Penerapan Algoritma TextRank dan Dice Similarity Untuk Verifikasi Berita Hoax

Christian Khontoro, Justinus Andjarwirawan, Yulia
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-Mail: christiankho72@gmail.com, justinus@petra.ac.id, yulia@petra.ac.id

ABSTRAK

Hoax atau dalam Bahasa Indonesia, hoaks adalah berita bohong atau berita yang tak punya sumber. *Hoax* merupakan serangkaian informasi yang dibuat sesat, tetapi dijual sebagai kebenaran[5]. Masalah ini menjadi dasar untuk pembuatan sistem verifikasi berita *hoax* ini. Algoritma *TextRank* dan *Dice Similarity* akan digunakan untuk membantu memverifikasi berita yang diinputkan itu hoax atau fakta. Dimana pada penelitian ini, algoritma TextRank digunakan untuk mencari Keyword terpenting di satu berita yang kemudian akan digunakan untuk menjadi *Keyword* di mesin pencari. Kemudian algoritma *Dice Similarity* digunakan untuk mengukur tingkat kemiripan berita yang diinputkan dengan berita yang didapatkan dari hasil search di mesin pencari. Sistem verifikasi *hoax* yang telah dilakukan ini telah diuji menggunakan beberapa bobot *similarity* untuk mencari bobot *similarity* mana yang paling optimal. Data yang digunakan sebanyak 50 berita *hoax* dan 50 berita fakta. Dari pengujian tersebut didapatkan bobot *similarity* optimal sebesar 40% dengan akurasi mencapai 84%. Dengan rincian dari 50 data *hoax* didapatkan 47 berita dinyatakan hoax, 2 berita dinyatakan fakta, dan 1 berita dinyatakan *unknown*. Dari 50 berita fakta didapatkan 37 berita yang dinyatakan fakta, 13 berita dinyatakan *hoax*, dan tidak ada berita dinyatakan *unknown*.

Kata Kunci: verifikasi hoax, *dice similarity*, *textrank*, *keyword extraction*, *web scraping*.

ABSTRACT

Hoax or in Indonesian, hoax is fake news or news that has no source. Hoax are a series of information that is misguided, but sold as truth[5]. The problems above are the basis for creating a verification system for this hoax news. The TextRank and Dice Similarity algorithms will be used to help verify the inputed news is a hoax or fact. Where in this study, the TextRank algorithm is used to find the most important keywords in a news which will then be used to become keywords in search engines. Then the Dice Similarity algorithm is used to measure the level of similarity of the news entered with the news obtained from search results on search engines. The hoax verification system that has been done has been tested using several similarity weights to find which similarity weights are the most optimal. The data used were 50 hoax news and 50 fact news. From this test, the optimal similarity weight is 40% with an accuracy of 84%. With details of 50 hoax data, 47 news were declared hoax, 2 news items were declared facts, and 1 news was declared unknown. Of the 50 fact news, 37 news were declared facts, 13 were declared hoax, and no news was declared unknown.

Keywords: *hoax verification*, *dice similarity*, *keyword extraction*, *textrank*, *web scraping*.

1. PENDAHULUAN

Sekarang ini, hampir setiap orang memiliki kapasitas untuk menyebarkan informasi, baik melalui media sosial ataupun platform lain. Informasi seharusnya mengandung berita yang valid, karena informasi bisa mempengaruhi emosi, perasaan, pikiran, dan tindakan sebuah individu. Jika informasi tersebut mengandung berita yang salah atau bahkan berita yang bersifat provokatif maka hal ini akan menggiring opini pembaca menjadi negatif.

