 ISSN:2302-8149
e-ISSN:2540-9719

Fitur *Word Embedding* untuk meningkatkan Kinerja *Machine Learning* pada Analisis Sentimen *Game Honor of Kings*

Word Embedding Features to Improve Machine Learning Performance in Sentiment Analysis of the Honor of Kings Game

[1]Abdul Harris*, [2]Agus Nugroho, [3]Yudi Novianto, [4]Jasmir Jasmir, [5]Dhea Fatma
[1,2,3,5]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dinamika Bangsa
[4]Program Studi Sistem Komputer, Fakultas Ilmu Komputer Universitas Dinamika Bangsa
[1,2,3,4,5]Jl. Jendral Sudirman, Tehok, Jambi Selatan, Kota Jambi, Indonesia
*e-mail: abdulharris@stikom-db.ac.id

(***received*:** 13 November 2025, ***revised*:** 13 December 2025, ***accepted*:** 10 January 2026)

Abstrak

Perkembangan media sosial mendorong semakin banyak penelitian mengenai analisis sentimen untuk memahami persepsi dan opini masyarakat. Penelitian ini bertujuan mengevaluasi kinerja algoritma *machine learning Naïve Bayes*, *K-Nearest Neighbor* (KNN), dan *Random Forest* dalam mengklasifikasikan sentimen ulasan pengguna terhadap *Game Honor of Kings*. Data penelitian diperoleh dari *Google Play Store* dengan total 900 ulasan, kemudian melalui tahapan prapemrosesan meliputi *cleaning*, *case folding*, *tokenization*, *stopword removal*, *stemming*, serta pelabelan sentimen positif dan negatif. Selanjutnya, tiga teknik *word embedding* digunakan, yaitu *Word2Vec*, *GloVe*, dan *FastText*, yang masing-masing diuji terhadap tiga algoritma *machine learning*. Hasil eksperimen menunjukkan bahwa penggunaan fitur *word embedding* secara signifikan meningkatkan akurasi klasifikasi dibandingkan tanpa fitur. KNN dengan *FastText* menghasilkan performa terbaik dengan akurasi 87,55%, sedangkan *Random Forest* dengan *FastText* memberikan akurasi terendah. *FastText* terbukti unggul karena kemampuannya merepresentasikan kata melalui sub-kata, sehingga lebih efektif dalam menangani kosakata langka maupun data berskala besar. Penelitian ini menegaskan bahwa kombinasi metode klasifikasi dengan fitur *word embedding* berperan penting dalam meningkatkan performa analisis sentimen. Ke depan, penelitian lanjutan dapat diarahkan pada optimasi hiperparameter, penerapan teknik prapemrosesan lanjutan, serta perluasan dataset agar model lebih tangguh dan mampu melakukan generalisasi yang lebih baik.

Kata kunci: analisis sentiment, *fasttext*, *glove*, *machine learning*, *word2vec*

Abstract

The rapid growth of social media has encouraged an increasing number of studies on sentiment analysis to better understand public perceptions and opinions. This study aims to evaluate the performance of three machine learning algorithms—Naïve Bayes, K-Nearest Neighbor (KNN), and Random Forest—in classifying user review sentiments toward the game Honor of Kings. The dataset was collected from the Google Play Store, consisting of 900 reviews. The data then underwent preprocessing steps including cleaning, case folding, tokenization, stopword removal, stemming, and sentiment labeling into positive and negative classes. Furthermore, three word embedding techniques were applied, namely Word2Vec, GloVe, and FastText, each of which was tested across the three machine learning algorithms. The experimental results indicate that the use of word embedding features significantly improves classification accuracy compared to models without embedding features. KNN combined with FastText achieved the best performance, reaching an accuracy of 87.55%, while Random Forest combined with FastText produced the lowest accuracy. FastText demonstrated superior performance due to its ability to represent words through subword information, making it more effective in handling rare vocabulary and large-scale datasets. This study confirms that combining machine learning classification methods with word embedding features plays a crucial role in improving sentiment analysis performance. Future research may focus on hyperparameter optimization, the application of more advanced preprocessing techniques, and dataset expansion to develop more robust models with better generalization capability.

 ISSN:2302-8149
e-ISSN:2540-9719

Keywords: *fasttext, glove, machine learning, sentiment analysis, word2vec*

1 Pendahuluan

Perkembangan teknologi informasi dan media sosial telah menghasilkan peningkatan signifikan dalam volume data teks yang memuat opini, emosi, dan persepsi pengguna. Analisis sentimen menjadi pendekatan penting untuk mengekstraksi polaritas emosi tersebut secara otomatis menggunakan teknik *Natural Language Processing* (NLP) dan *machine learning*[1][2]. Metode ini banyak dimanfaatkan dalam pengambilan keputusan bisnis, pemantauan opini publik, serta pengembangan sistem rekomendasi berbasis persepsi pengguna [3][4].

Berbagai penelitian telah menggunakan algoritma *machine learning* seperti *Naïve Bayes*, *K-Nearest Neighbor* (KNN), dan *Random Forest* untuk klasifikasi teks [5][6]. Namun, performa model sering bervariasi karena perbedaan representasi fitur. Pendekatan tradisional seperti *Bag-of-Words* (BoW) dan *Term Frequency–Inverse Document Frequency* (TF-IDF) kurang mampu menangkap hubungan semantik dan konteks linguistik yang kompleks [7]. Oleh karena itu, diperlukan representasi fitur yang lebih informatif, seperti *word embedding* (*Word2Vec, GloVe,* dan *FastText*) yang memetakan kata ke ruang vektor sehingga hubungan semantik dapat terpelihara [8][9].

Setiap teknik *word embedding* memiliki karakteristik dan keterbatasan masing-masing. *Word2Vec* unggul dalam mempelajari konteks semantik tetapi lemah terhadap kata jarang muncul [10]. *GloVe* efektif untuk pembelajaran berbasis ko-kemunculan kata, namun membutuhkan korpus besar untuk menghasilkan vektor yang stabil [11]. *FastText* lebih tangguh untuk kata langka karena menggunakan representasi sub-kata, tetapi efektivitasnya pada domain ulasan game berbahasa Indonesia belum banyak diuji [12]. *Research gap* yang muncul adalah minimnya kajian komparatif yang mengevaluasi bagaimana integrasi tiga teknik embedding tersebut dengan algoritma machine learning konvensional memengaruhi performa klasifikasi sentimen pada domain ulasan pengguna game daring, khususnya Honor of Kings.

