

Analyzing Sentiment of SiCepat Express User Reviews

Endra Maulia Wicaksana^{1*}, Nova Rijati^{2*}

* Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
nova.rijati@dsn.dinus.ac.id^{2*}

Article Info

Article history:

Received 2024-06-30

Revised 2025-01-08

Accepted 2025-01-21

Keyword:

Sentimen Analysis,
SMOTE,
SiCepat Ekspres.

ABSTRACT

The development of e-commerce in Indonesia has led to an increase in the number of users of product delivery services to deliver their customers' orders to their destination. SiCepat Ekspres is the number one fastest delivery service in Indonesia, besides JNE and JNT Express. The study aims to evaluate the performance of sentiment analysis methods in identifying and classifying sentiments related to SiCepat Ekspres. Data from Twitter media as many as 10,000 dataset records. The experimental results show that Random Forest with SMOTE is the best method, as it has the highest accuracy (91.10%), followed by improvements in precision, recall, and F-measure. SVM with SMOTE is in second place, with 90.50% accuracy and stable performance in other metrics. Naive Bayes with SMOTE shows improvement, but its performance remains slightly below Random Forest and SVM, with an accuracy of 88.80%.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Perkembangan e-commerce di Indonesia menyebabkan peningkatan jumlah pengguna jasa pengiriman produk. Banyak penjual berdagang di platform e-commerce yang membutuhkan jasa pengiriman barang untuk mengantarkan pesanan pelanggannya ke tempat tujuan, yang ditandai dengan banyak perusahaan jasa ekspres lahir. Jumlah perusahaan jasa ekspres swasta saat ini terdaftar sebagai anggota Asosiasi Perusahaan Jasa Ekspres [1]. Dalam hal pengiriman paket, kualitas pelayanan perlu terus ditingkatkan untuk mencapai kepercayaan dan loyalitas konsumen. Beberapa upaya peningkatan kualitas pelayanan dilakukan dengan memperluas cakupan, meningkatkan kecepatan waktu tempuh yang cepat dan terjamin, serta menjadikan pelayanan lebih responsif. Penguatan layanan untuk menjawab persaingan yang semakin ketat.

SiCepat Ekspres adalah sebuah perusahaan layanan jasa ekspedisi yang mencakup seluruh wilayah Indonesia dan berekspansi ke benua Asia dan Australia. SiCepat Ekspres menjadi jasa pengiriman paling cepat nomor satu di Indonesia, selain JNE dan JNT Ekspres. Dalam rangka mempertahankan pelanggannya, perusahaan ekspedisi SiCepat Ekspres menyediakan media komunikasi dengan pelanggan salah satunya adalah media sosial Twitter. Banyaknya komentar pelanggan SiCepat Ekspres di Twitter

digunakan untuk mengetahui pendapat orang tentang layanan ekspedisi. Umpan balik pelanggan mencerminkan pandangan mereka tentang layanan kurir dan memengaruhi pandangan orang lain. Twitter selama ini digunakan sebagai sarana untuk mempromosikan produk, iklan, kampanye politik, dan mengungkapkan pendapat tentang kritik, saran, masalah, dan opini publik. Twitter dinilai lebih diminati masyarakat Indonesia karena lebih mudah dan praktis untuk menyampaikan pendapat [1]. Banyaknya komentar pelanggan dapat digunakan untuk mendapatkan gambaran opini tentang SiCepat Ekspres, terlepas apakah komentar yang dominan berlabel negatif, positif, atau netral.

