

**KLASIFIKASI BERITA HOAX PILPRES MENGGUNAKAN METODE  
MODIFIED K-NEAREST NEIGHBOR DAN PEMBOBOTAN  
MENGGUNAKAN TF-IDF**

**Faizal Nur Rozi<sup>\*</sup>, Dwi Harini Sulistyawati<sup>\*\*</sup>**

Teknik Informatika, Fakultas Teknik, Universitas 17 Agustus 1945 Surabaya

E-Mail: <sup>\*</sup> faizalnurrozi@gmail.com, <sup>\*\*</sup> dwiharini@untag-sby.ac.id

**ABSTRAK**

Dewasa ini perkembangan teknologi informasi dan internet begitu pesat. Sangat mudahnya masyarakat dalam mengakses internet dan informasi. Dengan kemudahan akses internet membuka peluang masyarakat untuk menyebarkan informasi atau berita di jejaring sosial maupun media online. Namun dalam praktiknya banyak bermunculan berita yang memiliki informasi palsu (hoax) yang tidak dapat dipertanggungjawabkan berita dan sumbernya. Hoax merupakan upaya untuk memanipulasi audiens agar terpengaruh dengan opini yang dibawa. Untuk menangani masalah terkait berita hoax, dikembangkan sebuah prototipe untuk mengklasifikasikan berita tersebut. Metodologi penanganan berita hoax tersebut menggunakan pendekatan Term Frequency – Inverse Document Frequency untuk pembobotan setiap dokumen dan Modified K-Nearest Neighbor untuk pengklasifikasian berita berdasarkan dokumen yang telah dibobotkan.

**Kata Kunci :** Modified K-Nearest Neighbor, TF-IDF, NLP, hoax, Similarity.

**1. Pendahuluan**

Perkembangan teknologi internet saat ini begitu pesat. Sangat mudahnya masyarakat dalam mengakses internet dan informasi. Pada tahun 2017. Hasil perhitungan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) menyimpulkan bahwasanya pengguna internet di Indonesia hingga 143.26 juta jiwa atau sekitar 54.68 persen dari total penduduk Indonesia. Dari hasil perhitungan tersebut terjadi kenaikan sekitar 10.56 juta jiwa pada survei tahun 2016.

Mudahnya akses internet dikalangan masyarakat berpotensi menyebarkan informasi atau berita di jejaring social maupun media online. Pada kondisi lapangan beredar berita yang memiliki informasi yang palsu/*hoax* yang tidak dapat dipertanggungjawabkan. Maksud dari

penyebaran berita *hoax* tersebut adalah untuk meyakinkan pembaca agar terpengaruh oleh narasi yang dibangun pada berita *hoax* tersebut.

Hoax merupakan upaya untuk memanipulasi audiens agar terpengaruh dengan opini yang dibawa. Contoh dari berita *hoax* adalah meyakinkan sebuah kejadian yang sebenarnya tidak sesuai dengan fakta lapangan.

Dosen Universitas Atmajaya Yogyakarta bernama Danarka Sasongko beropini bahwa masyarakat tidak mampu membedakan berita *hoax* dan *non hoax*. Menurutnya hal tersebut terjadi akibat rendahnya narasi publik terhadap informasi di media sosial [1]. Budi Sutejo memaparkan bahwa keahlian pembaca dalam menelaah kemudian menulis kembali narasi yang diterimanya

disebut sebagai literasi media. Pakar *Information Technology* (IT) dari Universitas Kristen Duta Wacana Yogyakarta itu juga menilai bahwa literasi media dapat menangkal persebaran berita hoaks [2].

Dengan berkembangnya teknologi dan ilmu pengetahuan, permasalahan tersebut dapat diakomodir. Cabang ilmu pengetahuan yang mampu mengakomodir konteks permasalahan tersebut adalah *text mining*. Pengertian dari *Text mining* sendiri merupakan ekstraksi kata dari dokumen sehingga didapatkan keunikan dari sebuah dokumen.

Pada penelitian terdahulu yang dilakukan oleh Rasywir dan Purwarianti tentang pengklasifikasian berita hoax atau fakta menggunakan *Naïve Bayes* dengan nilai sebesar 91.36%. Dalam penelitian lain mengenai klasifikasi teks homograf juga menggunakan *Naïve Bayes* menghasilkan nilai f-measure lebih tinggi daripada *KNearest Neighbor* yakni dengan rata-rata 89% [3].

