Jurnal Elektronik Ilmu Komputer Udayana
Volume 10, No 4. May 2022

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Klasifikasi Berita Hoaks Covid-19 Menggunakan Kombinasi Metode *K-Nearest Neighbor* dan *Information Gain*

Marissa Audina[a1], AAIN Eka Karyawati[a2], I Wayan Supriana [a3], I Ketut Gede Suhartana [a4], I Gede Santi Astawa [a5], I Wayan Santiyasa[a6]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Bali, Indonesia
[1]marissaaudina@gmail.com
[2]eka.karyawati@unud.ac.id
[3]wayan.supriana@unud.ac.id
[4]ikg.suhartana@unud.ac.id
[5]santi.astawa@unud.ac.id
[6]santiyasa@unud.ac.id

Abstract

News is one of information resources that is being used by the public. However, not all news circulating in digital media are facts. Some people take the opportunity to share unfounded and irresponsible news. Since the Covid-19 pandemic hit Indonesia, hoax news about the pandemic has increasingly circulated in digital media. In this study, the author builds a model that can classify hoax news using the K-Nearest Neighbor method combined with the Information Gain feature selection. The data used are factual news data and hoax news data in Indonesian language. Evaluation is done by measuring the performance of the K-Nearest Neighbor model without feature selection and model performance by implementing Information Gain feature selection. The K-Nearest Neighbor model without feature selection with a value of k=5 obtained precision, recall, F1-Score, and accuracy performance of 87.5%, 96.5%, 91.8%, and 91.6%, respectively. While the K-Nearest Neighbor model with a combination of 0.5% Information Gain threshold feature selection with a value of k=3 obtained precision, recall, F1-Score, and accuracy performance of 93.3%, 96.6%, 95%, and 95%, respectively.

Keywords: *K-Nearest Neighbor*, *Information Gain*, TF-IDF, Klasifikasi Teks, Berita Hoaks

1. Pendahuluan

Berita digunakan oleh masyarakat sebagai salah satu sumber informasi. Tidak semua berita yang beredar di media digital adalah fakta. Beberapa individu atau kelompok mengambil kesempatan untuk menyebarkan berita atau informasi yang tidak dapat dipertanggungjawabkan kebenarannya dan terdapat indikasi *hoax*.[1] Data dari laman resmi kominfo.go.id menyatakan bahwa sebanyak 800.000 situs terindikasi sebagai situs penyebaran hoaks di Indonesia. Menurut Kamus Besar Bahasa Indonesia, hoaks (bahasa Inggris: *hoax*) memiliki makna informasi bohong. Sejak pandemi Covid-19 melanda Indonesia, berita hoaks mengenai pandemi tersebut semakin banyak beredar di media digital. Data terbaru dari Kementerian Komunikasi dan Informatika, sebanyak 5457 sebaran hoaks Covid-19 sudah ditindaklanjuti sejak 23 Januari 2020 hingga 18 Maret 2022 [2].

Berita-berita yang didapatkan dari media dapat diklasifikasikan menjadi berita hoaks dan berita fakta. Pengklasifikasian berita tersebut membutuhkan suatu metode atau algoritma agar tidak menggunakan cara manual dan menghabiskan waktu yang lama. Peranan informatika dibutuhkan dalam hal ini untuk membangun suatu model klasifikasi yang dapat mengkategorikan dua jenis berita tersebut. Penelitian mengenai klasifikasi berita hoaks telah dilakukan oleh beberapa peneliti seperti penelitian klasifikasi berita *clickbait* menggunakan *K-Nearest Neighbor* yang menghasilkan akurasi terbaik 71% dengan parameter nilai k=11 pada skenario 80% data latih dan 20% data uji [3]. Kemudian penelitian mengenai identifikasi hoaks berbasis *text mining* menggunakan *K-Nearest Neighbor* menghasilkan akurasi sebesar 75.4% pada nilai k optimal

bernilai 4 [4]. Penelitian selanjutnya adalah analisis sentimen terhadap ulasan pengguna MRT Jakarta menggunakan *Information Gain* dan *Modified K-Nearest Neighbor* dengan peningkatan akurasi 4-5% setelah menggunakan seleksi fitur *Information Gain* [5].