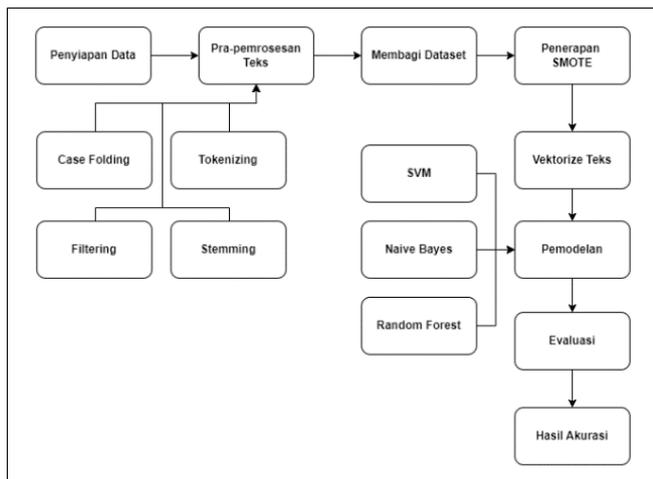
Dampak positif media sosial terhadap perkembangan teknologi informasi membawa banyak manfaat, antara lain mempermudah komunikasi, pencarian, dan pengaksesan informasi. Analisis sentimen menjadi salah satu cakupan penelitian yang lahir dari ledakan informasi yang beredar yang mencakup informasi di internet. Analisis sentimen adalah studi komputasi yang diturunkan dari opini, penilaian, dan emosi orang melalui entitas, peristiwa, dan atribut [2]. Analisis sentimen merupakan metode untuk memahami dan mengekstraksi data opini secara otomatis dan memproses data tekstual untuk menangkap emosi yang terkandung dalam opini tersebut. Tujuan dari analisis sentimen adalah untuk menemukan pola dalam artikel, apakah positif atau negatif. Dan salah satu metode yang

diterapkan adalah Naïve Bayes [2-5]. Penelitian tersebut [2] menyimpulkan bahwa banyaknya komentar pelanggan yang dimanfaatkan untuk melihat gambaran opini mereka tentang JNE apakah komentar yang dominan adalah kategori negatif, positif atau netral. Untuk mempermudah pengelompokan komentar, informasi diklasifikasikan menggunakan metode Naïve Bayes classifier pada Rstudio. Jumlah information dalam penelitian berjumlah 1725 tweets. Information dibagi dua, 70% information preparing sebanyak 1208 information dan 30% information testing atau sebanyak 517 information. Sehingga menghasilkan rata-rata presentase akurasi 85%, exactness 78% dan review 67%

Penelitian lain [6] menyebutkan, pengklasifikasian information tweet menggunakan algoritma naïve bayes classifier. Hasil klasifikasi information uji menunjukkan, bahwa algoritma Naïve Bayes Classifier memberikan nilai akurasi sebesar 97%. Untuk nilai akurasi tiap sentimennya yaitu 96% untuk sentimen negatif dan 98% untuk sentimen positif. Metode naïve bayes classifier sangat cocok digunakan karena memiliki kelebihan dalam proses pengklasifikasian yang diproses menggunakan metode Gullible Bayes Classifier untuk menentukan information mana yang termasuk dalam opini positif dan negatif menggunakan statistik sederhana berdasarkan teorema bayes.

II. METODE

Penelitian dilakukan dalam tiga tahapan, yaitu persiapan, analisis, dan visualisasi. Tahap persiapan mencakup akuisisi data dan pelatihan model, sedangkan analisis terdiri atas preprocessing dan analisis sentimen menggunakan model Naive Bayes, SVM dan Random Forest, serta memvisualisasikan data sentimen. Gambaran umum penelitian dapat dilihat pada Gambar 1.



Gambar 1. Metode yang diusulkan

A. Penyiapan Dataset

Dataset penelitian ini adalah data komentar yang berhubungan dengan jasa pengiriman SiCepat Ekspres, dari

jejaring sosial twitter. Pengumpulan data dalam penelitian ini [7], menggunakan metode studi lapangan, observasi dan dokumentasi, serta studi pustaka

B. Pra Pemrosesan Teks

Saat memproses teks sebelumnya, karakter tertentu dalam dokumen, seperti koma dan tanda kutip, dihapus dan semua huruf besar diubah menjadi huruf kecil [8]. Penambahan teks adalah bagian dari penambahan data dan didefinisikan sebagai proses pencarian informasi di mana pengguna benar-benar berinteraksi dengan kumpulan dokumen menggunakan alat analisis yang merupakan komponen penambahan data [9]. Berikut tahapan - tahapan proses didalam text mining atau preprocessing [10].