Pada penelitian lain yang dilakukan oleh Palinoan [4] mengenai klasifikasi dokumen berbahasa jawa yang memakai algoritma *K-Nearest Neighbor* menghasilkan akurasi hingga 95%.

Dalam pelaksanaan penelitian ini, klasifikasi data dari berita *hoax/non-hoax* memakai metode *Modified K-Nearest Neighbor*. *Modified K-Nearest Neighbor* disini melakukan perhitungan *cosine similarity* dan perhitungan nilai validitas pada semua data latih lalu melakukan pembobotan dokumen menggunakan metode *Term Frequency – Inverse Document Frequency* [5], [6].

## 2. Tinjauan Pustaka

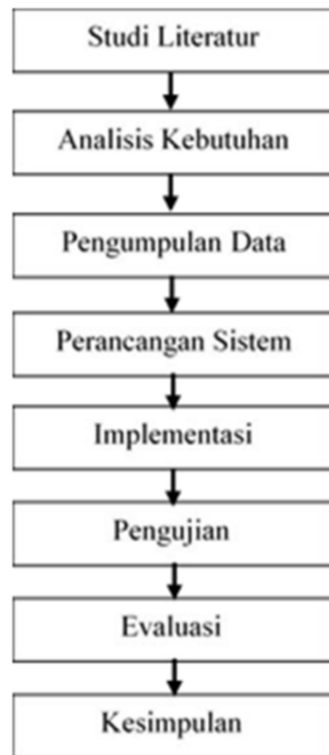
Penelitian mengenai klasifikasi berita banyak dilakukan. Diantaranya adalah penelitian Prasetijo dkk [7] yang mengembangkan filter hoax bahasa Indonesia berdasarkan representasi vektor teks berdasarkan Term Frequency dan Document Frequency serta teknik klasifikasi. Teknik klasifikasi yang digunakan dalam penelitian ini adalah Support Vector Machine dan Stochastic Gradient Descent (SGD). Akurasi tertinggi yang diperoleh dari pengklasifikasi SGD menggunakan Huber yang dimodifikasi adalah 86% lebih dari 100 tipuan dan 100 situs web non-tipuan yang dipilih secara acak di luar dataset yang digunakan dalam proses pelatihan.

Sedangkan Tacchini dkk [8] mendeteksi secara otomatis hoax dan bukan hoax pada postingan di Facebook berdasarkan pengguna yang “menyukainya”. Penelitian ini menerapkan dua teknik klasifikasi, satu berdasarkan regresi logistik, yang lain berdasarkan adaptasi baru dari algoritma crowdsourcing boolean. Pada dataset yang terdiri dari 15.500 posting Facebook dan 909.236 pengguna, diperoleh akurasi klasifikasi melebihi 99% bahkan ketika set pelatihan mengandung kurang dari 1% dari posting.

Ahmed dkk [9] memperkenalkan model *n-gram* baru untuk mendeteksi konten palsu secara otomatis dengan fokus khusus pada ulasan palsu dan berita palsu. Dalam penelitiannya juga dibandingkan 2 teknik ekstraksi fitur yang berbeda dan 6 teknik klasifikasi pembelajaran mesin. Evaluasi eksperimental menggunakan dataset publik yang ada dan dataset berita palsu yang baru.

### 3. Metode

Penelitian ini memakai metode *Modified k-Nearest Neighbor* yang sebelumnya memasukkan data latih terlebih dahulu yang digunakan untuk pembandingan dengan data uji, seperti tahapan pada Gambar 1.



**Gambar 1.** Flow diagram metodologi penelitian.

Lalu masing-masing dokumen dilakukan proses *Preprocessing* hingga pada tahap *Stemming*. Pada Proses tersebut bertujuan untuk menghilangkan semua kata imbuhan dan menghapus kata tidak penting hingga ditemukan kata dasar. Namun kata dasar tersebut belum mampu dijadikan kata kunci dokumen tersebut, dokumen tidak dapat diklasifikasikan dengan akurat. Untuk proses selanjutnya adalah dengan melakukan proses pembobotan setiap dokumen menggunakan TF-IDF dan mencari kedekatan antar dokumen menggunakan *Cosine Similarity*.