Hoax atau dalam Bahasa Indonesia, hoaks adalah berita bohong atau berita yang tak punya sumber. Hoax merupakan serangkaian informasi yang dibuat sesat, tetapi dijual sebagai kebenaran [5]. Penyebaran hoax bergantung pada pembaca yang dengan sengaja mengirimkan pesan atau berita tersebut ke korban potensial lainnya. Psikolog meyakini, berita hoax dihadirkan untuk memanipulasi banyak orang. Sebab, berita palsu bisa memanfaatkan kelompok orang yang takut, dan mengambil keuntungan ketakutan itu. Jangan menyepelekan dampak buruk berita hoax pada kesehatan mental. Sebab, efeknya bisa berlangsung dalam jangka panjang. Misalnya, mengganggu situasi emosional dan suasana hati yang berkepanjangan, sampai "menghantui" pikiran untuk waktu yang lama [9]. Menurut Staf Ahli Menteri Bidang Hukum Kementerian Komunikasi dan Informatika Henry Subiakto, berita Hoax mempunyai ciri yaitu sumber informasi atau medianya tidak jelas identitasnya, mengeksploitasi fanatisme SARA, tidak menggunakan 5W+1H (What, When, Who, Why, Where, How), dan juga berita Hoax cenderung meminta pembacanya untuk menyebarkannya semasif mungkin [2].

Masalah diatas menjadi dasar untuk pembuatan sistem verifikasi berita hoax ini. Algoritma TextRank dan Dice Similarity akan digunakan untuk memverifikasi berita yang diinputkan itu hoax atau fakta. Apabila berita inputan berasal dari media whitelist, tetapi judul berita yang dipublish oleh media whitelist tersebut mengindikasikan hoaks, maka berita inputan tersebut berpotensi hoaks.

Sistem verifikasi berita hoax ini pernah dilakukan oleh Sucipto & Indiarti R.[7]. Mereka melakukan verifikasi berita Hoax dengan menggunakan Text MIning Classification System. Penelitian dari Sucipto & Indiarti R. menunjukkan hasil dimana dari data konten berita sebanyak 20 isu yang terdiri dari 10 isu benar dan 10 isu salah. Kemudian sistem menghasilkan klasifikasi dengan rincian 13 isu termasuk salah dan 7 isu termasuk benar, maka jumlah

klasifikasi yang sesuai label aslinya sebanyak 15 isu. Berdasarkan hasil klasifikasi tersebut, didapatkan nilai akurasi sebesar 75% [7].

Penelitian Sucipto & Indiarti R. menggunakan metode TextRank & Cosine Similarity. Dimana pada penelitian ini akan menggunakan metode TextRank & Dice Similarity karena berdasarkan penelitian dari Fikri A.D[1]. Algoritma Dice Similarity memiliki keunggulan dalam memberikan Feedback dokumen yang relevan dengan kata kunci masukan(Recall) dibandingkan metode Cosine Similarity[1] .

Pada penelitian ini, akan digabungkan algoritma TextRank dengan Dice Similarity. Dimana algoritma TextRank digunakan untuk mencari Keyword terpenting di satu berita yang kemudian akan digunakan untuk menjadi Keyword di mesin pencari. Kemudian algoritma Dice Similarity untuk mengukur tingkat kemiripan berita yang diinputkan dengan berita yang didapatkan dari hasil search di mesin pencari atau biasa disebut dengan istilah Web Scraping.

Penelitian ini bertujuan untuk membantu orang-orang dalam memverifikasi berita yang didapatnya itu hoax atau fakta. Sehingga tidak terjadi maraknya penyebaran berita hoax yang beredar di masyarakat.

Rumusan Masalah

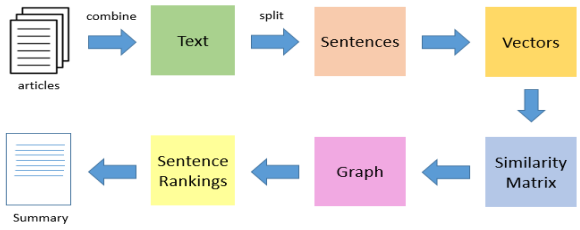
Berdasarkan permasalahan yang ada di latar belakang, maka rumusan masalah yang dapat diimplementasikan:

1. Berapakah persentase kebenaran sistem verifikasi *hoax* menggunakan algoritma *TextRank* dan *Dice Similarity* ini dalam menyatakan suatu berita *Hoax* atau bukan?
2. Berapa tingkat *Similarity* untuk menentukan relevansi berita yang optimal dalam memverifikasi berita *Hoax*?