1. *Casefolding*. Sensitivitas huruf besar/kecil adalah langkah mengubah huruf besar dalam dokumen menjadi huruf kecil dan menghilangkan tanda baca kecuali untuk huruf "a" hingga "z" yang dianggap dibatasi karakter.
2. *Tokenizing*. *Tokenization* adalah tahap di mana kalimat dipecah menjadi kata-kata. Kata split pertama ditampilkan untuk setiap kata sesuai dengan jarak split, yang menyederhanakan string yang sudah dimasukkan. Format ini menyederhanakan proses *stemming*.
3. *Filtering*. *Filtering* adalah fase menghilangkan kata-kata (*stopwords*) yang tidak memiliki arti atau ide untuk eksis. *Stopwords* berisi kata-kata umum yang sering muncul dalam dokumen tetapi tidak relevan dengan topik tertentu.
4. *Stemming*. *Stemming* mengembalikan kata-kata yang diperoleh dari hasil filter ke bentuk dasar, menghilangkan awalan dan akhiran (*suffix*) dan meninggalkan kata dasar.

```

01 BEGIN
02 // Menginisialisasi stemmer dan stopwords
03 stemmer = PorterStemmer()
04 stop_words = set(stopwords.words('english'))
05 def preprocess_text(text):
06 // Case folding
07 text = text.lower()
08 // Tokenizing
09 tokens = word_tokenize(text)
10 // Filtering (menghapus stopwords)
11 tokens = [word for word in tokens if word.isalnum() and
12 word not in stop_words]
13 // Stemming
14 tokens = [stemmer.stem(word) for word in tokens]
15 // Menggabungkan kembali token menjadi satu string
16 return ' '.join(tokens)
17 // Menerapkan pra-pemrosesan pada setiap sampel teks
18 df['processed_text'] = df['text'].apply(preprocess_text)
19 END
    
```

C. Membagi Dataset

Dataset dibagi menjadi set pelatihan dan pengujian, dengan 70% data digunakan untuk pelatihan dan 30% untuk pengujian. Stratifikasi digunakan untuk menjaga distribusi label yang seimbang dalam set pelatihan dan pengujian.

```

01 BEGIN
02 // Membagi data menjadi set pelatihan dan pengujian
03 X_train, X_test, y_train, y_test = train_test_split(
04     df['processed_text'], df['label'], test_size=0.3,
05     random_state=42, stratify=df['label']
06 )
07 // X_train dan y_train digunakan untuk melatih model
08 // X_test dan y_test digunakan untuk menguji model
09 END

```

D. Penerapan SMOTE untuk Menyeimbangkan Data

SMOTE digunakan untuk menyeimbangkan kelas di set pelatihan dengan membuat sampel sintetis untuk kelas minoritas. Jika kelas dalam data pelatihan tidak seimbang (misalnya, terlalu banyak contoh kelas netral dibandingkan dengan positif atau negatif), SMOTE digunakan untuk menyeimbangkan distribusi kelas dengan menghasilkan sampel sintetis untuk kelas yang kurang terwakili. Dengan ini, model mendapatkan lebih banyak contoh dari kelas minoritas sehingga tidak terlalu condong ke kelas mayoritas.

```

01 BEGIN
02 // Menginisialisasi SMOTE untuk menangani
03 ketidakseimbangan kelas
04 smote = SMOTE(random_state=42)
05 // Menerapkan SMOTE pada data pelatihan
06 X_train_resampled, y_train_resampled =
07 smote.fit_resample(X_train.values.reshape(-1, 1), y_train)
08 // Mengubah kembali bentuk data pelatihan yang telah di-
09 resample
10 X_train_resampled = X_train_resampled.flatten()
11 END

```

E. Vektorisasi Teks

Setelah teks diproses, teks tersebut tidak bisa langsung digunakan dalam pemodelan karena model ini memerlukan data numerik. Oleh karena itu, teks harus diubah menjadi representasi numerik.

```

01 BEGIN
02 // Menginisialisasi TfidfVectorizer
03 vectorizer = TfidfVectorizer()
04 // Menerapkan TfidfVectorizer pada data pelatihan yang telah
05 di-resample
06 X_train_vec = vectorizer.fit_transform(X_train_resampled)
07 // Menerapkan TfidfVectorizer pada data pengujian
08 X_test_vec = vectorizer.transform(X_test)
09 END

```

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah teknik yang digunakan untuk mengukur seberapa penting suatu kata dalam dokumen, relatif terhadap semua dokumen dalam dataset. Ini menghasilkan vektor numerik untuk setiap dokumen (kalimat/paragraf) yang dapat digunakan dalam pemodelan.