Ketika dokumen sudah diketahui tetangga terdekatnya, selanjutnya dokumen akan diklasifikasikan menggunakan *Modified K-Nearest Neighbor*. Dalam metode tersebut akan dilakukan proses pencarian nilai validasi setiap dokumen, mencari jarak antar dokumen menggunakan *Cosine Distance*, hingga mencari nilai validitas lebih tinggi dan terdekat dengan data uji menggunakan *Weight Voting*.

#### 2.1. Data Penelitian

Data yang digunakan merupakan data didapatkan dari internet berdasarkan informasi Kominfo untuk berita hoax dan dari media online untuk berita-berita non hoax tentang pilpres. Dari artikel yang didapat akan dijadikan pengujian algoritma MKNN. Pada penelitian ini dimulai dengan mempelajari studi literatur yang digunakan, berikutnya adalah menganalisis kebutuhan, mengolah data yang digunakan, merancang sistem yang ingin dibuat, menerapkan rancangan yang telah dibuat, melakukan evaluasi terhadap sistem yang telah dibuat, mengevaluasi hasil yang didapatkan berdasarkan dari hasil pengujian, langkah terakhir adalah membuat kesimpulan dari penelitian yang telah dilakukan.

#### 2.2. Preprocessing

Text Mining adalah penerapan teknik dan konsep data mining untuk mendapatkan pola pada dokumen. Pada tahap ekstraksi teks bertujuan untuk mencari informasi yang diinginkan. Definisi lain dari Text Mining adalah ekstraksi kata dari dokumen sehingga didapatkan keunikan dari sebuah dokumen, sehingga didapat keterkaitan antar dokumen.

### 2.2.1. Text Processing

*Text processing* adalah tahapan awal dimana ekstraksi dokumen menjadi data yang akan digunakan berikutnya. Sebuah teks yang ada dipecah menjadi bagian-bagian kecil sehingga didapatkan hasil akhir berupa potongan kata atau token. Selain itu akan dibuang simbol-simbol yang tidak diperlukan. Selain simbol, keberadaan digit angka juga dihilangkan. Berikut ini adalah tahapan *text processing*:

- Case Folding

*Case folding* merupakan transformasi huruf pada teks, dimana semua huruf diubah menjadi huruf kecil. Karakter huruf yang dihasilkan harus berupa 'a' sampai pada karakter huruf 'z'. Selain karakter huruf tersebut harus dianulir.

- Tokenizing

*Tokenizing* merupakan proses ekstraksi teks menjadi kata-kata, sehingga didapatkan bagian terkecil dari dokumen.

### 2.2.2. Text Transformation

Text Transformation merupakan proses yang dipakai untuk mencari kata paling dasar dari teks yang sudah dilakukan tokenizing, serta menghapus kata-kata yang dianggap tidak perlu diikuti dalam proses analisis istilah lainnya adalah stopword.

## 2.3. Pembobotan Kata

Pembobotan kata disini tergantung pada jumlah kemunculan masing-masing token dalam dokumen. Dibawah ini adalah proses perhitungan pembobotan kata [6].

### 2.3.1. Term Frequency (TF)

*Term Frequency* adalah cara untuk mencari bobot dari sebuah

dokumen. Dimana akan dicari jumlah kemunculan term pada dokumen. Semakin besar kemunculan sebuah term, akan memengaruhi besarnya bobot dan nilai kesesuaian. Berikut adalah persamaan dari *Term Frequency*:

$$W(d, t) = TF(d, t) \quad (1)$$

Dimana  $TF(d, t)$  merupakan frekuensi dari term  $t$  pada teks  $d$ .

### 2.3.2. Inverse Document Frequency (IDF)

*Inverse Document Frequency* merupakan metode untuk menghitung penyebaran term pada dokumen [10]. Dengan penyebaran term yang tidak merata memengaruhi optimasi perhitungan bobot pada dokumen. Peran dari *Inverse Document Frequency* adalah meredam penyebaran term yang terlalu bias. Jika terdapat nilai *document frequency* (DF) yang berada dibawah *threshold*, maka *term* tidak dihitung. Sehingga disimpulkan bahwa rendahnya nilai *document frequency* (DF) akan semakin tinggi nilai *Inverse Document Frequency* (IDF). Berikut adalah persamaan dari *Inverse Document Frequency* (IDF):

$$idf_t = \log_{10} \frac{N}{df_t} \quad (2)$$

Dimana  $df_t$  adalah banyaknya dokumen yang memuat  $t$  dan  $N$  merupakan jumlah total dokumen.