2. DASAR TEORI

2.1 TextRank

Algoritma TextRank merupakan sebuah algoritma untuk meringkas sebuah text agar kita mendapatkan kalimat-kalimat terpenting di text tersebut. Cara kerja TextRank adalah dengan mencari kalimat yang paling mirip dengan seluruh kalimat yang ada di text. Kalimat yang paling mirip dengan semua kalimat itulah yang akan menjadi kalimat terpenting di teks tersebut. Flow dari algoritma Textrank dapat dilihat pada Gambar 1.[4]



Gambar 1. Flow Algoritma *TextRank*

2.2 Dice Similarity

Dice Similarity yang dikenal juga dengan sebutan *Sørensen–Dice index* merupakan metode yang digunakan untuk membandingkan tingkat similaritas dari dua objek. Metode ini dipublikasikan oleh Sørensen dan Lee Raymond Dice pada 1948 dan 1945[3]. *Dice Similarity* mempunyai Persamaan yang bisa dilihat pada Persamaan 1.

$$\frac{2|D \cap Q|}{|D| + |Q|} \quad (1)$$

Keterangan :
|D n Q| : intersect antara set D dan set Q.
|D| : banyak elemen yang terdapat pada set D
|Q| : banyak elemen yang terdapat pada set Q

2.3 Web Scraping

Web scraping (panen web) adalah pengambilan sebuah dokumen semi-terstruktur dari internet, umumnya berupa halaman-halaman web dalam bahasa markup seperti HTML(*Hypertext Markup Language*) atau XHTML(*eXtensible HyperText Markup Language*), dan menganalisis dokumen tersebut untuk diambil data tertentu dari halaman tersebut. Istilah gampangnya yaitu pengambilan konten atau sebagian data dari suatu situs web.[8]

2.4 Whitelist Web

Whitelist merupakan daftar situs web yang diberikan sebuah hak istimewa untuk membantu memverifikasi berita *Hoax*. Daftar Whitelist ini akan diambil dari website resmi Dewan Pers Indonesia. Dimana isi dari daftar tersebut merupakan media-media cetak maupun online yang telah terverifikasi. Dan daftar media terverifikasi oleh Dewan Pers Indonesia ini akan selalu di perbarui. Contoh daftar situs yang masuk kedalam *Whitelist* seperti detik.com, liputan6.com, kompas.com, dan lain-lain.

2.5 Cekfakta.com

Cekfakta.com merupakan sebuah situs yang menyediakan informasi terkait berita yang beredar di masyarakat itu hoax atau bukan. Cekfakta.com ini dibangun diatas API Yudistira oleh MAFINDO(Masyarakat Anti Fitnah Indonesia) dan bekerja sama dengan media online yang tergabung dalam AJI (Aliasi Jurnalis Independen) serta didukung oleh Google News Initiative dan Internews serta Firstdraft. Data dari cekfakta.com ini akan digunakan sebagai sampel data pengujian pada skripsi ini.

2.6 Confusion Matrix

Confusion Matrix merupakan matrix yang memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi sebenarnya. Pada penelitian ini, *confusion matrix* digunakan untuk mendapatkan hasil akurasi dari klasifikasi hoax yang dilakukan oleh sistem[6]. Persamaan akurasi menggunakan *confusion matrix* dapat dilihat pada Persamaan 2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Keterangan :
TP = Data positif yang diprediksi benar
TN = Data negatif yang diprediksi benar
FP = Data negatif yang diprediksi sebagai data positif
FN = Data positif yang diprediksi sebagai data negative

2.7 Cosine Similarity

Cosine Similarity berfungsi untuk membandingkan kemiripan antar dokumen. Dalam menghitung cosine similarity, pertama yang di lakukan yaitu melakukan perkalian skalar antara query dengan dokumen kemudian dijumlahkan, setelah itu melakukan perkalian antara panjang dokumen dengan panjang query yang telah dikuadratkan, setelah itu di hitung akar pangkat dua. Selanjutnya hasil perkalian skalar tersebut di bagi dengan hasil perkalian panjang dokumen dan query[3]. Persamaan dari *Cosine Similarity* bisa dilihat pada Persamaan 3.