Pemilihan tiga algoritma *Naïve Bayes*, KNN, dan *Random Forest* didasarkan pada karakteristiknya yang berbeda sehingga memungkinkan analisis komparatif yang lebih kaya. *Naïve Bayes* dikenal efisien pada data teks berdimensi tinggi. KNN digunakan karena pendekatan berbasis kedekatan jarak sangat dipengaruhi kualitas *embedding*, sehingga cocok untuk mengukur kontribusi representasi vektor. *Random Forest* dipilih karena kemampuannya menangani non-linearitas dan variabilitas fitur sehingga sering menjadi baseline kuat pada klasifikasi teks. Kombinasi ini memungkinkan evaluasi menyeluruh mengenai bagaimana representasi *embedding* berinteraksi dengan tipe algoritma yang berbeda

Berdasarkan masalah tersebut, penelitian ini berkontribusi pada evaluasi pengaruh *Word2Vec, GloVe,* dan *FastText* terhadap kinerja tiga algoritma *machine learning* tersebut pada kasus analisis sentimen ulasan pengguna *Game Honor of Kings*. Kontribusi utama penelitian ini meliputi:

1. Memberikan analisis komprehensif mengenai efektivitas kombinasi antara algoritma *machine learning* dan fitur *word embedding* dalam meningkatkan akurasi klasifikasi sentimen.
2. Menunjukkan peran signifikan fitur *FastText* dalam mengurangi tingkat *false positive* dan *false negative* melalui pemanfaatan representasi sub-kata.
3. Memberikan rekomendasi empiris bagi peneliti dan praktisi NLP dalam memilih teknik representasi fitur yang sesuai untuk data ulasan pengguna game berbahasa Indonesia.

Dengan demikian, penelitian ini diharapkan dapat memperkuat pemahaman mengenai integrasi *word embedding* dan algoritma *machine learning* konvensional, sekaligus mendukung pengembangan model analisis sentimen yang lebih akurat dan generalis.

2 Tinjauan Literatur

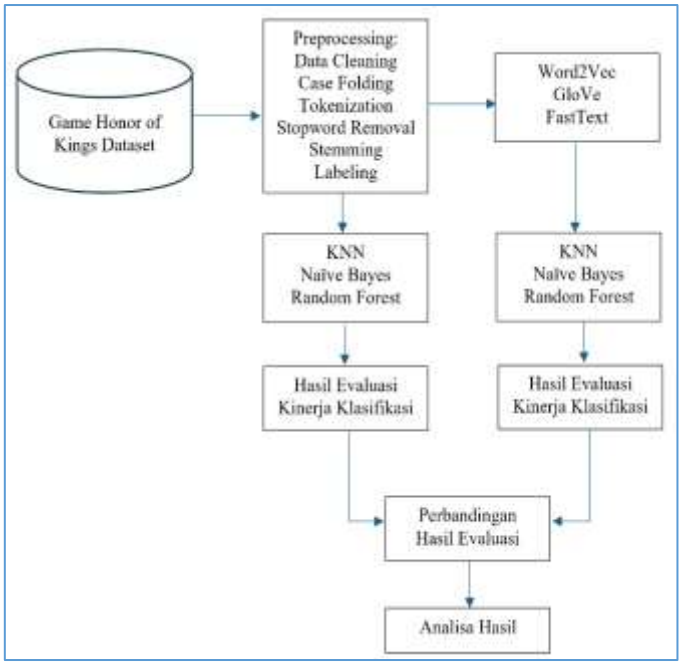
Beberapa penelitian sebelumnya telah mengeksplorasi analisis sentimen. Misalnya, Elik Hari Muktafin menganalisis kepuasan pelanggan pelayanan publik menggunakan KNN dengan fitur TF-IDF, mencapai akurasi 74%[5]. Heru Agus Santoso dkk. meneliti analisis sentimen berita hoax menggunakan *Naive Bayes*, menghasilkan akurasi sebesar 77%[6]. Sementara itu, M. Ali Fauzi melakukan analisis sentimen dalam bahasa Indonesia menggunakan *Random Forest*, memperoleh nilai rata-rata *Out of Bag* (OOB) sebesar 82,9%[7]. Ari Basuki melakukan penelitian tentang analisis sentimen ulasan pelanggan penyedia Jasa Pengiriman di Twitter menggunakan Klasifikasi *Naive*

 ISSN:2302-8149
e-ISSN:2540-9719

Bayes dan menghasilkan akurasi yang rendah yaitu 50,6%[8]. Kartikasari Kusuma Agustiningsih menganalisis sentimen masyarakat Indonesia terhadap vaksin COVID-19 di Twitter menggunakan BiLSTM dan fitur *Word Embedding* yaitu *FastText* dan *GloVe*. Kombinasi BLSTM dan *FastText* menghasilkan akurasi sebesar 75,76%. Kombinasi BLSTM dan *GloVe* menghasilkan akurasi sebesar 74,70%[9]. Studi-studi ini menunjukkan potensi untuk meningkatkan kinerja klasifikasi melalui eksperimen dengan berbagai metode dan fitur *machine learning.*

3 Metode Penelitian

Penelitian ini disusun dengan tahapan sistematis yang dirancang untuk memperoleh model klasifikasi sentimen yang optimal. Alur penelitian dirancang dalam bentuk kerangka kerja sebagaimana ditunjukkan pada *Gambar 1*, yang mencakup proses pengumpulan data, prapemrosesan, ekstraksi fitur menggunakan *word embedding*, pelatihan model *machine learning*, serta evaluasi kinerja



Gambar 1 Kerangka kerja penelitian

A. Dataset

Data penelitian ini diperoleh menggunakan library *Google Play Scraper* pada Python dengan objek aplikasi *Game Honor of Kings* (ID: **com.levelinfinite.sgameGlobal**). Proses *crawling* dilakukan dengan menetapkan parameter bahasa ulasan ke bahasa Indonesia (lang="id") dan negara ke Indonesia (country="id") untuk memastikan ulasan yang dikumpulkan relevan dengan persepsi pengguna dari wilayah tersebut. Pengambilan data dilakukan tanpa pembatasan skor, sehingga seluruh rentang penilaian (1–5 bintang) tetap disertakan agar distribusi sentimen yang dikumpulkan lebih representatif dan tidak bias terhadap sentimen tertentu. Ulasan diurutkan berdasarkan waktu unggah terbaru menggunakan parameter sort=Sort.NEWEST, dan total 900 ulasan berhasil dikumpulkan.

Meskipun demikian, proses pengumpulan data melalui *Google Play Store* memiliki beberapa potensi bias. Pertama, ulasan yang tersedia pada waktu *scraping* dapat dipengaruhi oleh tren sesaat, pembaruan aplikasi, atau isu tertentu yang sedang ramai, sehingga tidak selalu mencerminkan sentimen jangka panjang. Kedua, ulasan yang diambil berdasarkan bahasa dan negara tertentu berpotensi mengesampingkan opini dari kelompok pengguna lain yang memiliki gaya bahasa, preferensi, atau konteks penggunaan berbeda. Ketiga, tidak semua ulasan memiliki kualitas yang baik; beberapa mungkin berupa teks sangat pendek, tidak relevan, atau mengandung spam.

