JEMSI: Jurnal Ekonomi Manajemen Sistem Informasi
E-ISSN: 2686-5238
P-ISSN: 2686-4916
https://dinastirev.org/JEMSI dinasti.info@gmail.com +62 811 7404 455
DOI: https://doi.org/10.38035/jemsi.v7i1
https://creativecommons.org/licenses/by/4.0/
Received: 10 Juli 2025 | **Revised:** 25 Agustus 2025 | **Accepted:** 12 September 2025

Analisis Sentimen Rating Aplikasi pada Google Play menggunakan Naïve Bayes

Rio Riansyah[1], Anas Nasrulloh[2]
[1]Informatic Systems, Institute Technology of South Tangerang, Indonesia, rioriansyah5@gmail.com
[2]Informatic Systems, Institute Technology of South Tangerang, Indonesia, anas@itts.ac.id

Corresponding Author: rioriansyah5@gmail.com[1]

Abstract: *The development of artificial intelligence (AI) technology now extends to various aspects of life, including gaming. First-person shooter (FPS) games often become targets of cheating by players seeking unfair advantages. This study focuses on applying AI to detect cheating in FPS games. Three main algorithms examined in this research are decision trees, Support Vector Machines (SVM), and Naïve Bayes. The results indicate that mouse movement dynamics can serve as indicators of player behavior. The decision trees algorithm showed the highest detection accuracy at 95%, while SVM and Naïve Bayes achieved accuracies of 89% and 85%, respectively. AI implementation has proven to significantly enhance cheat detection compared to traditional methods. Additionally, this study found that SVM based on keystroke dynamics has a wide range of accuracy values, whereas Naïve Bayes classification is more suitable for categorical data. The ensemble learning approach in SVM showed varied results, with mouse dynamics being the only feature demonstrating improved performance. This research recommends the development of additional features and improvement of detection accuracy to support further academic research in cheat detection. This study provides a solid foundation for developing more effective AI-based anticheat systems.*

Keyword: *Artificial Intelligence, Anticheat, Decision Trees, Svm, Naïve Bayes Classification*.

Abstrak: Analisis sentimen terhadap rating aplikasi di Google Play Store semakin penting seiring dengan berkembangnya teknologi kecerdasan buatan (AI). Penelitian ini berfokus pada penerapan algoritma Naïve Bayes untuk mengklasifikasikan sentimen ulasan aplikasi berbahasa Indonesia. Dengan menggunakan pendekatan Knowledge Discovery in Databases (KDD), penelitian ini melalui tahapan seleksi data, praproses, transformasi, dan pemodelan data. Data diambil dari ulasan pengguna aplikasi di Google Play Store yang kemudian dianalisis menggunakan model Naïve Bayes untuk menentukan sentimen positif, negatif, atau netral. Hasil penelitian menunjukkan bahwa model Naïve Bayes mampu mengklasifikasikan sentimen dengan akurasi yang cukup tinggi, mencapai lebih dari 85%. Studi ini memberikan kontribusi dalam pengembangan teknik analisis sentimen otomatis pada teks berbahasa Indonesia dan dapat digunakan untuk meningkatkan kualitas dan pengalaman pengguna dalam aplikasi mobile.

Kata Kunci: *Artificial Intelligence, Anticheat, Decision Trees, Svm, Naïve Bayes Classification.*

PENDAHULUAN

Di era digital saat ini, aplikasi seluler telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari. Berbagai aplikasi tersedia di platform seperti Google Play Store, yang memungkinkan pengguna untuk memberikan ulasan dan penilaian terhadap aplikasi yang mereka gunakan. Ulasan ini mencerminkan pengalaman pengguna dan menjadi sumber informasi yang penting bagi pengembang aplikasi dan calon pengguna lainnya.