$$cosSim(d_j, q_k) = \frac{\sum_{i=1}^{n}(td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^{n} td_{ij}^2} \times \sqrt{\sum_{i=1}^{n} tq_{ik}^2}} \quad (3)$$

Keterangan :
cosSim(dj,qk) = tingkat kesamaan document dengan query
tdij = term ke-i dalam vector untuk dokumen ke-j
tqik = term ke-i dalam vector untuk dokumen ke-k
n = jumlah term yang unik dalam dataset

2.8 Text Preprocessing

Berdasarkan ketidakaturan suatu teks, maka akan dilakukan proses *text preprocessing*. Dimana *text preprocessing* akan dilakukan dengan beberapa tahap agar teks menjadi lebih terstruktur. Tahapan yang dilakukan adalah :

1. **Case Folding**
   Karena tidak semua dokumen konsisten dalam penggunaan huruf kapital, maka sistem akan menkonversi keseluruhan teks dalam dokumen menjadi huruf kecil.
2. **Tokenizing**
   Tahap *Tokenizing* adalah tahap pemotongan tiap kata dalam dokumen berdasarkan tiap kata yang menyusunnya.
3. **Stopword Remover**
   Tahap *Stopword Remover* merupakan tahap untuk mengambil kata-kata penting dari hasil *tokenizing.* Kata-kata yang tidak memiliki arti atau kata-kata yang kurang penting akan dibuang. Contoh kata yang termasuk dalam *stopword* adalah "yang","di", "dari", dan seterusnya.

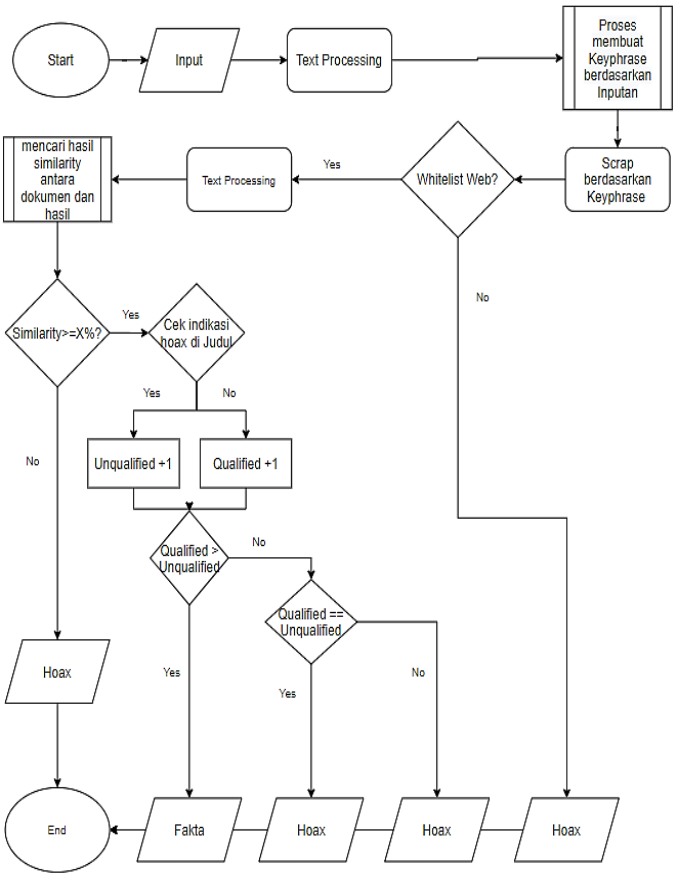
3. DESAIN SISTEM

Desain Implementasi Sistem

Berikut adalah analisis dan perancangan sistem *website* yang telah dibuat.