### 2.3.3. TF.IDF Weighting

Bobot disini merupakan hasil perkalian dari  $tft$ , dan  $idf_t$ , Rumus  $tf.idf$  weighting akan dijelaskan pada Persamaan (3) sebagai berikut.

$$W_{t,d} = \frac{W_{t,d}}{\sqrt{\sum_{t=1}^n W_{t,d}^2}} \quad (3)$$

## 2.4. Cosine Similarity

Untuk menghitung besarnya derajat kemiripan antar dokumen dan query diperlukan metode yang disebut cosine similarity. Nilai cosine similarity ditentukan oleh perhitungan besarnya nilai fungsi cosine terhadap sudut yang dibentuk oleh dua vektor yakni pada penelitian ini adalah sebuah representasi dari dokumen-dokumen antar data latih dan uji. Rumus untuk menghitung tingkat kemiripan dokumen satu dengan dokumen lain akan dijelaskan pada Persamaan (4).

$$\begin{aligned} \text{CosSim}(d_j, q) &= \vec{d}_j, \vec{q} \\ &= \sum_{i=1}^t (W_{ij} \cdot W_{iq}) \quad (4) \end{aligned}$$

Dimana  $d_j$  = data latih  $j$ ,  $q$  = tetangga data latih  $j$ ,  $W_{ij}$  = nilai pembobotan kata pada dokumen latih dan  $W_{iq}$  = nilai pembobotan kata pada tetangga dokumen latih.

## 2.5. Modified K-Nearest Neighbor

*KNearest Neighbor* (KNN) merupakan suatu metode yang menggunakan algoritma *supervised learning*. Dimana metode tersebut memerlukan data latih terlebih dahulu sebelum di proses. Tujuan dari algoritma *KNearest Neighbor* adalah untuk mencari klasifikasi baru berdasarkan tetangga terdekat yang menyusunnya atau berdasarkan pada data latih sebelumnya. Jumlah tetangga terdekat pada metode ini ditunjukkan dengan variabel  $k$ . Dengan nilai dari variabel  $k$  tersebut algoritma ini menentukan nilai prediksinya.

### 2.5.1. Nilai Validitas

Setiap data latih yang akan dianalisis menggunakan algoritma MKNN harus divalidasi berdasarkan nilai *threshold* yang sudah ditentukan

berdasarkan nilai rata-rata dari *Cosine Similarity* yang sudah didapatkan sebelumnya. Rumus nilai validitas akan dijelaskan pada Persamaan (5):

$$\text{Validity}(x) = \frac{1}{k} \sum_{i=1}^k S(\text{lbl}(x), \text{lbl}(Ni(x))) \quad (5)$$

Dimana *Validity* = Validitas antar data latih,  $k$  = Jumlah tetangga terdekat,  $(x)$  = Label pada kelas  $(x)$ ,  $(Ni(x))$  = Label kelas tetangga terdekat dengan  $(x)$ .

Sebelum pada persamaan (5) perlu diperhatikan dahulu, apakah nilai *Cosine Similarity* memenuhi syarat yang ditentukan. Dengan membandingkan data ke- $I$  dengan nilai rata-rata *Cosine Similarity*. Retang nilai dari tahap ini adalah antara 0 hingga 1. Tahap ini dirumuskan pada Persamaan (6).

$$S(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases} \quad (6)$$

Dimana  $S$  = Similarity,  $a$  = klasifikasi kelas  $a$  pada data latih dan  $b$  = klasifikasi kelas selain  $a$  pada data latih.

### 2.5.2. Cosine Distance

*Cosine distance* berfungsi untuk mencari jarak atau selisih nilai antara data uji dengan data latih. Maka rumus sederhana untuk mencari *cosine distance* adalah  $(1 - \text{Cosine Similarity})$ .

### 2.5.3. Weight Voting

Agar mendapatkan nilai *weight voting* adalah dengan mencari bobot setiap  $k$  terdekat dihitung dengan menggunakan  $1 / (de + \alpha)$ , lalu dikalikan dengan validitas setiap data berdasarkan cosine similarity. Tahap ini mempunyai dampak lebih besar terhadap data yang memiliki nilai validitas lebih tinggi dan terdekat dengan data.