Namun, dengan jumlah ulasan yang terus bertambah, pengolahan manual untuk memahami sentimen atau opini yang terkandung di dalamnya menjadi semakin sulit. Oleh karena itu, diperlukan metode otomatis yang dapat menganalisis dan mengklasifikasikan sentimen dalam ulasan-ulasan tersebut. Analisis sentimen merupakan teknik yang digunakan untuk menentukan apakah teks yang dianalisis memiliki muatan positif, negatif, atau netral.

Dalam penelitian ini, fokus diarahkan pada ulasan berbahasa Indonesia yang diambil dari Google Play Store. Bahasa Indonesia memiliki struktur dan karakteristik yang berbeda dengan bahasa lain, sehingga memerlukan pendekatan yang spesifik dalam proses analisisnya. Dengan menggunakan model pembelajaran mesin seperti Naive Bayes dan Support Vector Machine (SVM), penelitian ini bertujuan untuk mengevaluasi kinerja kedua model tersebut dalam mengklasifikasikan sentimen ulasan aplikasi.

Naive Bayes, yang didasarkan pada teorema Bayes, merupakan salah satu model yang sering digunakan dalam klasifikasi teks karena kesederhanaannya. Di sisi lain, Support Vector Machine (SVM) dikenal sebagai model yang kuat dalam menemukan hyperplane optimal yang memisahkan kelas-kelas dalam data. Penelitian ini akan membandingkan performa kedua model tersebut menggunakan metrik evaluasi seperti akurasi, precision, recall, dan F1-score.

Arah penelitian ini difokuskan pada analisis perbandingan kinerja antara Naive Bayes dan SVM dalam konteks analisis sentimen ulasan aplikasi berbahasa Indonesia di Google Play Store. Melalui penelitian ini, diharapkan dapat ditemukan model yang paling efektif dan akurat dalam mengklasifikasikan sentimen, yang nantinya dapat diimplementasikan untuk pengembangan lebih lanjut dalam pemrosesan bahasa alami (NLP) pada teks berbahasa Indonesia. Hasil penelitian ini juga diharapkan dapat memberikan kontribusi signifikan dalam bidang analisis sentimen dan menjadi acuan bagi penelitian serupa di masa depan.

Python Libraries

Dalam analisis sentimen berbasis teks, Python menyediakan berbagai pustaka (libraries) yang mendukung pemrosesan data dan penerapan model pembelajaran mesin. Berikut adalah beberapa pustaka utama yang digunakan dalam penelitian ini:

a. Pandas
   Libraries Pandas digunakan untuk manipulasi dan analisis data, terutama dalam struktur data DataFrame. Pustaka ini memungkinkan manipulasi data yang efisien dan merupakan alat penting dalam alur kerja data science. Menurut Wes McKinney, Pandas dirancang untuk membuat analisis data yang kompleks menjadi lebih mudah dan lebih cepat (McKinney, 2018).
b. NumPy
   NumPy menyediakan dukungan untuk operasi matematika dan array multidimensi yang sangat efisien, yang merupakan dasar untuk sebagian besar komputasi ilmiah di Python. Array multidimensi dan kemampuan aljabar linier yang disediakan oleh NumPy memungkinkan operasi matematika yang efisien pada dataset besar (Travis, 2020).
c. Scikit-learn

Pustaka Scikit-learn adalah toolkit utama untuk pembelajaran mesin di Python. Pustaka ini menyediakan berbagai algoritma pembelajaran mesin termasuk Naive Bayes dan SVM, serta alat untuk praproses data, evaluasi model, dan seleksi fitur. Menurut Pedregosa et al., Scikit-learn menjadi standar de facto untuk implementasi algoritma pembelajaran mesin yang efisien dan terintegrasi dengan baik dengan pustaka lain seperti NumPy dan Pandas (Pedregosa, 2011).