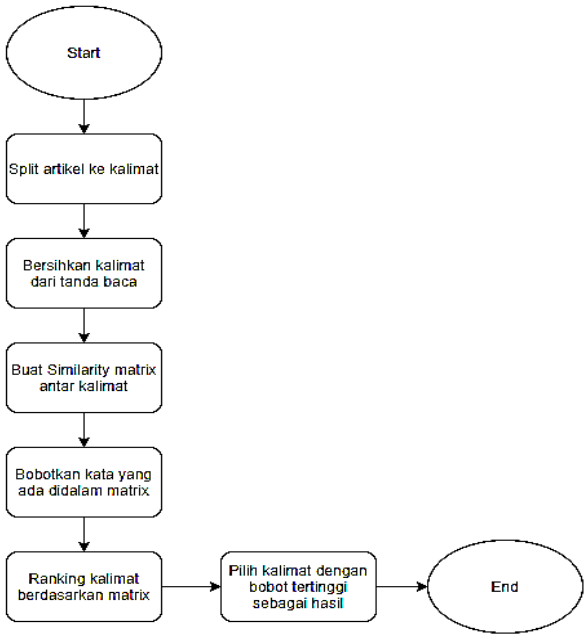
Flowchart Sistem

Berdasarkan Gambar 2, pengguna bisa menginputkan suatu artikel atau berita di dalam input form. Kemudian inputan tersebut akan dicari *keyphrase*nya kemudian sistem akan melakukan pencarian berdasarkan *keyphrase* tersebut di mesin pencari, atau bisa disebut juga dengan *scraping* dan mencari hasil paling mirip. Apabila hasil tersebut linknya berada di dalam whitelist, maka akan dilanjutkan untuk menghitung kemiripan dokumen dan hasil. Jika kemiripan dokumen dan hasil melebihi batas *similarity* yang telah ditentukan, maka judul dari hasil tersebut akan dicek. Apabila judul dari hasil tersebut mengandung kata 'Hoax','Disinformasi','Salah', dan lain-lain, maka variabel unqualified di tambah 1. Jika judul dari hasil tersebut tidak mengandung kata-kata tersebut, maka variabel qualified ditambah 1. Jika hasil dari qualified lebih dari unqualified, maka artikel tersebut adalah fakta. Jika hasil Qualified sama dengan Unqualified, maka artikel tersebut adalah Unknown. Namun, jika hasil Qualified kurang dari Unqualified, maka artikel tersebut adalah Hoax.



Gambar 2. Flowchart Sistem

Proses Membuat Keyphrase Berdasarkan Inputan



Gambar 3. Flowchart *TextRank*

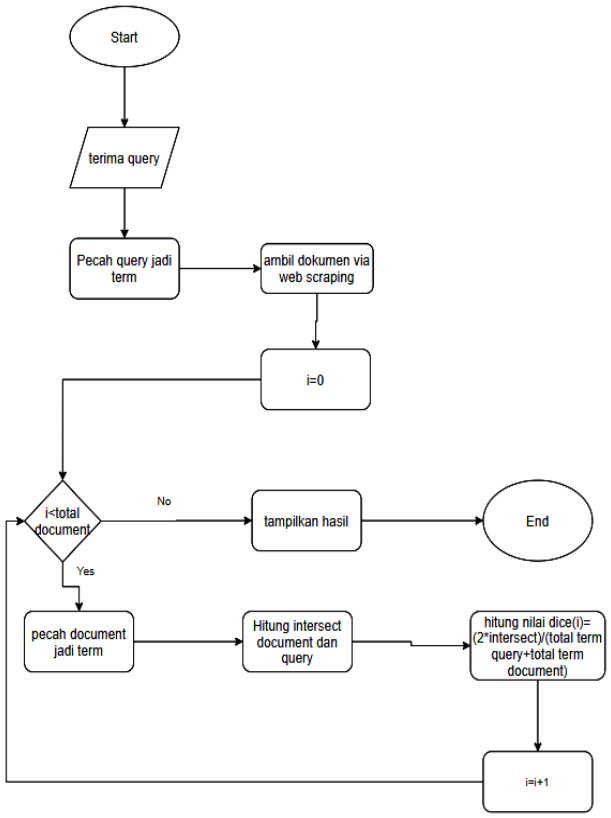
Algoritma dari *TextRank* ini bertujuan untuk mengambil kalimat terpenting dalam suatu artikel. Awalnya, artikel inputan akan dipecah menjadi kalimat. Kemudian setelah dipecah menjadi