Berdasarkan penelitian yang dilakukan sebelumnya, pada penelitian ini penulis melakukan klasifikasi berita hoaks menggunakan metode *K-Nearest Neighbor* yang dikombinasikan dengan seleksi fitur *Information Gain*. Penulis berharap bahwa dengan menggunakan kombinasi metode ini dapat menghasilkan performa *precision, recall, f1-score,* dan akurasi yang lebih baik dibandingkan penelitian sebelumnya.

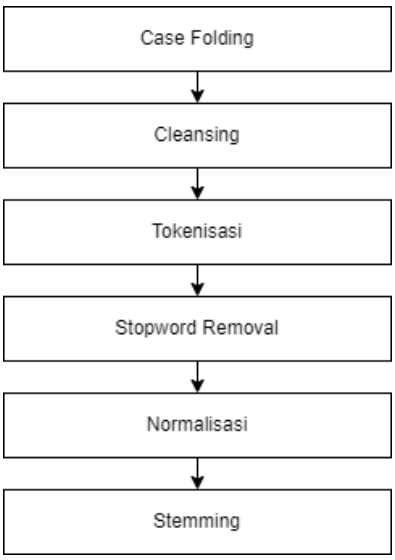
2. Metode Penelitian

2.1 Dataset

Jenis data sekunder digunakan pada penelitian ini. Dataset diperoleh dalam bentuk berita yang berkaitan dengan Covid-19. Data berita hoaks bersumber dari https://cekfakta.com, sedangkan data berita fakta bersumber dari https://detik.com. Bagian berita yang digunakan adalah isi berita. Data berjumlah 300 dengan format *file* *.csv yang meliputi 150 berita hoaks dan 150 berita fakta yang bersumber dari media internet. Seluruh data sudah dilabeli oleh lembaga media internet tersebut. Data berita kemudian dibagi dengan presentase data latih sebesar 80% dan data uji sebesar 20%. Data latih tersebut kemudian dibagi lagi menjadi data latih dan data validasi untuk digunakan dalam proses pelatihan model dengan menggunakan *N-Fold Cross Validation* dengan nilai $N = 10$.

2.2 *Preprocessing*

Sebelum melakukan tahap pembobotan, data terlebih dahulu melalui tahapan *preprocessing. Preprocessing* adalah proses yang dilakukan untuk mengolah data ulasan yang belum terstruktur menjadi terstruktur sehingga data dapat dilanjutkan ke proses klasifikasi. Adapun alur *preprocessing* seperti pada Gambar 1.



Gambar 1. Alur *Preprocessing*

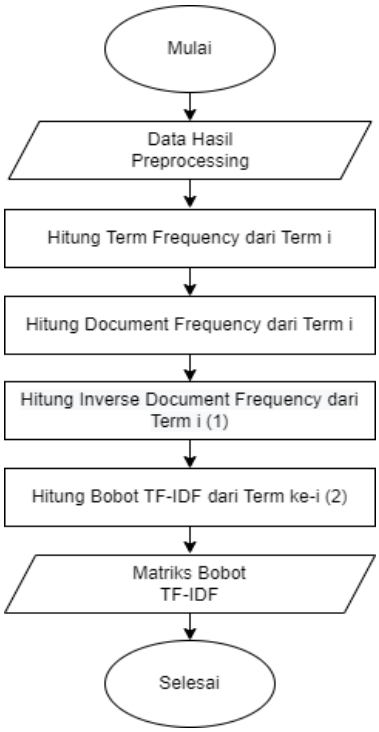
Data dokumen berita akan melalui proses *case folding* yaitu proses untuk membuat bentuk data yang sama yaitu hanya berisi huruf kecil. *Case folding* dilakukan agar data yang ada menjadi sama rata [6]. Kemudian proses *cleansing* untuk menghapus seluruh karakter yang berupa HTML ataupun web yang tidak memiliki makna atau kaitan terhadap analisis sentimen. Pada proses ini juga dilakukan proses penghapusan *punctuation* atau tanda baca. *Tokenization* adalah proses pemisahan kata dalam suatu paragraf atau kalimat sehingga terbagi menjadi token-token tertentu. *Stopword removal* merupakan proses penghapusan kata atau fitur yang tidak berpengaruh dan tidak penting terhadap klasifikasi. Penghapusan ini dilakukan untuk membuat proses klasifikasi berjalan efisien [6]. Kemudian proses normalisasi, yaitu mengubah dan