**2.6. Confusion Matrix**

Confusion matrix merupakan metode untuk menguji atau mengevaluasi algoritma prediksi. Pada penelitian ini yang akan diuji adalah proses pengklasifikasian dengan metode MKKN. Diimana akan dicari nilai *Precision*, *Recall*, *Accuracy* dan *F-Measure*. Pada perhitungannya digunakan istilah *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, *False Negative (FN)*. Berikut adalah table *Confusion Matrix*.

**Tabel 1.** Confusion Matrix

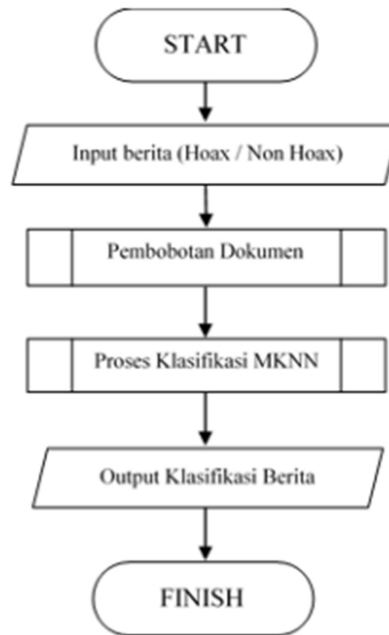
Kategori X	Predicted	
	Actual	TP
	TN	FN

**2.7. Perancangan Sistem**

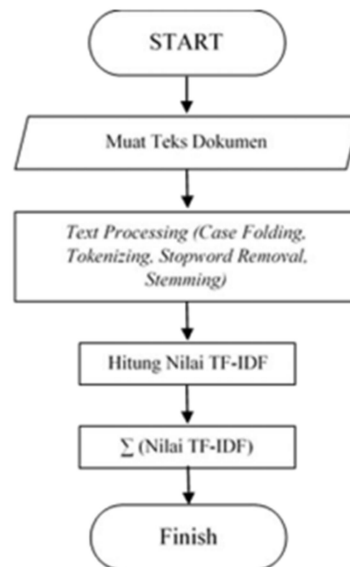
Perancangan ini memiliki penjelasan kebutuhan sebuah sistem sampai pada tahap klasifikasi sebuah dokumen, sehingga didapatkan sebuah berita hoax dan non hoax.

**2.7.1. Analisis Kebutuhan**

Tahap awal sistem adalah dengan melakukan pembobotan data latih yang bertujuan untuk mempersempit rentang data. Tahap berikutnya adalah penerapan metode MKNN dengan menghitung nilai kemiripan antar data dengan menggunakan Cosine Similarity. Setelah didapatkan nilai Cosine Similarity, selanjutnya adalah menghitung nilai validitas data berdasarkan nilai kemiripan antar data. Dari perhitungan validitas akan diambil data dengan kemiripan paling banyak dengan menggunakan metode Weight Voting. Dengan nilai dari Weight Voting akan menghasilkan klasifikasi berita hoax atau non hoax. Perancangan sistem akan ditunjukkan pada Gambar 2.



**Gambar 2.** Perancangan sistem



**Gambar 3.** Pembobotan Dokumen

**2.7.2. Perancangan Proses Pembobotan Data**

Data yang diperlukan dalam proses klasifikasi berupa dokumen yang sudah diberi bobot. Data tersebut dilakukan pembobotan agar lebih mempersempit jarak antar masingmasing data. Sistem akan melakukan perhitungan bobot data

yang akan diuji terhadap data latih yang sudah ada. Sehingga nilai bobot antar dokumen akan variatif berdasarkan data-data yang akan dilakukan proses uji. Pada Gambar 3 dijelaskan alur proses pembobotan.

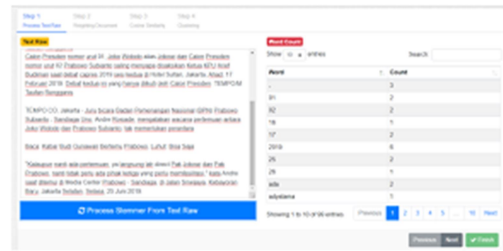
Dari Gambar 3 dijelaskan alir dari proses perhitungan TF dan IDF setiap kata pada dokumen. Kemudian dijumlahkan semua nilai pada dokumen tersebut, sehingga didapatkan sebuah bobot dari data latih maupun data uji.