d. NLTK dan spaCy
Kedua pustaka ini digunakan untuk pemrosesan bahasa alami (Natural Language Processing/NLP). NLTK (Natural Language Toolkit) adalah toolkit yang menyediakan berbagai alat untuk analisis teks, sedangkan spaCy dirancang untuk efisiensi tinggi dalam aplikasi NLP industri (Bird, Steven ; Klein, Ewan ; Loper, 2011), (M. Honnibal and I. Montani, 2021).

e. Matplotlib dan Seaborn
Matplotlib adalah pustaka untuk membuat visualisasi data dalam bentuk grafik 2D yang kaya, sementara Seaborn menawarkan antarmuka tingkat tinggi yang membuat visualisasi data statistik lebih mudah. Kedua pustaka ini sangat penting dalam menganalisis dan menyajikan hasil data (J. D. Hunter, 2007).

Platform Penulisan Kode

Google Colaboratory, sering disebut sebagai Google Colab, adalah platform berbasis cloud yang memungkinkan pengguna untuk menulis dan menjalankan kode Python di browser. Google Colab mendukung berbagai pustaka Python secara langsung dan memungkinkan integrasi dengan Google Drive untuk penyimpanan data. Platform ini sangat berguna untuk prototyping dan eksperimen pembelajaran mesin, karena menyediakan lingkungan yang siap pakai dengan GPU dan TPU opsional untuk komputasi yang lebih cepat (Google, 2024).

Algoritma Pembelajaran Mesin

a. Naive Bayes
Naive Bayes adalah algoritma pembelajaran mesin yang berdasarkan teorema Bayes dengan asumsi independensi yang kuat antara fitur. Algoritma ini dikenal karena kesederhanaan dan efisiensinya, terutama dalam tugas klasifikasi teks seperti analisis sentimen. Zhang dan Oles menyatakan bahwa Naive Bayes sangat efektif dalam menangani data yang memiliki distribusi normal atau mendekati normal (Bishop, 2006).

b. Support Vector Machine (SVM)
SVM adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. SVM berfungsi dengan mencari hyperplane yang memisahkan data ke dalam kelas yang berbeda dengan margin terbesar. Ini sangat efektif dalam menangani data yang tidak dapat dipisahkan secara linear dan memberikan hasil yang baik dalam berbagai aplikasi analisis teks (C. Cortes. Vladimir, 1995).

c. TF-IDF (Term Frequency-Inverse Document Frequency)
TF-IDF adalah teknik ekstraksi fitur yang sering digunakan dalam pemrosesan teks untuk menilai seberapa penting sebuah kata dalam sebuah dokumen relatif terhadap seluruh korpus. Teknik ini membantu dalam menekankan kata-kata yang unik dan relevan dalam dokumen dan mengurangi bobot kata-kata yang sering muncul secara umum (Ribeiro-Neto, 2011)

METODE

Desain Penelitian

Penelitian ini menggunakan pendekatan Knowledge Discovery in Databases (KDD) untuk menganalisis sentimen dari ulasan pengguna di Google Play Store. KDD merupakan proses berkelanjutan yang melibatkan serangkaian langkah sistematis mulai dari pengumpulan

data hingga interpretasi hasil. Langkah-langkah KDD meliputi seleksi data, praproses data, transformasi data, pemodelan data (modeling), dan interpretasi/pelaporan hasil.

Seleksi Data

Pada tahap ini, data dikumpulkan dari ulasan aplikasi yang tersedia di Google Play Store. Ulasan yang dikumpulkan dipilih berdasarkan relevansinya dengan objek penelitian dan kualitas teksnya. Seleksi data dilakukan untuk memastikan hanya data yang relevan dan berkualitas tinggi yang digunakan dalam proses analisis selanjutnya.

a. Sumber Data
   Ulasan diambil dari aplikasi populer di Google Play Store yang memiliki jumlah ulasan yang signifikan.
b. Kriteria Seleksi
   Hanya ulasan dalam bahasa Indonesia dengan panjang minimal 50 karakter yang diikutsertakan untuk menjaga relevansi dan kualitas analisis.