p-ISSN: 2301-5373
e-ISSN: 2654-5101

mengembalikan bentuk penulisan tidak baku ke bentuk penulisan yang sesuai dengan KBBI. Pada proses ini digunakan korpus yang berisi kumpulan kata tidak baku dan bentuk baku dari kata tersebut. Proses terakhir adalah *stemming* yang berfungsi agar kata-kata berimbuhan (awalan dan akhiran) dapat diekstraksi ke bentuk akarnya atau dapat dikatakan sebagai kata dasar. *Stemming* dilakukan untuk menyamakan data yang berbeda penulisannya [6].

2.3 *Term Frequency Inverse Document Frequency* (TF-IDF)

TF-IDF merupakan metode pembobotan kata untuk menentukan keterhubungan kata pada suatu dokumen [4]. Data yang diproses akan diubah menjadi data numerik dengan metode pembobotan TF-IDF, yang merupakan penggabungan dua konsep yaitu TF dan IDF. *Term Frequency* (TF) merupakan frekuensi dari kemunculan kata dalam sebuah dokumen, sedangkan *Inverse Document Frequency* (IDF) adalah perhitungan dari distribusi kata secara luas pada koleksi dokumen. Kata atau *term* yang muncul di dalam sebagian besar dokumen akan mempunyai nilai IDF mendekati nol. Adapun tahapan dari TF-IDF ditunjukkan oleh Gambar 2.



Gambar 2. TF-IDF

a. Menghitung jumlah kemunculan *term* i dalam dokumen j ($tf_{i,j}$).
b. Menghitung jumlah dokumen yang mengandung *term* i (df_i)
c. Menghitung nilai bobot *inverse document frequency* (idf) dengan menggunakan persamaan :

$$idf_{\mathrm{i}} = log\,(\frac{N}{df_i}) \quad (1)$$

Keterangan :
N = jumlah dokumen secara keseluruhan

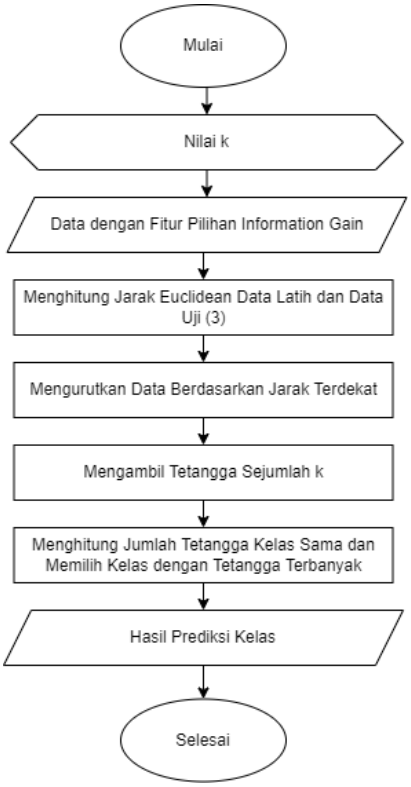
d. Menghitung nilai bobot TF-IDF dengan menggunakan persamaan :

$$w_{i,j} = tf_{i,j}\, x\, idf_i \quad (2)$$

Keterangan :
$w_{i,j}$ = bobot *term* i terhadap dokumen j
$tf_{i,j}$ = frekuensi *term* i pada dokumen j
idf_i = nilai bobot IDF *term* i

2.4 *K-Nearest Neighbor* (KNN)

Metode KNN sering diterapkan pada *data mining* dan *text mining*. KNN merupakan metode pengklasifikasian objek berdasarkan tetangga yang paling dekat dengannya. KNN memberikan keanggotaan kelas ke data berdasarkan mayoritas tetangganya, dengan objek yang ditetapkan ke kelas yang paling umum di antara k tetangga terdekatnya (k adalah bilangan bulat positif bernilai kecil). Pada penelitian ini, fitur kata yang sudah melalui proses pembobotan TF-IDF akan menghasilkan suatu matriks yang berisikan bobot nilai TF-IDF dengan dokumen sebagai baris dan fitur kata sebagai kolom. Setiap vektor dokumen dengan nilai bobot fitur pada data latih akan dihitung jaraknya dengan vektor pada data uji. Adapun tahapan metode ditunjukkan oleh Gambar 3.