Untuk meminimalkan bias tersebut, penelitian ini menerapkan sejumlah pertimbangan kualitas data, termasuk menjaga keberagaman skor ulasan, memeriksa keberadaan duplikasi, dan memastikan

 ISSN:2302-8149
e-ISSN:2540-9719

setiap ulasan memiliki konten teks yang dapat diproses. Selain itu, hanya tiga atribut inti yang digunakan, yaitu:

1. *reviewId* — ID unik setiap ulasan
2. *content* — isi teks ulasan pengguna
3. *score* — skor bintang (1–5)

Pengendalian kualitas ini bertujuan memastikan bahwa dataset yang digunakan cukup representatif, relevan, dan mendukung analisis sentimen secara lebih objektif

B. *Preprocessing*

Setelah proses pengumpulan data, dilakukan tahapan prapemrosesan untuk memastikan data bersih, terstruktur, dan siap digunakan dalam proses klasifikasi sentimen. Tahapan prapemrosesan mencakup beberapa langkah penting, yaitu:

1. *Data Cleaning*. Membersihkan teks dari elemen yang tidak relevan seperti tanda baca berlebih, URL, emoji, dan karakter non-alfabet. Pengaruh terhadap model*:* pembersihan ini mengurangi *noise* yang dapat mengganggu pembelajaran pola oleh model dan membantu algoritma fokus pada informasi yang benar-benar bermakna.
2. *Case Folding*. Mengubah seluruh teks menjadi huruf kecil. Pengaruh terhadap model*:* langkah ini mencegah kata yang sama dianggap berbeda hanya karena perbedaan huruf besar/kecil (misalnya "*Good*" dan "*good*"), sehingga mengurangi redundansi dan meningkatkan konsistensi representasi fitur.
3. *Tokenization*. Memecah kalimat menjadi token atau kata-kata individual. Pengaruh terhadap model*:* tokenisasi memungkinkan model mengolah teks pada level paling mendasar sehingga proses ekstraksi fitur dari setiap kata dapat dilakukan dengan tepat.
4. *Stopword Removal*. Menghapus kata-kata umum seperti “dan”, “yang”, atau “ke”, yang tidak memberikan kontribusi signifikan terhadap penentuan sentimen. Pengaruh terhadap model: mengurangi dimensi fitur dengan membuang kata yang tidak informatif, sehingga algoritma dapat lebih fokus pada kata-kata bermuatan sentimen dan meningkatkan efisiensi perhitungan.
5. *Stemming*. Mengubah kata ke bentuk dasar (root word), misalnya “bermain”, “memainkan”, dan “pemain” menjadi “main”. Pengaruh terhadap model*:* menyatukan variasi kata ke bentuk dasar membantu model mengenali pola yang lebih konsisten dan mengurangi sparsitas data, terutama pada representasi *embedding* atau *vectorization.*
6. *Labeling*. Memberikan label sentimen berdasarkan skor bintang: skor rendah (1–2) direpresentasikan sebagai negatif, skor menengah (3) sebagai netral, dan skor tinggi (4–5) sebagai positif. Pengaruh terhadap model: labeling yang konsisten sangat penting karena menjadi dasar pembelajaran model dalam mengenali hubungan antara fitur teks dan kategori sentimen.

Secara keseluruhan, rangkaian prapemrosesan ini tidak hanya membersihkan data, tetapi juga meningkatkan kualitas representasi teks sehingga model *machine learning* dapat mempelajari pola sentimen dengan lebih efektif dan menghasilkan prediksi yang lebih akurat.

C. *Word Embedding*

Setiap kata direpresentasikan sebagai vektor numerik berdimensi rendah, yang menangkap detail semantik dari korpus teks yang luas. Kami menggunakan model yang telah dilatih sebelumnya untuk tiga teknik *Word Embedding*:

1.*GloVe*

GloVe (*Global Vectors for Word Representation*) bergantung pada ko-kemunculan dan faktorisasi matriks untuk menghasilkan vektor, membangun hubungan statistik antara kata-kata dengan membangun matriks besar ko-kemunculan kata. [10][11].

2.*Word2Vec*

Word2Vec membuat vektor berdasarkan kemunculan kata, memprediksi kata-kata di sekitarnya dari kata tertentu atau model *Bag-of-Words* memprediksi kata-kata dari konteks tertentu[12][13].

3.*FastText*

FastText merepresentasikan setiap kata sebagai kumpulan karakter *n-gram*, menangkap esensi kata-kata yang lebih pendek dan memahami awalan serta akhiran. Pendekatan ini memungkinkan *FastText* untuk menangani kata-kata yang tidak ada dalam data pelatihan dengan menguraikannya menjadi n-gram[14][15].

D.Learning Model

 ISSN:2302-8149
e-ISSN:2540-9719

Kami mengevaluasi kinerja tiga algoritma *machine learning* menggunakan fitur *Word Embedding* yaitu:

1.K-Nearest Neighbor

KNN mengklasifikasikan objek berdasarkan kedekatan titik data pelatihan [16][17] [18]. KNN tidak melibatkan fase pelatihan offline[19] . Sebaliknya, KNN menyimpan semua dokumen pelatihan dan menghitung jarak selama fase prediksi[20]. KNN menetapkan kelas berdasarkan tetangga terdekat dan kategorinya[21]**.**

2.Naïve Bayes

Naïve Bayes adalah pengklasifikasi probabilistik sederhana yang mengasumsikan independensi fitur [22][23]. Ini menyeimbangkan kinerja dengan efisiensi komputasi dan berkinerja baik dengan ukuran sampel kecil karena regularisasi yang melekat [24][25]. Namun, ia mengalami kesulitan dengan interaksi antar fitur.[26].

3.Random Forest

Random Forest adalah metode pembelajaran *ensemble* yang membangun beberapa pohon keputusan independen[27][28]. Setiap pohon memberikan suara pada kelas contoh uji, dan suara mayoritas menentukan prediksi akhir[29]. *Random Forest* mengatasi overfitting dengan menggabungkan beragam pohon yang dibuat melalui pemilihan fitur dan data acak[30].

4 Hasil dan Pembahasan

Bagian ini memberikan ikhtisar hasil dan pertimbangan yang berasal dari eksperimen yang dilakukan sesuai dengan kerangka kerja penelitian yang diuraikan pada bagian sebelumnya. Eksperimen ini berfokus pada penilaian data teks media sosial menggunakan berbagai metode *machine learning* dan fitur *word embedding*. Dengan validasi split 80:20. Pengujian yang dilakukan dalam penelitian ini meliputi pengujian *machine learning* dengan variasi *word embedding*. Jenis-jenis metode *machine learning* yang digunakan adalah: *Naive Bayes* (NB), *K-Nearest Neighbor* (KNN), dan *Random Forest* (RF). Untuk *word embedding*, kami menggunakan 3 fitur, yaitu: *Word2Vec*, *GloVe*, dan Fast Text.