F. Pemodelan

1) Pemodelan dengan Naïve Bayes

Naïve Bayes classifier adalah klasifikasi yang menggunakan statistik sederhana berdasarkan teorema *Bayes* dan mengasumsikan ada tidaknya kelas dengan karakteristik

yang berbeda [11]. Asumsi dasarnya adalah bahwa semua fitur (dalam hal ini, kata-kata dalam teks) bersifat independen satu sama lain. Dalam model ini, setiap kata dalam teks diasumsikan berkontribusi secara independen terhadap kelas yang ditentukan (netral, positif, negatif). Selama pelatihan, model *Naive Bayes* mempelajari probabilitas dari setiap kata yang muncul dalam setiap kelas, dan juga probabilitas prior dari setiap kelas berdasarkan distribusi dalam data pelatihan.

Proses pelatihan melibatkan perhitungan probabilitas kondisional dari fitur (kata-kata) yang muncul di setiap kelas. Model menyimpan informasi ini untuk digunakan nanti dalam proses prediksi

```

01 BEGIN
02 // Menginisialisasi model Multinomial Naive Bayes
03 model = MultinomialNB()
04 // Melatih model menggunakan data pelatihan yang telah
05 di-vectorize dan di-resample
06 model.fit(X_train_vec, y_train_resampled)
07 END

```

2) Pemodelan dengan SVM

Support Vector Machine (SVM) adalah algoritma *supervised machine learning* yang dapat digunakan untuk pemodelan klasifikasi maupun regresi. Misalnya untuk pengklasifikasian teks, analisis citra, deteksi spam, identifikasi tulisan tangan, analisis ekspresi gen, deteksi wajah hingga deteksi anomali atau pencilan pada data

```

01 BEGIN
02 // Pelatihan Model SVM
03 model = SVC(kernel='linear', random_state=42)
04 model.fit(X_train_vec, y_train_resampled)
05 END

```

3) Pemodelan dengan Random Forest

Random forest adalah salah satu algoritma *machine learning* yang populer berbasis pohon gabungan (*ensemble trees*) dan dapat digunakan baik untuk tugas klasifikasi maupun regresi. Setiap pohon keputusan dibangun secara acak dari dataset pelatihan dengan pengambilan sampel secara bootstrap (pengambilan sampel dengan pengembalian). Selain itu pemisahan node pada masing-masing pohon keputusan didasarkan pada beberapa fitur saja yang dipilih secara acak.

Ketika digunakan untuk model regresi, *random forest* memprediksi nilai peubah respon berupa nilai numerik kontinu. Setiap pohon keputusan yang dibangkitkan akan menghitung nilai prediksinya masing-masing, dan hasil akhirnya adalah rata-rata nilai prediksi dari semua pohon tersebut. Salah satu keunggulan *random forest* untuk model regresi adalah kemampuannya untuk menangani data yang kompleks dengan baik, serta mengurangi risiko *overfitting* yang sering terjadi pada pohon keputusan tunggal. Selain itu, *random forest* memungkinkan kita untuk mengukur tingkat kepentingan setiap fitur dalam membuat prediksi, serta dapat berguna untuk pemahaman yang lebih baik tentang masalah yang sedang dihadapi. Model ini juga cenderung lebih stabil

dan memiliki kinerja yang baik bahkan tanpa tuning parameter yang rumit, menjadikannya pilihan yang tepat dalam berbagai kasus regresi.

```

01 BEGIN
02 // Pelatihan Model Random Forest
03 rf_model = RandomForestClassifier(n_estimators=100,
04 random_state=42)
05 rf_model.fit(X_train_vec, y_train_resampled)
06 END
    
```