kalimat, kalimat tersebut dibersihkan dari tanda baca yang ada. Kalimat-kalimat yang didapatkan ini kemudian akan dimasukkan kedalam sebuah *similarity matrix*. Kalimat dalam matrix ini kemudian akan diberikan bobot di tiap kalimat yang ada. Kalimat-kalimat tersebut akan diranking sesuai dengan bobotnya. Untuk kalimat terpenting, maka akan dipilih kalimat dengan bobot tertinggi.

Proses Mencari Similarity Antara Dokumen dan Hasil

Inputan yang didapatkan akan dipecah menjadi term. Term merupakan tiap kata yang ada didalam sebuah artikel. Kemudian akan dilakukan pencarian dokumen dengan *web scraping*. Inisialisasi variable I sebagai counter, variable I diset dengan angka 0. Jika jumlah dokumen masih melebihi nilai dari variable I, maka dokumen yang didapatkan akan dipecah menjadi term. Lalu akan dihitung intersect antara document, dengan query (inputan).

Kemudian akan dihitung nilai kemiripan antara dokumen dengan query menggunakan *dice similarity* dengan Persamaan 1.Kemudian, nilai counter akan ditambah 1, dan akan dicek lagi didalam loop. Jika nilai counter sudah melebihi total dokumen, maka hasil akan ditampilkan dalam bentuk table, dan akan bisa dilihat oleh pengguna.



Gambar 4. Flowchart *Dice Similarity*

4. PENGUJIAN SISTEM

4.1 Pengujian Akurasi Verifikasi Hoax

Pada tahap ini akan dilakukan pengujian akurasi dari sistem verifikasi hoax yang telah diimplementasikan kedalam *website*. Pengujian ini akan dilakukan dengan beberapa bobot similarity untuk menentukan mana yang paling optimal. Semua pengujian bobot *similarity* dilakukan dengan 50 data hoax dan 50 data fakta yang sama.

Tabel 1. Kesimpulan Akurasi

Bobot	Hasil Akurasi	Pengujian Berita Hoax diverifikasi Fakta(FP)
20%	85%	9
30%	84%	4
40%	85%	2
50%	82%	0
60%	79%	0

Dari Tabel 1, akurasi terbaik diperoleh ketika bobot *similarity* ditetapkan dengan nilai 40% dan 20%, dengan hasil akurasi sebesar 85%. Namun, berita hoax yang diprediksi fakta sangat krusial, karena bisa menyesatkan pengguna. Bobot *similarity* 20% mendapatkan total 9 berita hoax diprediksi fakta, sedangkan bobot *similarity 40%* mendapatkan total 2 berita hoax diprediksi fakta. Maka, disimpulkan bahwa bobot *similarity* optimal adalah sebesar 40%.

4.2 Analisa Tren Berdasarkan Boboty Similarity

Dari pengujian dengan beberapa bobot *similarity* yang telah dilakukan, maka didapatkan tren seperti yang ditampilkan pada Tabel 2.