Tabel 1 Hasil evaluasi *naive bayes* pada seluruh konfigurasi

Model	TP	FP	FN	TN	Accuracy	Precision	Recall	F1-Score	AUC
NB baseline	349	123	99	329	0.68	0.74	0.78	0.76	0.70
NB + Word2Vec	511	100	70	219	0.81	0.84	0.88	0.86	0.83
NB + GloVe	439	94	88	279	0.76	0.82	0.83	0.82	0.78
NB + FastText	469	79	69	283	0.84	0.86	0.87	0.86	0.86

Tabel 1 menyajikan evaluasi performa model *Naive Bayes* pada empat konfigurasi berbeda, yakni tanpa fitur *word embedding*, serta dengan tiga jenis *embedding: Word2Vec*, *GloVe*, dan *FastText*. Performa dinilai menggunakan *confusion matrix* (TP, FP, FN, TN) serta metrik umum evaluasi klasifikasi seperti akurasi, presisi, recall, F1-score, dan AUC.

Model *baseline* menghasilkan akurasi sebesar 0.68, dengan presisi 0.74 dan *recall* 0.78. Nilai *false positive* (123) dan *false negative* (99) yang relatif tinggi menunjukkan bahwa model belum mampu membedakan sentimen secara optimal. Hasil ini memperlihatkan keterbatasan *Naive Bayes* ketika hanya menggunakan representasi teks sederhana, karena konteks dan makna kata tidak ditangkap secara efektif.

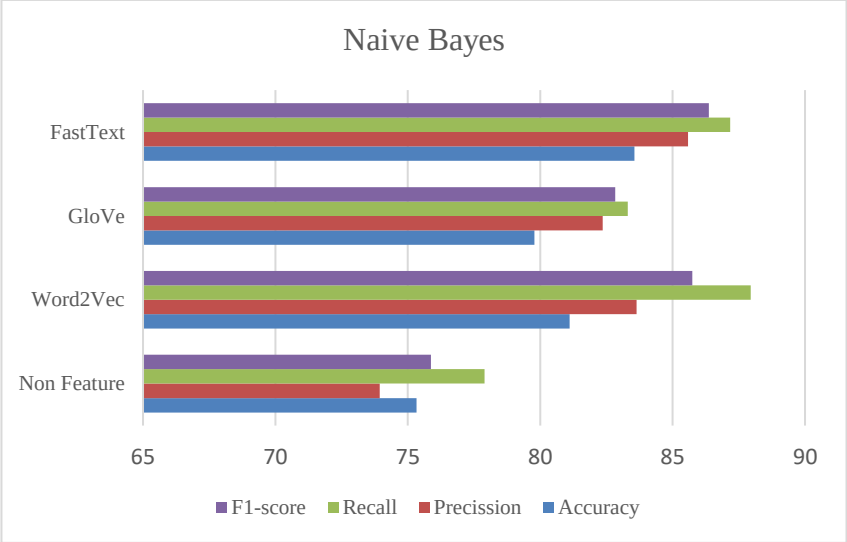
Penambahan fitur *Word2Vec* memberikan peningkatan performa yang signifikan. Akurasi meningkat menjadi 0.81, dengan F1-score 0.86 dan AUC 0.83. Jumlah TP meningkat drastis (511), sementara FN turun menjadi 70. Hal ini menunjukkan bahwa *Word2Vec* mampu menangkap hubungan semantik antar kata secara lebih baik, sehingga model lebih efektif mendeteksi sentimen positif maupun negatif. Word2Vec menjadi konfigurasi dengan performa terbaik kedua.

Penggunaan *embedding GloVe* juga meningkatkan performa dibanding *baseline*, namun tidak setinggi *Word2Vec* atau *FastText*. Akurasi tercatat 0.76 dengan F1-score 0.82. Nilai TP (439) dan FN (88) menunjukkan bahwa model cukup baik dalam mendeteksi sentimen positif, namun masih terdapat misklasifikasi yang lebih banyak dibanding konfigurasi lainnya. Hal ini dapat disebabkan

 ISSN:2302-8149
e-ISSN:2540-9719

oleh sifat *GloVe* yang sangat bergantung pada distribusi global, sehingga kurang fleksibel pada teks ulasan dengan variasi bahasa pengguna.

FastText memberikan performa tertinggi pada hampir seluruh metrik, dengan akurasi 0.84, F1-score 0.86, dan AUC 0.86. Nilai FP paling rendah (79) menunjukkan bahwa model lebih jarang memberikan prediksi positif yang salah. Selain itu, FN (69) juga merupakan yang terendah kedua. Kelebihan *FastText* dalam memanfaatkan *subword* membuat *embedding* lebih *robust* terhadap kata tidak baku, typo, atau variasi penulisan yang sering muncul dalam ulasan pengguna. Hal ini menjadikannya konfigurasi paling efektif untuk dataset ini.



Gambar 2 Perbandingan nilai evaluasi *naïve bayes* dengan dan tanpa fitur

Dari Gambar 2 terlihat bahwa penambahan fitur *embedding* berkontribusi nyata terhadap peningkatan seluruh metrik evaluasi. *Model Naive Bayes + FastText* menunjukkan nilai akurasi dan F1-score tertinggi, yang menandakan keseimbangan terbaik antara presisi dan recall. Hasil ini mengonfirmasi bahwa representasi semantik berbasis *subword* seperti FastText lebih efektif dibandingkan *Word2Vec* maupun *GloVe* dalam konteks analisis sentimen teks tidak terstruktur. Secara umum, dapat dikatakan bahwa *Naive Bayes* seringkali bekerja dengan baik pada data teks karena asumsi independensi kondisionalnya sesuai dengan model representasi kata (*Bag-of-Words atau* TF-IDF). Namun, ketika menggunakan fitur *Word Embedding*, *Naive Bayes* tidak sepenuhnya memanfaatkan informasi ini

Tabel 2 Hasil evaluasi KNN pada seluruh konfigurasi

Model	TP	FP	FN	TN	Accuracy	Precision	Recall	F1-Score	AUC
KNN baseline	469	99	93	239	0.74	0.83	0.83	0.83	0.75
KNN + Word2Vec	499	96	81	224	0.76	0.84	0.86	0.85	0.77
KNN + GloVe	491	98	76	235	0.77	0.83	0.87	0.85	0.78
KNN + FastText	**509**	**66**	**48**	**276**	**0.87**	**0.89**	**0.91**	**0.90**	**0.90**

Tabel 2 menampilkan evaluasi performa algoritma KNN pada empat konfigurasi: tanpa fitur embedding, serta menggunakan *Word2Vec*, *GloVe*, dan *FastText*. Evaluasi dilakukan menggunakan *confusion matrix* (TP, FP, FN, TN) serta metrik seperti akurasi, *presisi*, *recall*, *F1-Score*, dan AUC. Secara umum, hasil menunjukkan bahwa kombinasi KNN dengan fitur *word embedding* secara signifikan meningkatkan performa model dibandingkan KNN tanpa fitur.