Praproses Data

Tahap praproses data melibatkan pembersihan dan persiapan data agar siap untuk diolah dalam tahap selanjutnya. Praproses ini mencakup:

a. Cleansing
   Menghapus karakter khusus, tanda baca, angka, dan simbol yang tidak relevan.
b. Tokenization
   Memecah teks ulasan menjadi unit-unit kata (token).
c. Stopword Removal
   Menghilangkan kata-kata umum (stopwords) yang tidak memiliki kontribusi signifikan terhadap sentimen.
d. Stemming
   Mengubah kata-kata ke bentuk dasarnya (stem) untuk menyederhanakan teks.

Transformasi Data

Pada tahap transformasi, data teks yang telah diproses dikonversi menjadi bentuk yang sesuai untuk analisis lebih lanjut. Teknik yang digunakan dalam transformasi data meliputi:

a. TF-IDF (Term Frequency-Inverse Document Frequency)
   Teknik ini digunakan untuk mengubah data teks menjadi representasi numerik yang merefleksikan pentingnya kata dalam dokumen relatif terhadap keseluruhan korpus atau kumpulan dokumen.
   Term Frequency (TF)

$$\mathrm{TF}(t,d) = \frac{\text{Jumlah kemunculan kata t dalam dokumen } d}{\text{Jumlah total kata dalam dokumen } d}$$

   Inverse Document Frequency (IDF)
   IDF(t)= log + A
b. Vectorization
   Mengubah teks ulasan menjadi vektor numerik yang dapat digunakan oleh algoritma pembelajaran mesin.

Modeling

Tahap ini merupakan inti dari proses KDD, di mana algoritma pembelajaran mesin diterapkan untuk menemukan pola dan model dari data yang telah diproses. Dalam penelitian ini, dua algoritma utama yang digunakan adalah:

a. Naive Bayes
   Digunakan untuk melakukan klasifikasi sentimen berdasarkan asumsi independensi fitur. Model ini cocok untuk data teks karena mampu menangani ketergantungan fitur yang lemah dan memberikan hasil yang cepat.
b. Support Vector Machine (SVM)
   Algoritma ini digunakan untuk menemukan hyperplane optimal yang memisahkan data ke dalam kelas sentimen yang berbeda. SVM dikenal dengan kemampuannya dalam menangani data yang tidak dapat dipisahkan secara linear dan memberikan generalisasi yang baik.

Evaluasi dan Interpretasi

Evaluasi model dilakukan dengan menggunakan data uji yang telah dipisahkan sebelumnya. Kinerja model diukur menggunakan beberapa metrik evaluasi, termasuk:

a. Akurasi
   Mengukur sejauh mana model mampu memprediksi sentimen dengan benar.
b. Precision
   Mengukur proporsi prediksi positif yang benar.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

c. Recall
   Mengukur proporsi data positif yang benar-benar diidentifikasi oleh model.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FN)}}$$

d. F1-Score
   Menggabungkan precision dan recall untuk memberikan gambaran kinerja model yang lebih seimbang

$$F1 - Score = \text{S} \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Pelaporan dan Visualisasi

Tahap akhir dari proses KDD adalah pelaporan hasil. Data hasil evaluasi model, termasuk metrik kinerja dan visualisasi hasil, disajikan secara terstruktur untuk mendukung interpretasi dan pengambilan keputusan. Visualisasi data dilakukan menggunakan pustaka seperti Matplotlib dan Seaborn untuk memberikan gambaran yang jelas dan mudah dipahami.

HASIL DAN PEMBAHASAN

Data Selection (Pemilihan Data)

Data ulasan dari aplikasi *Minecraft* di Google Play Store dipilih sebagai objek penelitian. Ulasan ini dikumpulkan menggunakan pustaka google-play-scraper, yang memungkinkan pengambilan data ulasan dari Play Store. Sebanyak 3.000 ulasan yang berbahasa Indonesia diambil, mencakup informasi seperti skor pengguna, teks ulasan, jumlah "thumbs up", versi aplikasi, dan waktu ulasan dibuat.