Gambar 3. Klasifikasi *K-Nearest Neighbor*

Tahapan dari KNN adalah sebagai berikut :

a. Menentukan jumlah tetangga k yang akan digunakan.
b. Melakukan perhitungan jarak antara data latih dan data uji menggunakan rumus persamaan *Euclidean distance* di bawah [7]:

$$\mathrm{d}(\mathrm{x},\mathrm{y}) = \sqrt{\sum_{\mathrm{k}=1}^{\mathrm{n}}(\mathrm{x_k} - \mathrm{y_k})^2} \quad (3)$$

Keterangan :
d = jarak
x = data uji (*data testing*)
y = data latih (*data training*)

p-ISSN: 2301-5373
e-ISSN: 2654-5101

c. Mendapatkan hasil pengklasifikasian

2.5 *Information Gain*

Information gain banyak digunakan pada klasifikasi data tekstual sebagai metode seleksi fitur. *Information gain* melakukan perhitungan mengenai pengaruh suatu fitur terhadap keseragaman kelas pada data. Seleksi fitur ini menghitung hadir tidaknya suatu kata yang berkontribusi pada pengambilan keputusan klasifikasi yang benar di kelas apapun [8]. Data tersebut dipecah menjadi sub data dengan nilai fitur tertentu. Jika suatu fitur memperoleh nilai *information gain* yang tinggi, maka fitur tersebut dikatakan memiliki pengaruh pada proses klasifikasi. Adapun tahapan *Information Gain* ditunjukkan oleh Gambar 4.

Gambar 4. *Information Gain*

Langkah-langkah dari *Information Gain* adalah sebagai berikut :
a. Isi data dan label data sebagai input.
b. Melakukan perhitungan nilai *Information Gain* dari setiap fitur dengan rumus berikut [9] :

$$\text{Information Gain I(tj) of tj} = -\sum_{r=1}^{k} \frac{n_r}{n} \log\left(\frac{n_r}{n}\right) - \text{E(tj)} \quad (4)$$

Dimana E(tj) adalah *entropy* bersyarat yang dihitung dengan persamaan :

$$\text{E(tj)} = -\sum_{r=1}^{k} \left\{\left[\frac{n(tj)}{n}\right] P(c_r|tj).\log[P(c_r|tj)] + \left[\frac{n-n(tj)}{n}\right] \text{P}(c_r|\neg\text{tj}).\log[\text{P}(c_r|\neg\text{tj})]\right\} \quad (5)$$

Keterangan :
n_r : jumlah total dokumen dengan kelas r.

n : jumlah total dokumen.
n(tj) : banyaknya dokumen yang mengandung term tj dari korpus berukuran n ≥ n (tj).
$P(c_r|tj)$: peluang *term* tj terdapat pada kelas r dalam dokumen.
$P(c_r|\neg tj)$: peluang *term* tj tidak terdapat pada kelas r dalam dokumen.

c. Mengurutkan fitur-fitur berdasarkan nilai *Information Gain* tertinggi.
d. Memilih fitur terbaik sesuai dengan *threshold* yang diberikan.

2.6 Evaluasi

Pada tahap evaluasi, *confusion matrix* digunakan untuk menghitung akurasi, *recall, precision,* dan *error rate. Confusion matrix* dapat digunakan untuk mengevaluasi kualitas *classifier*. Pada *confusion matrix* dua kelas, matriks menunjukkan *true positives, true negatives, false positives,* dan *false negatives*. C*onfusion matrix* untuk dua kelas ditunjukkan pada Tabel 1 [10].