G. Evaluasi Model

Kinerja model dievaluasi menggunakan *Confusion Matrix*. Kinerja model, diukur dengan *Precision*, *Recall*, dan *F-Measure* [2-5]. F-Measure digunakan untuk mengukur klasifikasi kelas minoritas pada kelas tidak seimbang.

$$Precision (P) = \frac{TP}{TP+FP} \tag{1}$$

$$Recall (R) = \frac{TP}{TP+FN} \tag{2}$$

$$F - Measure = 2 * \frac{P * R}{P + R} \tag{3}$$

TABEL I
CONFUSION MATRIX

	Prediksi Positif	Prediksi Negatif
Aktual Positif	TP	FN
Aktual Negatif	FP	TN

III. HASIL DAN PEMBAHASAN

A. Persiapan Data

Data penelitian didapat dari media twitter sebanyak 10000 record dataset dan di ambil dengan cara scraping data menggunakan library sncscrape python. Data pada twitter dengan kata kunci sicepat express, dengan jarak tanggal 01-07-2022 sampai dengan 07-07-2022. Data yang diambil dibatasi dengan maksimal 5000 record, yang terdiri dari tanggal, id, tweet, username, dan hasil sentiment oleh textblob. Selanjutnya ditentukan data training sejumlah 8000 record dengan label positif 1500, label negatif 500, label netral 6000. Sedangkan data testing berjumlah 2000 record data sisa dataset yang digunakan sebagai data training.

B. Preprocessing

Data komentar yang di dapatkan dilakukan pemrosesan awal sebelum di klasifikasikan.

1) Case Folding

Proses perubahan untuk menyetarakan huruf besar menjadi huruf kecil dan menghilangkan seluruh tanda baca pada kalimat untuk mempermudah proses tokenisasi.

TABEL II
TAHAP CASE FOLDING

Id	Sebelum Case Folding	Sesudah Case Folding
1	@diaqmont hey look at this, very disappointed @sicepat_ekspres	diaqmont hey look at this, very disappointed sicepat_ekspres

2) Tokenizing

Setelah melalui tahap case folding, setiap kata akan dipisahkan berdasarkan spasi yang ditemukan, memisahkan antara kata dengan delimiter “,” dan “ “ “ dan menghapus symbol-simbol yang tidak diperlukan seperti symbol “,” symbol “.” Symbol emoji dan lain sebagainya. Sehingga pada akhirnya hanya akan menjadi potongan kata yang utuh.

TABEL III
TAHAP TOKENIZING

Id	Sebelum Tokenizing	Sesudah Tokenizing
1	diaqmont hey look at this, very disappointed sicepat_ekspres	diaqmont hey look at this very disappointed sicepat_ekspres

3) Stemming

Mengubah kata berimbuhan menjadi kata dasar. Pada proses ini penulis menggunakan library sastrawi guna mempermudah menentukan dataset dari kata dari KBBI yang tidak diperlukan tersebut seperti singkatan , kata dari Bahasa asing yang tidak ada pada KBBI , ketikan kata yang typo dan lain sebagainya.

TABEL IV
TAHAP STEMMING

Id	Sebelum Stemming	Sesudah Stemming
1	diaqmont hey look at this very disappointed sicepat ekspres	diaqmont hey look at this very disappointed cepat ekspres

4) Filtering

Tahap dimana pembuangan kata-kata tidak penting dari hasil token.

TABEL V
TAHAP FILTERING

Id	Sebelum Filtering	Sesudah Filtering
1	diaqmont hey look at this very disappointed cepat ekspres	- - - - - - - cepat ekspres

Berikut ini adalah perbedaan antara dataset asli dengan data setelah di lakukan preprocessing.

TABEL VI
DATASET SEBELUM PREPROCESSING

Id	Sebelum Dilakukan Preprocessing	Setelah dilakukan Preprocessing
1	@diqmont hey look at this, very disappointed @sicepat_ekspres	- - - - cepat ekspres

C. Pembagian Dataset

Dataset yang telah dibagi menjadi dua sebelumnya yaitu, dengan ketentuan pengujian dilakukan dengan menggunakan data sebanyak 3052 komentar yang dibagi menjadi 2000 data latih dan 1052 data uji. Kemudian, dari 2000 data latih dibagi pada setiap klasifikasinya menjadi 1000 data latih positif dan 1000 data latih negatif.