#### 4. Hasil dan Pembahasan

Dari penelitian ini memiliki 2 tahapan, yaitu adalah penginputan data training dan pengklasifikasian data uji. Data training didapat dari media online dan informasi berita hoax berdasarkan kominfo pada situs resminya. Data training dan uji yang didapatkan disimpan dalam dokumen berbentuk JSON. Pada pengujian data uji dilakukan pengklasifikasian dokumen yang masih abstrak menjadi terkategori sesuai dengan tetangga terdekatnya menggunakan pendekatan Modified K-Nearest Neighbor. Setelah dokumen uji terklasifikasikan, admin dapat menambahkan dokumen tersebut menjadi data training baru.

#### 3.1. Preprocessing

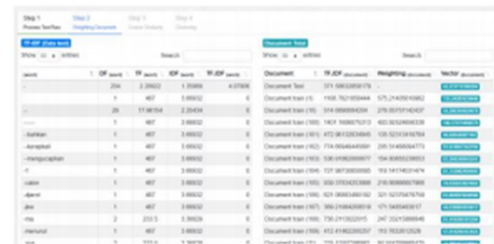
Pada tahap ini, dokumen uji akan dicari kata dasar dari setiap kalimat yang disebut dengan istilah stemming. Tahapan yang akan dilalui sistem pada proses ini adalah mulai dari Text Processing hingga Text Transformation. Tahap ini sangat penting karena awal dari proses pembobotan dan pengklasifikasian. Pada Gambar 4 menggambarkan antar muka tahap preprocessing.



Gambar 4. Antar Muka Preprocessing

#### 3.2. Implementasi TF-IDF

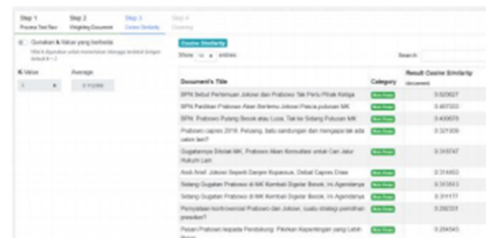
Pada tahap ini, dokumen uji maupun dokumen latih yang telah dihitung jumlah katanya, akan dicari bobotnya setiap dokumen menggunakan TF-IDF. Sehingga dari pembobotan tersebut didapatkan jarak antar dokumen dengan menggunakan vektor. Pada Gambar 5 menampilkan antar muka TF-IDF.



Gambar 5. Antar Muka TF-IDF

#### 3.3. Implementasi Cosine Similarity

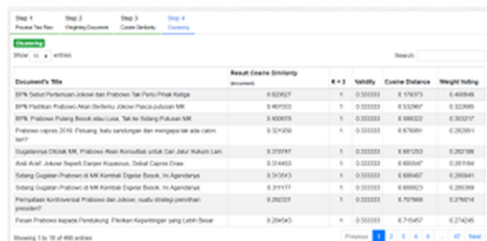
Implementasi Cosine Similarity merupakan tahap awal dari pengklasifikasian metode Modified K-Nearest neighbor. Pada tahap ini akan dicari nilai Cosine Similarity dari nilai vektor yang sudah didapatkan sebelumnya. Pada Gambar 6 adalah implementasi dari antar muka Cosine Similarity.



**Gambar 6.** Antarmuka cosine similarity.

**3.4. Implementasi Clustering**

Implementasi Clustering merupakan tahap akhir dari pengklasifikasian dari dokumen abstrak menjadi dokumen yang terklasifikasi. Pada tahap ini, sistem mengimplementasikan perhitungan Nilai Validitas, Cosine Distance dan Weight Voting sesuai dengan kaidah metode pada penelitian ini. Penerapan Modified K-Nearest Neighbor sebenarnya adalah untuk mengurutkan hasil Cosine Similarity yang sudah didapatkan sebelumnya. Dari tahap inilah artikel terindikasi hoax atau non hoax berdasarkan besarnya nilai Weight Voting yang didapat berdasarkan nilai K terdekat. Gambar 7 adalah antar muka untuk penerapan Clustering.



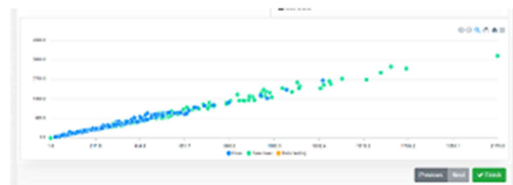
**Gambar 7.** Antar Muka Clustering

Setelah perhitungan clustering dilakukan, didapatkanlah hasil dari dokumen uji. Dokumen uji tersebut apakah masuk kedalam klasifikasi berita hoax atau masuk kedalam klasifikasi berita non hoax. Gambar 8 adalah antar muka Result Clustering.



