9 *Precission* dan *Recall* Berdasarkan Data Pengujian

Aktual / Prediksi	Dokumen *Hoax*	Dokumen*Non-Hoax*
Terklasifikasi *Hoax*	*True positive* (tp)	*False positive* (fp)
Terklasifikasi *Non-Hoax*	*False negative* (fn)	*True negative* (tn)

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

Kemudian berikut adalah rumus akurasinya :

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

Gambar IV.1 *Form Home User*

Gambar IV.8*Form Input* Berita

4. KESIMPULAN

Terdapat langkah-langka untuk menerapkan algoritma *Levenshtein Distance* dalam sistem pendeteksi berita *hoax*, yaitu :

a. Pembuatan Dokumen Data Target yang di dalamnya terdapat kumpulan kata *hoax* yang sudah disederhanakan dalam Prapemrosesan Kata (*Prepocessing Text*) dan Penyeleksian kata dengan memberi bobotpada setiap kata menggunakan TF-IDF.
b. Pembuatan Sistem Deteksi *Hoax* yang di dalamnya terdapat beberapa proses hingga menghasilkan nilai klasifikasi yaitu Prapemrosesankata sumber, membandingkan kata sumber dan katatarget, menghitung jarak (*Levenshtein Distance*), memberi bobot(TF-IDF), dan menghitung hasil akhir sekaligus pengklasifikasian.

Sistem yang dibangun dapat melakukan penerapan algoritma TF-IDF dan algoritma levenshtein distance yang mendeteksi berita hoaks dan menghasilkan nilai keakurasian hasil deteksi berita.

Kuantitas data latih, keakuratan data kata dasar, dan *stopword* mempengaruhi keakuratan hasil deteksi berita. Penggunaan proses *stemming* akan lebih mampu mendeteksi berita jika dalam *database* memiliki kata dasar yang umum terdapat dalam berita hoaks sehingga tidak menghasilkan kerancuan dalam kata dasar yang dihasilkan. Batas 0,1dengan data uji 40 dokumen denganpembagian 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 melakukanpendeteksian

.

REFERENSI

[1] [1] 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

[2] [2] Silverman, Craig. ‚Lies, Damn Lies and Viral Content.' Columbia Journalism Review, 2015, 1–149. https://doi.org/10.7916/D8Q81RHH.

[3] [3] Weddiningrum, Frista Gifti. Deteksi Konten Hoax Berbahasa Indonesia Pada Media Social Menggunakan Metode Levenshtein Distance, Skripsi, Surabaya [ID] : Universitas Negeri Sunan Ampel Surabaya.

[4] [4] Ryansyah Adi dan Sri Andayani. Implementasi Algoritma TF-IDF pada Pengukuran Kesamaan Dokumen, Jurnal Sistem dan Teknologi Informasi Komunikasi 1(1) : 2, diakses tanggal 18 Juli 2019.

[5] [5] Rahadi Dedi, Rianto. 2017. Perilaku Pengguna Dan Informasi Hoax Di Media Sosial, Jurnal Manajemen dan Kewirausahaan 5(1) : 61:62,jurnal.unmer.ac.id/index.php/jmdk/article/download/1342/933,diakses tangggal 24 Mei 2019.

[6] [6] Nangili, Supandi, dkk. 2014. Pengujian Algoritma Levenshtein Distance dan Algoritma Term Frequency Inverse Document Frequency (TF-IDF) untuk penilaian jawaban essay, Karya Ilmiah, Gorontalo [ID] : Universitas Gorontalo.

[7] [7] Omar, Braddley Muhammad,dkk. Pengoreksian Ejaan Kata Berbahasa Indonesia Menggunakan Algoritma Levenshtein Distance, Prosiding Annual Research Seminar 2017 3(1): 169 :170, diakses tanggal 15 Agustus 2019..