Konfigurasi dasar KNN tanpa *embedding* mencapai akurasi 0.74 dan F1-score 0.83. Jumlah TP (469) dan FN (93) menunjukkan bahwa model mampu mengidentifikasi sentimen positif, namun masih terdapat sejumlah kesalahan prediksi, terutama karena keterbatasan representasi teks berbasis frekuensi. Keterbatasan ini membuat model sulit menangkap konteks dan makna kata secara semantik.

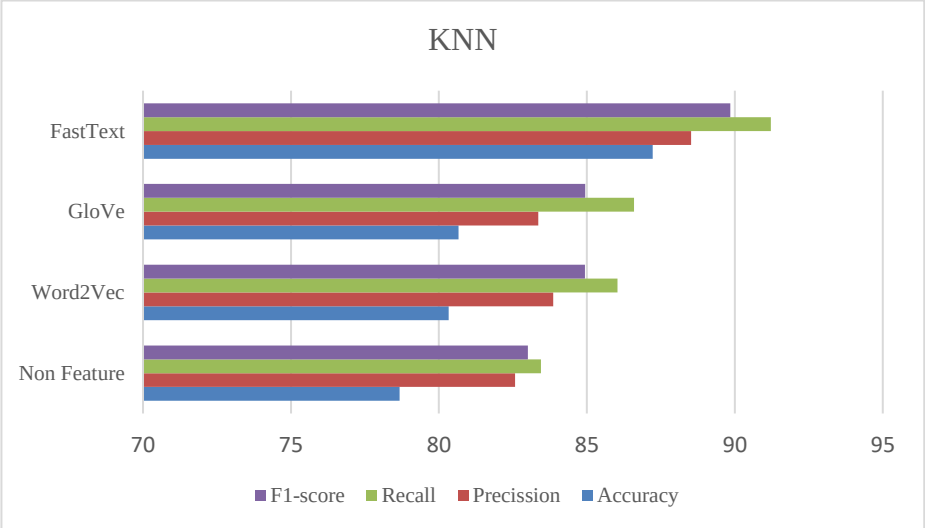
 ISSN:2302-8149
e-ISSN:2540-9719

Penggunaan *Word2Vec* meningkatkan akurasi menjadi 0.76 dengan *F1-score* 0.85. Peningkatan TP (499) dan penurunan FN (81) dibanding *baseline* menunjukkan bahwa *embedding* berbasis distribusi lokal ini membantu KNN lebih efektif mengukur kedekatan antar vektor ulasan. *Embedding Word2Vec* memberikan pemahaman semantik yang lebih baik, sehingga model lebih akurat dalam memprediksi sentimen.

KNN dengan *GloVe* memberikan akurasi 0.77 dan F1-score 0.85, sedikit lebih baik dibanding *Word2Vec*. Jumlah FN paling rendah kedua (76) menunjukkan bahwa model semakin kuat dalam mengidentifikasi sentimen positif. Keunggulan *GloVe* dalam memanfaatkan informasi statistik global membuat representasi kata lebih stabil, sehingga meningkatkan kualitas perhitungan jarak antar vektor ulasan pada KNN.

Konfigurasi KNN + *FastText* menunjukkan performa paling tinggi dalam seluruh metrik, dengan akurasi mencapai 0.87, *F1-score* 0.90, dan AUC 0.90. Nilai FP (66) dan FN (48) merupakan yang terendah, menandakan bahwa model memiliki keseimbangan prediksi yang sangat baik. Kelebihan FastText dalam representasi subword membuat embedding tahan terhadap variasi kata, typo, dan struktur morfologi, sehingga vektor yang dihasilkan jauh lebih informatif untuk perhitungan jarak KNN.

Gambar 3 menunjukkan bahwa penambahan fitur *word embedding* secara umum memberikan peningkatan akurasi dan keseimbangan antara presisi serta recall. Peningkatan paling signifikan terjadi pada model KNN + FastText yang mampu mencapai akurasi hingga 87,1%, menunjukkan efektivitas representasi berbasis *subword* dalam menangkap karakteristik linguistik teks ulasan pengguna**.**



Gambar 3 Perbandingan nilai evaluasi KNN dengan dan tanpa fitur

Secara umum, KNN bekerja dengan mencari jarak terpendek antar vektor fitur. Dengan fitur *Word Embedding*, KNN dapat memberikan hasil yang baik jika jarak antar vektor secara efektif memisahkan kelas-kelas. Namun, KNN dapat berjalan lambat dan kurang efisien pada data besar karena harus menghitung jarak ke semua titik dalam dataset pelatihan.

Tabel 3 Hasil evaluasi RF pada seluruh konfigurasi

Model	TP	FP	FN	TN	Accuracy	Precision	Recall	F1-Score	AUC
RF (tanpa fitur)	353	151	157	239	0.62	0.70	0.69	0.69	0.63
RF + Word2Vec	**459**	**79**	**71**	**291**	**0.85**	**0.85**	**0.87**	**0.86**	**0.88**
RF + GloVe	410	108	92	290	0.77	0.79	0.82	0.81	0.80
RF + FastText	408	138	126	228	0.69	0.75	0.76	0.75	0.70

Tabel 3 menyajikan hasil evaluasi algoritma *Random Forest* (RF) pada empat konfigurasi: tanpa embedding, serta menggunakan *Word2Vec*, *GloVe*, dan *FastText*. Hasil menunjukkan bahwa performa RF sangat dipengaruhi oleh kualitas representasi fitur, di mana *Word2Vec* memberikan peningkatan kinerja paling signifikan, sedangkan *FastText* tidak memberikan hasil optimal pada RF untuk kasus

 ISSN:2302-8149
e-ISSN:2540-9719

ini.

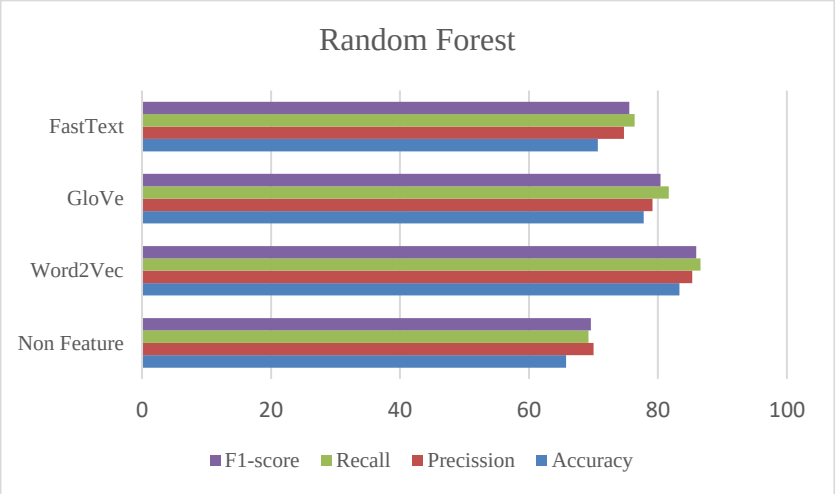
Random Forest tanpa embedding menghasilkan akurasi rendah, yaitu 0.62 dengan F1-score 0.69. Nilai FP (151) dan FN (157) cukup tinggi, menunjukkan bahwa model sering salah mengklasifikasi baik sentimen positif maupun negatif. Representasi teks tradisional yang digunakan pada baseline tidak mampu menangkap konteks kata secara memadai, sehingga berdampak pada ketidakstabilan prediksi antar pohon keputusan dalam ensemble.