**Gambar 8.** Antarmuka result clustering

Pada sisi kiri merupakan data/dokumen uji yang diinputkan kedalam sistem yang kemudian dibandingkan dengan data latih yang ada. Informasi penting yang disajikan adalah kapan dokumen tersebut diuji dan berapa jumlah kata yang terkandung didalam dokumen tersebut. Untuk membandingkan dokumen latih yang paling mendekati dokumen uji, ditunjukkan pada antar muka sisi sebelah kanan. Hasil pada sisi kanan meliputi informasi berapa jumlah kata yang terhitung didalam dokumen tersebut, dan berapa prosentase keakuratan sistem dalam menemukan dokumen latih serupa dengan dokumen uji. Terakhir adalah pembuktian klasifikasi dokumen kedalam Scatter Diagram yang akan ditunjukkan pada Gambar 9. Diagram tersebut berfungsi untuk menampilkan jarak antar dokumen (hoax, non hoax). Berikut adalah antar muka untuk Scatter Diagram.



**Gambar 9.** Scatter Diagram

**3.5. Pengujian Metode Modified KNearest Neighbor**

Pengujian dilakukan dengan menggunakan 65 dokumen uji yang terverifikasi oleh situs turnbackhoax.com dan diberi label BANAR atau SALAH dan 458 dokumen latih yang terklasifikasi hoax atau non hoax. Pengujian metode Modified KNearest Neighbor adalah dengan menggunakan Confusion Matrix. Dari hasil data uji yang dilakukan, sistem pengklasifikasian dievaluasi dengan Confusion Matrix. Berikut adalah hasil perhitungan



menggunakan Confusion Matrix yang dijelaskan pada Table 2.

**Tabel 2.** Hasil Perhitungan dengan Confusion Matrix

Hasil Prediksi	Hasil Aktual	
	Hoax	Non Hoax
Hoax	30	2
Non Hoax	3	30

Jika dilihat dari table diatas didapat nilai True Positive sebanyak 30, True Negative sebanyak 2, False Negative sebanyak 3 dan False Positive sebanyak 30. Setelah didapat nilai TP, TN, FN, FP selanjutnya adalah dengan menghitung nilai Precision, Recall, Accuracy, dan F-Measure dengan Confusion Matrix.

$$\begin{aligned} Precision &= \frac{TP + TN}{(TP + FP)} \times 100\% \\ &= \frac{30}{(30+2)} \times 100\% = \mathbf{93.75\%} \end{aligned}$$

$$\begin{aligned} Recall &= \frac{TP}{(TP + FN)} \times 100\% \\ &= \frac{30}{(30+3)} \times 100\% = \mathbf{90.90\%} \end{aligned}$$

$$\begin{aligned} Accuracy &= \frac{TP + TN}{(TP + FN + FP + TN)} \times 100\% \\ &= \frac{30+30}{(30+3+30+2)} \times 100\% \\ &= \mathbf{92.31\%} \end{aligned}$$

$$\begin{aligned} FMeasure &= 2x \frac{Precision \times Recall}{(Precision + Recall)} \times 100\% \\ &= 2x \frac{93.75 \times 90.90}{(93.75 + 90.90)} \times 100\% \\ &= \mathbf{92.31\%} \end{aligned}$$

## 5. Penutup

Metode Modified K-Nearest Neighbor dapat digunakan untuk melakukan klasifikasi berita hoax pilpres yang direkam kedalam sistem, kemudian dilakukan preprocessing sehingga didapatkanlah kata paling dasar dari setiap paragraf. Selanjutnya adalah dengan menghitung bobot setiap dokumen latih dan uji dengan menggunakan Term Frequency Inverse Document Frequency (TF-IDF). Setelah didapatkan sebuah bobot dari setiap dokumen, dicarilah jarak antar

dokumen latih dan dokumen uji dengan perhitungan Cosine Similarity. Pada tahap akhir adalah dengan mengurutkan dan menentukan klasifikasi dokumen uji dengan menghitung Weight Voting.