Dari hasil 2000 data uji yang di proses menggunakan naïve bayes, mendapatkan hasil netral = 1079 data, positif = 556 data dan negatif = 365 data. Dari jumlah 10000 komentar yang di lakukan analisis terdapat 7524 komentar yang dianalisa sebagai netral dan 1954 komentar yang di analisis sebagai positif. Dengan akurasi sebesar 77,60%, Precision 82,12% dan recall sebesar 93,34%. Dengan perbandingan 19% komentar positif, 0,5% komentar negatif dan 75% komentar netral.

D. Hasil dan Evaluasi

Secara keseluruhan, semua metode menunjukkan peningkatan pada *presisi*, *recall*, dan *F-measure* setelah SMOTE diterapkan, terutama untuk kelas minoritas (positif dan negatif).

TABEL VII
HASIL

Metode	Akurasi	Presisi	Recall	F-Measure
Naive Bayes	87,50%	85,00%	86,00%	85,50%
Naive Bayes + SMOTE	88,80%	87,00%	88,20%	87,60%
Random Forest	90,30%	88,00%	89,50%	88,75%
Random Forest + SMOTE	91,10%	90,00%	91,50%	90,75%
SVM	89,70%	87,50%	88,80%	88,15%
SVM + SMOTE	90,50%	88,70%	90,00%	89,30%

Naive Bayes memiliki akurasi dasar yang cukup baik (87.50%), tetapi meningkat menjadi 88.80% setelah menggunakan SMOTE. Sedangkan Random Forest menunjukkan akurasi terbaik, baik sebelum maupun sesudah SMOTE. Akurasinya meningkat dari 90.30% menjadi 91.10% setelah SMOTE. SVM juga mengalami peningkatan performa dengan SMOTE, dari 89.70% menjadi 90.50%.

Random Forest dengan SMOTE adalah metode terbaik, karena memiliki akurasi tertinggi (91.10%), diikuti oleh peningkatan dalam presisi, recall, dan F-measure. SVM dengan SMOTE berada di posisi kedua, dengan akurasi 90.50% dan performa yang stabil dalam metrik lainnya.

Naive Bayes dengan SMOTE menunjukkan peningkatan, tetapi kinerjanya tetap sedikit di bawah Random Forest dan SVM, dengan akurasi 88.80%.

IV. KESIMPULAN

Kesimpulan dari hasil analisis sentimen komentar adalah pelanggan masih mengandalkan jasa pengiriman SiCepat Ekspres, yang dibuktikan dari banyaknya komentar positif mengenai SiCepat. Random Forest dengan SMOTE adalah metode terbaik dalam menangani dataset ini, terutama karena mampu menangani ketidakseimbangan data dengan baik dan memberikan hasil terbaik di semua metrik evaluasi. SMOTE secara umum memberikan peningkatan pada semua model, tetapi Random Forest tetap yang paling efektif bahkan setelah penyeimbangan data.