Tabel 1. *Confusion Matrix*

Kelas Sebenarnya	**Prediksi Kelas**	
	Positif	**Negatif**
Positif	TP	FN
Negatif	FP	TN

Keterangan :

TP = *True Positive* (total prediksi benar dari data positif)

FN = *False Negative* (total prediksi salah dari data positif)

TN = *True Negative* (total prediksi benar dari data negatif)

FP = *False Positive* (total prediksi salah dari data negatif)

Adapun rumus untuk menghitung *precision, recall, F1-Score,* dan akurasi adalah sebagai berikut:

$$\textit{Precision} = \frac{TP}{(TP+FP)} \quad (6)$$

$$\textit{Recall} = \frac{TP}{(TP+FN)} \quad (7)$$

$$\textit{F1-Score} = \frac{2\ x\ recall\ x\ precision}{(recall+precision)} \quad (8)$$

$$\text{Akurasi} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (9)$$

3. Hasil dan Pembahasan

Sebanyak 80% dari total data digunakan pada tahap pelatihan dan validasi. Beberapa eksperimen dilakukan yaitu model KNN tanpa seleksi fitur, model KNN dengan eksperimen beberapa nilai *threshold Information* Gain, dan pengujian kedua model terbaik dengan menggunakan data baru yaitu 20% data uji yang sudah disiapkan sebelumnya. Perubahan nilai k dilakukan pada metode *K-Nearest Neighbor.* Perubahan nilai k pada eksperimen adalah k=3, k=5, k=7, k=9, dan k=11. Pengujian terhadap *threshold* dilakukan pada seleksi fitur *Information Gain*. *Threshold* adalah persentase jumlah fitur yang terseleksi dari seluruh fitur yang telah diurutkan berdasarkan nilai *Information Gain* tertinggi. *Threshold* yang digunakan adalah 50%, 25%, 20%, 10%, 5%, 2%, 1%, 0.5%, 0.2%, dan 0.1%. Pada setiap iterasi 10-*Fold Cross Validation*, akan dihitung rata-rata performa *F1-Score* dan akurasi dengan menggunakan persamaan (8) dan (9). Nilai k yang menghasilkan performa *F1-Score* tertinggi dipilih sebagai model terbaik yang kemudian digunakan pada proses *testing* data baru. Nilai k yang

p-ISSN: 2301-5373
e-ISSN: 2654-5101

menghasilkan *F1-Score* tertinggi memiliki makna bahwa hasil prediksi berita hoaks akan lebih akurat kebenarannya.

Setelah dilakukan proses pelatihan dan validasi terhadap model *K-Nearest Neighbor* menggunakan 10-*Fold Cross Validation*, didapatkan nilai k dengan performa *F1-Score* terbaik yaitu k = 5 dengan nilai *F1-Score* 93.9% serta akurasi 94.2%. Sehingga, nilai k = 5 dipilih sebagai model terbaik dan akan digunakan pada proses pengujian data uji dengan menggunakan data baru yang belum pernah melalui proses pelatihan dan validasi.

Tabel 2. Hasil Evaluasi Pengujian *K-Nearest Neighbor* tanpa Seleksi Fitur

Nilai k	Ukuran Evaluasi (Rata-Rata Fold)	
	F1-Score	Akurasi
3	0.908	0.908
5	0.939	0.942
7	0.927	0.929
9	0.934	0.938
11	0.935	0.938