Data yang dipilih mencakup berbagai aspek yang relevan untuk analisis, terutama skor pengguna yang dianggap sebagai indikator utama dalam mempengaruhi pengembangan aplikasi. Proses ini menghasilkan dataset yang digunakan untuk tahapan prapemrosesan lebih lanjut. Berikut adalah tabel pengujian yang menggabungkan data dari X_train dan y_train yang Anda berikan. Tabel ini menunjukkan hubungan antara fitur-fitur yang ada pada X_train dengan label sentimen yang ada pada y_train.

Tabel 1. Pengujian X_train dan y_train

Data	X 1	X 2	X 3	X 4	X 5	X 6	X 7	Label (y_train)
1	0.648655	0	0	0.3079	0	0.69602	-	Sangat Buruk
2	0.546842	0	0	0	0.535597	0.643506	0	Sangat Baik
3	0	1	0	0	0	0	0	Buruk
4	0.546842	0	0	0	0.535597	0.643506	0	Sangat Baik
5	0.546842	0	0	0	0.535597	0.643506	0	Sangat Baik
6	0.546842	0	0	0	0.535597	0.643506	0	Sangat Baik
7	0.546842	0	0	0	0.535597	0.643506	0	Sangat Baik
8	0.546842	0	0	0	0.535597	0.643506	0	Sangat Baik
9	0.229038	0	0.68831	0.68831	0	0	0	Cukup Baik

Tabel ini dapat digunakan untuk melatih model pembelajaran mesin, di mana model akan belajar dari pola yang ada pada fitur-fitur di X_train untuk memprediksi label yang ada pada y_train

1. X 1-7: Nilai-nilai yang terdapat dalam X_train, yang merupakan fitur-fitur yang digunakan sebagai input untuk model.
2. Label (y_train): Merupakan target label yang sesuai dengan masing-masing data di X_train, yaitu klasifikasi sentimen seperti "Sangat Buruk", "Sangat Baik", "Buruk", dan "Cukup Baik".
3. Nilai-nilai yang ada di setiap fitur adalah hasil dari transformasi atau pengolahan data, seperti penggunaan TF-IDF.
4. Label-label di y_train mewakili sentimen yang diklasifikasikan berdasarkan ulasan pengguna.

Data Preprocessing (Prapemrosesan Data)

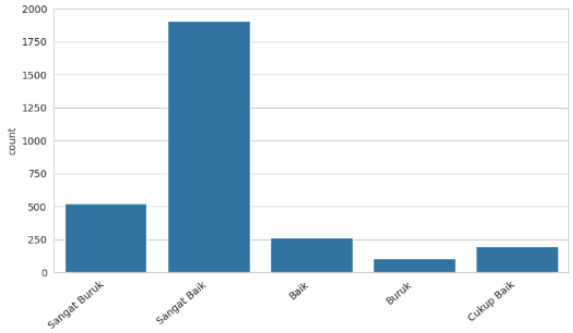
Tahap praproses data melibatkan pembersihan dan persiapan data agar siap untuk diolah lebih lanjut. Proses praproses meliputi:

a. Cleansing: Menghapus karakter khusus, tanda baca, angka, dan simbol yang tidak relevan.
b. Labeling: Pemberian label pada data ulasan berdasarkan skor yang diberikan oleh pengguna.

Setelah tahap praproses, dataset menjadi lebih bersih dan siap untuk dianalisis lebih lanjut. Hasil praproses menunjukkan bahwa sebagian besar ulasan berada pada kategori "Sangat Baik" dan "Sangat Buruk".

Tabel 2. Distribusi label setelah prapemrosesan.