Hasil dari pengklasifikasian 65 dokumen uji beserta 466 dokumen latih menggunakan Modified K-Nearest Neighbor pada setiap dokumen, muncul nilai Precision sebesar 93.75%, nilai Recall sebesar 90.90% dan nilai Accuracy sebesar 92.30%. Dalam penentuan nilai hasil klasifikasi yang dihitung oleh sistem, semuanya bergantung kepada frekuensi kata setiap dokumen. Ketidak akuratan pengujian dikarenakan kurangnya data latih yang diinputkan kedalam sistem. Sehingga menimbulkan rendahnya prosentase kemiripan antar dokumen.

Setelah melakukan implementasi metode kedalam sistem terjadi beberapa kendala ketika melakukan penambangan text dari setiap dokumen. Yaitu memiliki selisih waktu respon bervariasi yang disebabkan oleh :

- Sumber daya / perangkat yang kurang memadai
- Kurangnya optimasi algoritma pada program
- Banyaknya dokumen latih yang dihitung

Dari sebab-sebab diatas salah satunya adalah banyaknya dokumen latih yang dihitung. Merupakan sebuah keniscayaan jika dokumen latih akan bertambah banyak. Karena dengan banyaknya data latih akan memengaruhi hasil pengujian dokumen uji. Saran untuk penelitian lebih lanjut, yaitu sebagai berikut :

- Optimasi sumber daya / perangkat yang digunakan, agar dapat melakukan perhitungan dengan cepat.

- Optimasi algoritma pemrograman, dengan menganulir variable-variabel atau fungsi-fungsi yang tidak digunakan.
- Memperbanyak dokumen latih yang dimasukkan kedalam sistem, agar memperkaya kamus dokumen sistem.
- Mengintegrasikan langsung ke media atau situs terpercaya.
- Untuk penelitian lebih lanjut, agar dapat menilai dari konteks kata atau kalimat dengan pendekatan deep learning dan lain sebagainya.
- Pembacaan urutan sintak dan maksud dari kalimat pada artikel yang diuji.

## 6. Daftar Pustaka

- [1] A. A. Sawitri, "4 Penyebab Hoax Mudah Viral di Media Sosial," *Tempo.co*, 2017. [Online]. Available: <https://nasional.tempo.co/read/838621/4-penyebab-hoax-mudah-viral-di-media-sosial>. [Accessed: 09-Aug-2018].
- [2] Novaldi, "Pakar IT: Tangkal Hoax dengan Literasi Media," *Kominfo*, 2017. [Online]. Available: [https://kominfo.go.id/content/detail/9725/pakar-it-tangkal-hoax-dengan-literasi-media/0/sorotan\\_media](https://kominfo.go.id/content/detail/9725/pakar-it-tangkal-hoax-dengan-literasi-media/0/sorotan_media). [Accessed: 21-Sep-2018].
- [3] R. Anggono, A. A. Suryani, and A. P. Kurniati, "Analisis Perbandingan Metode K-Nearest Neighbor Dan Naive Bayes Classifier Dalam Klasifikasi Teks," Universitas Telkom, 2009.
- [4] W. P. Veverly, "Sistem Klasifikasi Dokumen Bahasa Jawa Dengan Metode K- Nearest Neighbor ( K-NN )," Universitas Sanata Dharma Yogyakarta, 2014.
- [5] A. Hutapea, M. T. Furqon, and Indriati, "Penerapan Algoritme Modified K-Nearest Neighbour Pada Pengklasifikasian Penyakit Kejiwaan Skizofrenia," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 10, pp. 3957–3961, 2018.
- [6] A. R. Prasetyo and P. P. Adikara, "Klasifikasi Hoax Pada Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Modified K-Nearest Neighbor," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 7466–7473, 2018.
- [7] A. B. Prasetijo, R. R. Isnanto, D. Eridani, Y. A. A. Soetrisno, M. Arfan, and A. Sofwan, "Hoax detection system on Indonesian news sites based on text classification using SVM and SGD," in *Proceedings - 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2017*, 2017, no. October, pp. 45–49.
- [8] E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro, "Some like it Hoax: Automated fake news detection in social networks," in *CEUR Workshop Proceedings*, 2017, vol. 1960, pp. 1–15.
- [9] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Secur. Priv.*, vol. 1, no. 1, p. e9, 2018.
- [10] F. A. Hermawati and D. A. Zuhdi, "Aplikasi Sistem Temu Kembali Dokumen dengan Metode Vector Space Model," *KONVERGENSI*, vol. 5, no. 2, pp. 38–49, 2009.