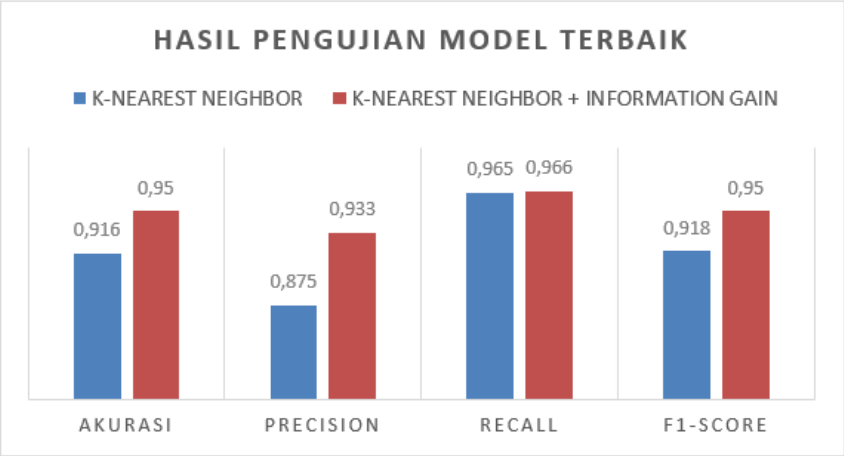
Jumlah keseluruhan fitur adalah 4934 fitur. Pada setiap eksperimen perubahan nilai *threshold*, dilakukan juga perubahan nilai k sehingga didapatkan kombinasi *threshold* dan parameter nilai k yang menghasilkan performa terbaik. Kombinasi *threshold* 0.5% dengan parameter nilai k=3 menghasilkan performa terbaik pada proses pelatihan dan validasi yaitu *F1-Score* sebesar 97.3%, serta akurasi sebesar 97.5%, sehingga kombinasi ini dipilih menjadi model terbaik. *Threshold* ini menyeleksi sekitar 25 fitur dengan nilai *Information Gain* tertinggi. Fitur-fitur tersebut terdiri dari beberapa fitur yang dominan pada dokumen kelas hoaks, dan beberapa fitur lainnya dominan pada dokumen kelas fakta. Fitur-fitur tersebut menjadi ciri khas dari kedua kelas berita sehingga model dapat mengklasifikasikan berita dengan baik, ditunjukkan oleh performa akurasi yang dihasilkan.

Tabel 3. Hasil Evaluasi Pengujian *K-Nearest Neighbor* dengan Seleksi Fitur

Threshold	Ukuran Evaluasi (Rata-Rata Fold)	
	F1-Score	Akurasi
50%	0.255	0.575
25%	0.271	0.579
20%	0.266	0.58
10%	0.869	0.875
5%	0.912	0.913
2%	0.952	0.954
1%	0.958	0.958
0.5%	0.973	0.975
0.2%	0.937	0.942
0.1%	0.925	0.929

Dua model terbaik yang dipilih adalah model yang menghasilkan *F1-Score* terbaik, yaitu model KNN tanpa seleksi fitur dengan nilai k = 5, dan model KNN dengan kombinasi seleksi fitur *Information Gain threshold* 0.5% dengan nilai k=3. Kedua model tersebut kemudian diuji kembali menggunakan data baru yang belum pernah melewati tahap pelatihan dan validasi sebelumnya.

Setelah melakukan pengujian tehadap kedua model dengan menggunakan data baru, hasil performa model ditunjukkan pada Gambar 5.



Gambar 5. Hasil Pengujian Model Terbaik

Pada Gambar 5 ditunjukkan bahwa kombinasi metode KNN dan *Information Gain* menghasilkan performa yang lebih baik dalam klasifikasi berita hoaks. Terdapat peningkatan pada setiap performa evaluasi model. Model KNN tanpa seleksi fitur dengan nilai k=5 menghasilkan performa *precision*, *recall, F1-Score,* dan akurasi secara berturut-turut yaitu 87.5%, 96,5%, 91,8%, dan 91,6%. Sedangkan model KNN dengan kombinasi seleksi fitur *Information Gain threshold* 0.5% dengan nilai k=3 menghasilkan performa *precision*, *recall, F1-Score,* dan akurasi secara berturut-turut yaitu 93.3%, 96.6%, 95%, dan 95%. Pada nilai *recall* kedua model, tidak terdapat perbedaan yang signifikan. *Recall* menghitung presentase prediksi kelas hoaks benar terhadap seluruh data yang kelas sebenarnya adalah hoaks. Hal ini menandakan bahwa kedua model terbaik yang diuji sama-sama dapat dengan baik mengklasifikan berita hoaks ke dalam kelas hoaks dengan sedikit kesalahan pada klasifikasi berita kelas hoaks ke dalam kelas fakta.