Kombinasi RF dengan *Word2Vec* menghasilkan performa paling tinggi, dengan akurasi 0.85, F1-score 0.86, dan AUC 0.88. Nilai TP meningkat tajam menjadi 459 dan FN turun menjadi 71, menandakan bahwa model lebih konsisten dalam mengidentifikasi sentimen positif. *Word2Vec* memberikan vektor representasi yang padat dan stabil, sehingga fitur yang dimasukkan ke pohon keputusan lebih informatif dan memudahkan ensemble RF dalam membangun batas keputusan yang akurat.

RF dengan *GloVe* memberikan performa menengah, dengan akurasi 0.77 dan *F1-score* 0.81. Meskipun hasilnya lebih baik daripada baseline, peningkatannya tidak setinggi konfigurasi Word2Vec. Nilai FP (108) dan FN (92) menunjukkan bahwa meskipun *GloVe* mampu menangkap informasi statistik global, representasi yang dihasilkan belum sepenuhnya optimal untuk model pohon seperti RF yang sensitif terhadap skala dan kedalaman fitur.

Konfigurasi RF + *FastText* menghasilkan performa paling rendah di antara embedding lainnya, dengan akurasi 0.69 dan F1-score 0.75. Nilai FP (138) dan FN (126) relatif tinggi dibandingkan Word2Vec dan GloVe. Hal ini kemungkinan terjadi karena FastText menghasilkan vektor berbasis subword yang lebih halus dan kompleks sehingga membuat pohon keputusan dalam RF sulit menemukan batas keputusan yang jelas. Struktur vektor FastText yang lebih kaya justru kurang cocok untuk model berbasis pohon yang bekerja paling baik pada fitur yang lebih terpisah secara diskriminatif.

Berdasarkan Gambar 4 terlihat bahwa *Word2Vec* memberikan peningkatan performa paling besar dibandingkan ketiga fitur lainnya. Nilai akurasi mencapai 85,0% dengan F1-score sebesar 85,9%, menandakan keseimbangan yang baik antara presisi dan *recall*. Sebaliknya, *FastText* justru menurunkan performa RF karena mekanisme *bag-of-subwords* tidak terintegrasi secara optimal dalam proses pemilihan atribut pada ensemble decision tree.



Gambar. 4. Perbandingan nilai evaluasi *random forest* dengan dan tanpa fitur

Secara umum dapat dikatakan bahwa *Random Forest* cenderung memberikan kinerja yang lebih baik karena memanfaatkan sejumlah besar pohon keputusan dan fitur acak untuk mengurangi overfitting. Dengan fitur word embedding, *Random Forest* dapat menangkap lebih banyak interaksi antar fitur yang mungkin diabaikan oleh *Naive Bayes*.

Dari seluruh pengujian *machine learning* yang dilakukan, dapat dilihat bahwa algoritma KNN mencapai tingkat akurasi tertinggi sebesar 79,11%, sedangkan algoritma *Random Forest* menghasilkan akurasi terendah sebesar 66,33%, sebelum menyertakan fitur *word embedding* apa pun. Sementara itu, hasil akurasi tertinggi setelah menggunakan fitur *word embedding* diperoleh dari

 ISSN:2302-8149
e-ISSN:2540-9719

algoritma KNN dengan fitur *FastText* dengan nilai sebesar 87,55% dan akurasi terendah diperoleh dari algoritma *Random Forest* dengan fitur *FastText*. Setelah melihat seluruh nilai evaluasi kinerja klasifikasi, yaitu akurasi, presisi, recall dan f1-score, algoritma terbaik adalah algoritma KNN dengan hasil evaluasi yang stabil pada seluruh *word embedding*. Semua pengujian masih mentoleransi kesalahan *False Positive* dan *False Negative*. Semua algoritma masih menggunakan parameter asli. Hal ini dapat menjadi celah untuk penelitian lebih lanjut, seperti mengurangi nilai *False Positive* atau *False Negative*. Celah untuk meningkatkan akurasi juga dapat dicapai dengan menyetel semua hiperparameter.

Lebih lanjut, *Word Embedding* terbaik yang dihasilkan dalam eksperimen ini adalah *FastText* yang memberikan skor tertinggi pada algoritma KNN. Hal ini sangat sejalan dengan cara kerja masing-masing metode. Salah satu keunggulan *FastText* terletak pada kecepatan dan efisiensinya yang luar biasa. Desainnya memfasilitasi pelatihan cepat pada korpus besar, sehingga ideal untuk aplikasi waktu nyata dan kumpulan data skala besar. Selain itu, kemampuan untuk menghasilkan penyisipan untuk unit subkata secara signifikan meningkatkan kegunaannya, terutama dalam skenario yang melibatkan kata-kata langka atau tak terlihat. Atribut-atribut ini telah terbukti sangat berharga di berbagai lanskap linguistik dan domain tertentu.

Eksperimen ini mencakup pengujian *machine learning* dengan variasi *Word Embedding*. Metode *machine learning* yang digunakan untuk klasifikasi sentimen dalam penelitian ini adalah *Naive Bayes* (NB), K-Nearest Neighbor (KNN), dan *Random Forest* (RF). Untuk *Word Embedding*, kami menggunakan tiga fitur: *Word2Vec*, *GloVe*, dan *FastText*. Hasilnya menunjukkan bahwa penggabungan fitur *Word Embedding* secara signifikan meningkatkan kinerja model klasifikasi sentimen. *FastText* secara konsisten memberikan hasil terbaik di berbagai algoritma karena kemampuannya menangkap informasi subkata dan efisiensinya dalam menangani kumpulan data besar.

Penurunan performa *Random Forest* ketika menggunakan FastText terjadi melalui karakteristik internal FastText itu sendiri. *FastText* menghasilkan representasi berbasis *subword* yang kaya detail, sehingga setiap kata dibentuk dari gabungan *n-gram* karakter. Representasi yang sangat granular ini justru kurang optimal bagi algoritma berbasis pohon seperti *Random Forest*, karena mekanisme pemilihan fitur pada decision tree bekerja dengan menentukan threshold yang memisahkan nilai vektor secara jelas. Ketika vektor kata terlalu padat oleh informasi *subword*, model cenderung mengalami *over-segmentation* dan mempelajari pola yang tidak general (*overfitting*). Hal ini berbeda dengan *Word2Vec*, yang memberikan representasi lebih stabil dan lebih mudah ditangani oleh decision tree karena memodelkan konteks secara lokal. Secara keseluruhan, embedding berbasis *subword* seperti *FastText* lebih cocok untuk algoritma berbasis jarak seperti KNN, namun tidak selalu memberikan performa optimal pada tree-based model seperti *Random Forest*.