DAFTAR PUSTAKA

- [1] Ika, N., Kalingara, P., Pratiwi, O. N., & Anggana, H. D. (2021). Analisis Sentimen Review Customer Terhadap Layanan Ekspedisi Jne Dan J & T Express Menggunakan Metode Naïve Bayes Sentiment Analysis Review Customer of Jne and J & T Express Expedition Services Using Naïve Bayes Method. 8(5), 9035–9048.
- [2] F. S. Jumeilah, "Klasifikasi Opini Masyarakat Terhadap Jasa Ekspedisi JNE dengan Naïve Bayes," *Jurnal Sistem Informasi Bisnis*, vol. 8, no. 1, pp. 92-98, Apr. 2018. <https://doi.org/10.21456/vol8iss1pp92-98>
- [3] Y. Marwanta and B. B., "Analisis Sentimen Pencitraan Perguruan Tinggi di Yogyakarta Menggunakan Metode Naïve Bayes Classifier", *JAIC*, vol. 7, no. 1, pp. 21-27, Jul. 2023.
- [4] D. Vonega, A. Fadila, and D. Kurniawan, "Analisis Sentimen Twitter Terhadap Opini Publik Atas Isu Pencalonan Puan Maharani dalam PILPRES 2024", *JAIC*, vol. 6, no. 2, pp. 129-135, Nov. 2022.
- [5] B. Ramadhan, R. Adam, and I. Maulana, "Analisis Sentimen Ulasan pada Aplikasi E-Commerce dengan Menggunakan Algoritma Naïve Bayes", *JAIC*, vol. 6, no. 2, pp. 220-225, Dec. 2022.
- [6] Mahardika, Y. S., & Zuliarso, E. (2018). Analisis Sentimen Terhadap Pemerintahan Joko Widodo Pada Media Sosial Twitter Menggunakan Algoritma Naives Bayes. *Prosiding SINTAK 2018*, 2015, 409–413.
- [7] Salmaa. (2017). Teknik Pengumpulan Data: Pengertian, Jenis, dan Contoh. <https://penerbitdepublish.com/teknik-pengumpulan-data/>
- [8] Slamet, C., Atmadja, A. R., Maylawati, D. S., Lestari, R. S., Darmalaksana, W., & Ramdhani, M. A. (2018). Recent citations Automated Text Summarization for Indonesian Article Using Vector Space Model. <https://doi.org/10.1088/1757-899X/288/1/012037>
- [9] Sugara, B.; D. D. (2019). View of Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode Vector Space Model. <http://jurnal.stiki.ac.id/J-INTECH/article/view/189/162>
- [10] Zeniarja, J., Salam, A., & Achsanu, I. (2020). Sistem Koreksi Jawaban Esai Otomatis (E-Valuation) dengan Vector Space Model pada Computer Based Test (CBT). *Seri Prosiding Seminar Nasional Dinamika Informatika*, 4(1). <http://prosiding.senadi.upy.ac.id/index.php/senadi/article/view/134>
- [11] Lorosae, T. A., & Prakoso, B. D. (2018). Analisis Sentimen Berdasarkan Opini Masyarakat Pada Twitter Menggunakan Naïve Bayes. *Seminar Nasional Teknologi Informasi Dan Multimedia 2018*

- Universitas Amikom Yogyakarta, 10 Februari 2018, 25–30.
- [12] S. Fauziah, D. N. Sulistyowati, and T. Asra, “Optimasi Algoritma Vector Space Model Dengan Algoritma K-Nearest Neighbour Pada Pencarian Judul Artikel Jurnal”, *pilar*, vol. 15, no. 1, pp. 21–26, Mar. 2019.2
- [13] Darmawan, A., Kustian, N., & Rahayu, W. (2018). Implementasi Data Mining Menggunakan Model SVM untuk Prediksi Kepuasan Pengunjung Taman Tabebuya. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 2(3), 299. <https://doi.org/10.30998/string.v2i3.2439>
- [14] Insanudin, E. (2019). Implementation of python source code comparison results with Java using bubble sort method. *Journal of Physics: Conference Series*, 1280(3). <https://doi.org/10.1088/1742-6596/1280/3/032027>
- [15] Nitami, Mahardika Tania ; Februariyanti, H. (2022). *Jurnal Manajemen Informatika nformatika & Sistem Informasi (MISI) i Jurnal Manajemen Informatika nformatika & Sistem Informasi (MISI) ISSN : 2614-1701 (Cetak) – 2614-3739 (Online) ii. 5.*
- [16] Song, J., Kim, K. T., Lee, B., Kim, S., & Youn, H. Y. (2017). A novel classification approach based on Naïve Bayes for Twitter sentiment analysis. *KSII Transactions on Internet and Information Systems*, 11(6), 2996–3011. <https://doi.org/10.3837/tiis.2017.06.011>
- [17] Widiyanto, M. H. (2019). *Algoritma Naive Bayes | BINUS University Bandung - Kampus Teknologi Kreatif.* <https://binus.ac.id/bandung/2019/12/algoritma-naive-bayes/>