4. Kesimpulan

Setelah dilakukan validasi model dengan 10-*Fold Cross Validation*, nilai k=5 dipilih menjadi model terbaik pada eksperimen metode *K-Nearest Neighbor*. Pada pengujian data baru, model *K-Nearest Neighbor* tanpa seleksi fitur dengan nilai k=5 menghasilkan performa *precision*, *recall, F1-Score,* dan akurasi secara berturut-turut yaitu 87.5%, 96.5%, 91.8%, dan 91.6%. Pada eksperimen seleksi fitur *Information Gain*, kombinasi *threshold* 0.5% dengan parameter nilai k=3 adalah kombinasi yang dipilih menjadi model terbaik. *Threshold* ini menyeleksi sekitar 25 fitur dengan nilai *Information Gain* tertinggi. Pada pengujian data baru, model *K-Nearest Neighbor* dengan seleksi fitur *Information Gain* menghasilkan performa *precision*, *recall, F1-Score,* dan akurasi secara berturut-turut yaitu 93.3%, 96.6%, 95%, dan 95%. Sehingga, dapat disimpulkan bahwa pada klasifikasi berita hoaks dengan data yang digunakan pada penelitian ini, kombinasi metode *K-Nearest Neighbor* dan *Information Gain* menghasilkan performa *precision*, *recall, F1-Score,* dan akurasi yang lebih tinggi dibandingkan dengan metode *K-Nearest Neighbor* tanpa seleksi fitur.

Daftar Pustaka

[1] C. Juditha, “Interaksi Komunikasi Hoax di Media Sosial serta Antisipasinya,” *J. Pekommas*, vol. 3, no. 1, pp. 31–44, 2018, [Online]. Available: https://jurnal.kominfo.go.id/index.php/pekommas/article/view/2030104.

[2] Kominfo.go.id, “Penanganan Sebaran Konten Hoaks Covid-19 Jumat (18/02/2022),” 2022. https://kominfo.go.id/content/detail/40067/penanganan-sebaran-konten-hoaks-

p-ISSN: 2301-5373
e-ISSN: 2654-5101

covid-19-jumat-18022022/0/infografis (accessed Mar. 26, 2022).

[3] R. Sagita, U. Enri, and A. Primajaya, “Klasifikasi Berita Clickbait Menggunakan K-Nearest Neighbor (KNN),” *JOINS (Journal Inf. Syst.*, vol. 5, no. 2, pp. 230–239, 2020, doi: 10.33633/joins.v5i2.3705.

[4] I. W. Santiyasa, G. P. A. Brahmantha, I. W. Supriana, I. G. G. A. Kadyanan, I. K. G. Suhartana, and I. B. M. Mahendra, “Identification of Hoax Based on Text Mining Using K-Nearest Neighbor Method,” *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 10, no. 2, pp. 217–226, 2021, doi: 10.24843/jlk.2021.v10.i02.p04.

[5] A. A. Paramitha, Indriati, and Y. A. Sari, “Analisis Sentimen Terhadap Ulasan Pengguna MRT Jakarta Menggunakan Information Gain dan Modified K-Nearest Neighbor,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 4, pp. 1125–1132, 2020.

[6] K. D. Yonatha Wijaya and A. A. I. N. E. Karyawati, “The Effects of Different Kernels in SVM Sentiment Analysis on Mass Social Distancing,” *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 9, no. 2, p. 161, 2020, doi: 10.24843/jlk.2020.v09.i02.p01.

[7] H. P. Hadi and T. S. Sukamto, “Klasifikasi Jenis Laporan Masyarakat Dengan K-Nearest Neighbor Algorithm,” *JOINS (Journal Inf. Syst.*, vol. 5, no. 1, pp. 77–85, 2020, doi: 10.33633/joins.v5i1.3355.

[8] A. B. P. Negara, H. Muhardi, and I. M. Putri, “Analisis Sentimen Maskapai Penerbangan Menggunakan Metode Naive Bayes dan Seleksi Fitur Information Gain,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 3, pp. 599–606, 2020, doi: 10.25126/jtiik.2020711947.

[9] C. C. Aggarwal and C. C. Aggarwal, *Machine Learning for Text: An Introduction*. 2018.

[10] M. A. Imron and B. Prasetiyo, “Improving Algorithm Accuracy K-Nearest Neighbor Using Z-Score Normalization and Particle Swarm Optimization to Predict Customer Churn,” *J. Soft Comput. Explor.*, vol. 1, no. 1, pp. 56–62, 2020, [Online]. Available: https://shmpublisher.com/index.php/joscex/article/view/7%0Ahttps://shmpublisher.com/index.php/joscex/index.

This page is intentionally left blank.