Meskipun terdapat peningkatan, semua model masih menunjukkan *False Positive* dan *False Negative*. Penelitian lebih lanjut dapat difokuskan pada pengurangan kesalahan ini dengan menyetel hiperparameter dan mengeksplorasi teknik praproses lanjutan. Selain itu, peningkatan kumpulan data pelatihan dengan sampel yang lebih beragam dapat semakin meningkatkan ketahanan model. Temuan ini menegaskan bahwa kecepatan dan kemampuan representasi subkata *FastText* membuatnya sangat efektif untuk tugas analisis sentimen, selaras dengan hasil yang diamati dalam studi ini.

5 Kesimpulan

Penelitian ini menunjukkan bahwa penerapan fitur *word embedding* berkontribusi signifikan dalam meningkatkan kinerja klasifikasi sentimen. Dari ketiga algoritma yang diuji, KNN dengan *FastText* memberikan hasil terbaik dengan akurasi 87,55%, sementara *Random Forest* dengan *FastText* menghasilkan performa terendah. Secara umum, *FastText* menjadi fitur *Word Embedding* paling efektif karena mampu merepresentasikan informasi sub-kata dan bekerja efisien pada data berukuran besar. Perbedaan ini disebabkan oleh karakteristik FastText yang menghasilkan representasi vektor berbasis subword yang padat dan sangat halus (*dense and smooth embeddings*). Representasi seperti ini sangat sesuai untuk algoritma berbasis jarak seperti KNN, tetapi kurang optimal untuk algoritma berbasis pohon keputusan seperti *Random Forest* yang lebih efektif menangani fitur bersifat diskrit dan tidak terlalu padat. Akibatnya, FastText tidak memberikan keuntungan yang sama pada *Random Forest* seperti pada model lain. Secara praktis, model terbaik

 ISSN:2302-8149
e-ISSN:2540-9719

dalam penelitian ini dapat diterapkan untuk sistem analisis sentimen pada ulasan gim, seperti pemantauan opini pemain secara real-time, deteksi isu gameplay, penilaian kepuasan pengguna, serta membantu pengembang dalam pengambilan keputusan berbasis data. Meskipun terjadi peningkatan pada sebagian besar konfigurasi, beberapa model masih menghasilkan false positive dan false negative yang cukup tinggi. Oleh karena itu, penelitian selanjutnya disarankan untuk melakukan penyetelan hiperparameter, menerapkan teknik prapemrosesan lanjutan, serta memperluas dan memperkaya dataset agar model mampu melakukan generalisasi dengan lebih baik.

Berdasarkan hasil penelitian ini, terdapat beberapa rekomendasi penerapan model analisis sentimen dalam pengembangan sistem ulasan game, khususnya untuk Honor of Kings dan game dengan karakteristik serupa yaitu : Pemantauan Sentimen *Real-Time: Model KNN + FastText* dapat dijadikan engine untuk dashboard analitik guna mendeteksi sentimen pemain terhadap update, hero, atau event tertentu. Analisis Kualitas Fitur & *Gameplay*: Informasi sentimen membantu *developer* mengidentifikasi masalah utama seperti *bug, matchmaking*, atau ketidak seimbangan hero. Peningkatan Layanan Pelanggan: Sistem dapat mengelompokkan keluhan secara otomatis sehingga tim support dapat memprioritaskan isu dengan sentimen negatif paling dominan. Rekomendasi Pengembangan Fitur: Sentimen positif dapat digunakan untuk mengetahui fitur yang paling disukai pemain dan menjadi dasar perencanaan update selanjutnya.

Ucapan Terima Kasih

Terima kasih kami ucapkan kepada Yayasan Dinamika Bangsa Jambi melalui Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Dinamika Bangsa atas dukungan biaya dan fasilitas dan menyelesaikan penelitian ini melalui program Hibah Penelitan Dosen Universitas Dinamika Bangsa dengan nomor 012/MOU/LPPM/UNAMA/IV/2025.

Referensi

[1] R. Fatmasari, V. Mega Ayu, B. Pratama, and W. Gata, "Analisis Sentimen dalam Pengkategorian Komentar Youtube terhadap Layanan Akademik dan Non-Akademik Universitas Terbuka untuk Prediksi Kepuasan," *Technol. SCI.*, Vol. 4, No. 2, pp. 395–404, 2022, DOI: 10.47065/bits.v4i2.1738.

[2] F. Panjaitan *et al.*, "Studi Komparatif *Algoritma Machine Learning* pada Analisis Sentimen Media Sosial," *JATI (Jurnal Mhs. Tek. Inform.*, Vol. 9, No. 2, pp. 3145–3152, 2025.

[3] D. A. S. Suhdi, "Integrasi Analisis Sentimen berbasis Aspek dan Intelijen Bisnis untuk Analisis Ulasan Pelanggan di Instagram dalam," *Karapan Netw. J.*, No. I, 2025.

[4] L. O. M. Y. Muhamad Djufri Rachim, "Analisis Sentimen Publik Terhadap Penggunaan Teknologi AI dalam Berita Politik dan Implikasinya Terhadap Pertumbuhan Ekonomi," *J. Online Progr. Stud. Pendidik. Ekon.*, Vol. 9, No. 4, pp. 1535–1551, 2024.

[5] E. H. Muktafin and P. Kusrini, "*Sentiments Analysis of Customer Satisfaction in Public Services using K-Nearest Neighbors Algorithm and Natural Language Processing Approach*," *Telkomnika (Telecommunication Comput. Electron. Control.*, Vol. 19, No. 1, pp. 146–154, 2021, DOI: https://doi.org/10.12928/TELKOMNIKA.V19I1.17417.

[6] H. A. Santoso, E. H. Rachmawanto, A. Nugraha, A. A. Nugroho, D. R. I. M. Setiadi, and R. S. Basuki, "*Hoax Classification and Sentiment Analysis of Indonesian News using Naive Bayes Optimization*," *Telkomnika (Telecommunication Comput. Electron. Control.*, Vol. 18, No. 2, pp. 799–806, 2020, DOI: https://doi.org/10.12928/TELKOMNIKA.V18I2.14744.

[7] M. A. Fauzi, "*Random Forest Approach fo Sentiment Analysis in Indonesian Language*," *Indones. J. Electr. Eng. Comput. SCI.*, Vol. 12, No. 1, pp. 46–50, 2018, DOI: https://doi.org/10.11591/ijeecs.v12.i1.pp46-50.

[8] A. Basuki, "*Sentiment Analysis of Service Provider on Twitter Tweet using Naive Bayes Classifier*," *J. Ilm. Tek. Elektro Komput. dan Inform.*, Vol. 5, No. 2, pp. 13–23, 2023, DOI: https://doi.org/10.47080/iftech.v5i2.2752.