Tabel 2. Analisa Tren

Bobot	TN	TP	FN	FP	Akurasi
20%	44	41	6	9	85%
30%	46	38	12	4	84%
40%	48	37	13	2	85%
50%	50	32	18	0	82%
60%	50	29	21	0	79%

Berdasarkan analisa pada Tabel 2, Tren menunjukkan kecenderungan apabila bobot *similarity* dinaikkan, akan semakin banyak berita hoax yang diprediksi hoax(TN), dan berita fakta diprediksi fakta(TP) semakin menurun. Namun, berita hoax yang diprediksi fakta(FP) sangat krusial, karena bisa menyesatkan pengguna. Di bobot *similarity* 50% dan 60%, tidak ditemukan berita hoax yang diprediksi fakta(FP). Namun, akurasi yang dimiliki bobot *similarity* 50% dan 60% lebih kecil dibandingan bobot *similarity*.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dalam penerapan algoritma *Textrank* dan *dice similarity* untuk verifikasi berita hoax adalah sebagai berikut:

- Tingkat persentase kebenaran sistem verifikasi hoax menggunakan algoritma *Textrank* dan *dice similarity* didapatkan sebesar 85% dengan bobot *similarity* optimal sebesar 40% dan data berita hoax diprediksi fakta sebesar 2 data.
- Auto update *whitelist* berdasarkan dari website dewanpers.or.id berhasil.
- Apabila bobot *similarity* terlalu tinggi, maka akan sulit untuk memverifikasi berita. Karena berdasarkan pengujian yang telah dilakukan, apabila bobot *similarity* dinaikkan, akurasi sistem dalam memverifikasi berita semakin menurun.

5.2 Saran

Saran untuk penerapan algoritma *Textrank* dan *dice similarity* untuk verifikasi berita hoax adalah sebagai berikut:

- *Website* mungkin bisa disempurnakan menjadi aplikasi mobile, sehingga lebih memudahkan pengguna memverifikasi hoax.
- Mencari metode untuk mendapatkan hasil *scraping* isi konten yang lebih optimal dan tidak memakan banyak waktu.
- Menggunakan metode pengecekan relevansi dokumen yang lebih baik dari *dice similarity*.

6. DAFTAR REFERENSI

[1] Fikri A.D 2019. Perbandingan metode *Dice Similarity* dengan *Cosine Similarity* menggunakan *Query Expansion* pada pencarian *Ayatul Ahkam* dalam terjemah *Alquran* berbahasa Indonesia. Teknik Informatika UIN Malang. 5(4), 3-12. URL = http://etheses.uin-malang.ac.id/13814/1/13650031.pdf

[2] Farisa F.C 2019. Ini Empat Ciri Hoaks Menurut Kominfo. Retrieved May 30, 2020, from https://nasional.kompas.com/read/2019/08/20/14512191/ini-empat-ciri-hoaks-menurut-kominfo

[3] Informatikalogi 2019. Vector Space Model (VSM) dan Pengukuran Jarak pada Information Retrieval (IR). Retrieved May 20, 2020 from https://informatikalogi.com/vector-space-model-pengukuran-jarak/

[4] Joshi 2018. An Introduction to Text Summarization using the TextRank Algorithm. Retrieved May 26, 2020, from https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/

[5] Kiram. 2019, 13 Juni. Yuk, Kenal Lebih Jauh tentang Hoax biar Nggak Kemakan Hoax! Retrieved May 23, 2020, from https://www.quipper.com/id/blog/tips-trick/kenal-lebih-jauh-tentang-hoax/

[6] Nugroho S.K 2019. Confusion Matrix untuk Evaluasi Model pada Supervised Learning. Retrieved December 12, 2020, from https://medium.com/@ksnugroho/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f

[7] Sucipto & Indriati R. 2018. Deteksi Hoax Pada Media Sosial Berbasis Text Mining Classification System. Jurnal Informatika PGRI Kediri, 1(3),1-10. URL = http://simki.unpkediri.ac.id/mahasiswa/file_artikel/2018/14.1.03.03.0052.pdf

[8] Syabab 2019. Apa itu Web Scrapping?. Retrieved Mei 26, 2020, from https://pesonainformatika.com/other-notes/apa-itu-web-scraping/

[9] Wisnubrata 2019. Dampak Buruk Berita Hoax pada Kesehatan Mental, Ini Penjelasannya. Retrieved May 20, 2020, from https://lifestyle.kompas.com/read/2019/10/08/120209420/dampak-buruk-berita-hoax-pada-kesehatan-mental-ini-penjelasannya?page=all