[9] K. K. Agustiningsih, E. Utami, and O. M. A. Alsyaibani, "*Sentiment Analysis and Topic Modelling of the COVID-19 Vaccine in Indonesia on Twitter Social Media using Word Embedding*," *J. Ilm. Tek. Elektro Komput. dan Inform.*, Vol. 8, No. 1, p. 64, 2022, DOI: https://doi.org/10.26555/jiteki.v8i1.23009.

 ISSN:2302-8149
e-ISSN:2540-9719

[10] A. George, H. B. Barathi Ganesh, M. Anand Kumar, and K. P. Soman, *Significance of Global Vectors Representation in Protein Sequences Analysis*, Vol. 31. Springer International Publishing, 2019. DOI: https://doi.org/10.1007/978-3-030-04061-1_27.

[11] N. Badri, F. Kboubi, and A. H. Chaibi, "*Combining FastText and Glove Word Embedding for Offensive and Hate Speech Text Detection*," *Procedia Comput. SCI.*, Vol. 207, No. Kes, pp. 769–778, 2022, DOI: https://doi.org/10.1016/j.procs.2022.09.132.

[12] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "*Word2vec Model Analysis for Semantic Similarities in English Words*," *Procedia Comput. SCI.*, Vol. 157, pp. 160–167, 2019, DOI: https://doi.org/10.1016/j.procs.2019.08.153.

[13] R. Rahmanda and E. B.Setiawan, "*Word2Vec on Sentiment Analysis with Synthetic Minority Oversampling Technique and Boosting Algorithm*," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, Vol. 6, No. 4, pp. 599–605, 2022, DOI: 10.29207/resti.v6i4.4186.

[14] I. N. Khasanah, "*Sentiment Classification using FastText Embedding and Deep Learning Model*," *Procedia CIRP*, Vol. 189, pp. 343–350, 2021, DOI: https://doi.org/10.1016/j.procs.2021.05.103.

[15] M. A. Raihan and E. B. Setiawan, "*Aspect based Sentiment Analysis with FastText Feature Expansion and Support Vector Machine Method on Twitter*," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, Vol. 6, No. 4, pp. 591–598, 2022, DOI: https://doi.org/10.29207/resti.v6i4.4187.

[16] R. Chivukula, T. Jaya Lakshmi, S. S. Uday, and S. T. Pavani, "*Classifying Clinically Actionable Genetic Mutations using KNN and SVM*," *Indones. J. Electr. Eng. Comput. SCI.*, Vol. 24, No. 3, pp. 1672–1679, 2021, DOI: https://doi.org/10.11591/ijeecs.v24.i3.pp1672-1679.

[17] M. S. Islam *et al.*, "*Machine Learning-based Music Genre Classification with Pre-Processed Feature Analysis*," *J. Ilm. Tek. Elektro Komput. dan Inform.*, Vol. 7, No. 3, p. 491, 2022, DOI: https://doi.org/10.26555/jiteki.v7i3.22327.

[18] J. Zhang, Y. Li, F. Shen, Y. He, H. Tan, and Y. He, "*Hierarchical Text Classification with Multi-Label Contrastive Learning and KNN*," *Neurocomputing*, Vol. 577, No. January, p. 127323, 2024, DOI: https://doi.org/10.1016/j.neucom.2024.127323.

[19] L. V. Nguyen, Q. T. Vo, and T. H. Nguyen, "*Adaptive KNN-based Extended Collaborative Filtering Recommendation Services*," *Big Data Cogn. Comput.*, Vol. 7, No. 2, 2023, DOI: https://doi.org/10.3390/bdcc7020106.

[20] M. Nadeem *et al.*, "*Preventing Cloud Network from Spamming Attacks using Cloudflare and KNN*," *Comput. Mater. Contin.*, Vol. 74, No. 2, pp. 2641–2659, 2023, DOI: https://doi.org/10.32604/cmc.2023.028796.

[21] Y. Yang and X. Liu, "*A Re-Examination of Text Categorization Methods*".

[22] D. Berrar, "*Bayes' Theorem and Naive Bayes Classifier*," *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma.*, Vol. 1–3, No. 2018, pp. 403–412, 2018, DOI: 10.1016/B978-0-12-809633-8.20473-1.

[23] J. K. Alwan, D. S. Jaafar, and I. R. Ali, "*Diabetes Diagnosis System using Modified Naive Bayes Classifier*," *Indones. J. Electr. Eng. Comput. SCI.*, Vol. 28, No. 3, pp. 1766–1774, 2022, DOI: https://doi.org/10.11591/ijeecs.v28.i3.pp1766-1774.

[24] S. Wang, J. Ren, and R. Bai, "*A Semi-Supervised Adaptive Discriminative Discretization Method Improving Discrimination Power of Regularized Naive Bayes*," *Expert Syst. Appl.*, Vol. 225, No. April, p. 120094, 2023, DOI: 10.1016/j.eswa.2023.120094.

[25] C. J. Anderson *et al.*, "*A Novel Naïve Bayes Approach to Identifying Grooming Behaviors in the Force-Plate Actometric Platform*," *J. Neurosci. Methods*, Vol. 403, No. July 2023, p. 110026, 2024, DOI: 10.1016/j.jneumeth.2023.110026.

[26] M. Badar and M. Fisichella, "*Fair-CMNB: Advancing Fairness-Aware Stream Learning with Naïve Bayes and Multi-Objective Optimization*," *Big Data Cogn. Comput.*, Vol. 8, No. 2, 2024, DOI: https://doi.org/10.3390/bdcc8020016.

[27] A. A. Khaleel, A. A. M. Al-Azzawi, and A. M. Alkhazraji, "*Random Forest for Lung Cancer Analysis using Apache Mahout and Hadoop based on Software Defined Networking*," *Indones. J. Electr. Eng. Comput. SCI.*, Vol. 32, No. 2, pp. 1086–1093, 2023, DOI: https://doi.org/10.11591/ijeecs.v32.i2.pp1086-1093.

 ISSN:2302-8149
e-ISSN:2540-9719

[28] T. A. Assegie, R. Subhashni, N. K. Kumar, J. P. Manivannan, P. Duraisamy, and M. F. Engidaye, "*Random Forest and Support Vector Machine-based Hybrid Liver Disease Detection*," *Bull. Electr. Eng. Informatics*, Vol. 11, No. 3, pp. 1650–1656, 2022, DOI: https://doi.org/10.11591/eei.v11i3.3787.

[29] A. Sekulić, M. Kilibarda, G. B. M. Heuvelink, M. Nikolić, and B. Bajat, "*Random Forest Spatial Interpolation*," *Remote Sens.*, Vol. 12, No. 10, pp. 1–29, 2020, DOI: https://doi.org/10.3390/rs12101687.

[30] H. Syahputra and A. Wibowo, "*Comparison of Support Vector Machine (SVM) and Random Forest Algorithm for Detection of Negative Content on Websites*," *J. Ilm. Tek. Elektro Komput. dan Inform.*, Vol. 9, No. 1, pp. 165–173, 2023, DOI: https://doi.org/10.26555/jiteki.v9i1.25861.