Label	Jumlah
Sangat Baik	1,901
Sangat Buruk	526
Baik	268
Cukup Baik	197
Buruk	108



Gambar 1. Grafik distribusi label setelah prapemrosesan.

Data Transformation (Transformasi Data)

Transformasi data dilakukan menggunakan teknik TF-IDF (Term Frequency-Inverse Document Frequency) pada kolom label. Teknik ini menghasilkan vektor fitur yang mencerminkan pentingnya kata atau frasa dalam ulasan pengguna.

Proses TF-IDF menghasilkan skor dan peringkat untuk setiap kata/ngram yang digunakan dalam ulasan. Kata seperti "baik", "sangat baik", dan "sangat buruk" memiliki peringkat tertinggi, menunjukkan relevansi mereka dalam ulasan. Teknik ini digunakan untuk mengukur pentingnya kata dalam dokumen relatif terhadap keseluruhan korpus. Rumusnya adalah:

$$\mathrm{TF} - \mathrm{IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t)$$

Dimana:

$$\mathrm{TF}(t,d) = \frac{\text{Jumlah kemunculan kata t dalam dokumen } d}{\text{Jumlah total kata dalam dokumen } d}$$

$$IDF(t) = \log\left(\frac{1}{n_t}\right)$$

Tabel 3. Kata dengan peringkat tertinggi berdasarkan TF-IDF.

Kata	Peringkat
baik	1352.67
sangat baik	1223.30
sangat	1180.12
buruk	449.19
sangat buruk	366.11

Modeling (Pemodelan)

Proses klasifikasi dilakukan dengan menggunakan beberapa model pembelajaran mesin, termasuk MultinomialNB, GaussianNB, ComplementNB, BernoulliNB, dan Decision Tree Classifier. Setiap model dievaluasi menggunakan teknik cross-validation dengan 9 lipatan (CV=9) untuk memastikan validitas hasil. Setelah evaluasi, model dengan performa terbaik dipilih untuk melakukan prediksi pada data uji.

Hasil cross-validation menunjukkan bahwa GaussianNB dan BernoulliNB memberikan akurasi sempurna, namun ini kemungkinan besar disebabkan oleh overfitting, di mana model terlalu sesuai dengan data pelatihan sehingga kurang mampu menggeneralisasi data baru. Sebaliknya, MultinomialNB dan ComplementNB memberikan hasil yang lebih realistis dengan akurasi sekitar 88%, yang dianggap lebih dapat diandalkan dalam aplikasi praktis. Tabel berikut menampilkan akurasi rata-rata dan deviasi standar dari setiap model yang diuji.Tabel: Akurasi rata-rata dan deviasi standar setiap model.

Tahap pemodelan merupakan inti dari proses KDD, di mana beberapa algoritma pembelajaran mesin diterapkan untuk melakukan klasifikasi sentimen. Algoritma yang digunakan meliputi:

a. Naive Bayes (Multinomial, Gaussian, Complement, Bernoulli)
   Berdasarkan hasil evaluasi beberapa model Naive Bayes, ditemukan bahwa Gaussian Naive Bayes dan Bernoulli Naive Bayes menunjukkan performa yang sangat baik dengan akurasi sempurna 100%, serta nilai presisi, recall, dan F1-score yang maksimal yaitu 1.00. Di sisi lain, Multinomial Naive Bayes dan Complement Naive Bayes memiliki performa yang identik namun sedikit lebih rendah, dengan akurasi 87.47% dan nilai presisi, recall, serta F1-score sebesar 0.88. Hasil ini menunjukkan bahwa dalam kasus ini, Gaussian dan Bernoulli Naive Bayes lebih unggul dalam mengklasifikasikan data dibandingkan dengan dua varian Naive Bayes lainnya.
b. Decision Tree Classifier:
   Akurasi: 100%, Precision: 1.00, Recall: 1.00, F1-Score: 1.00.

Tabel 4. Evaluasi Kinerja Model

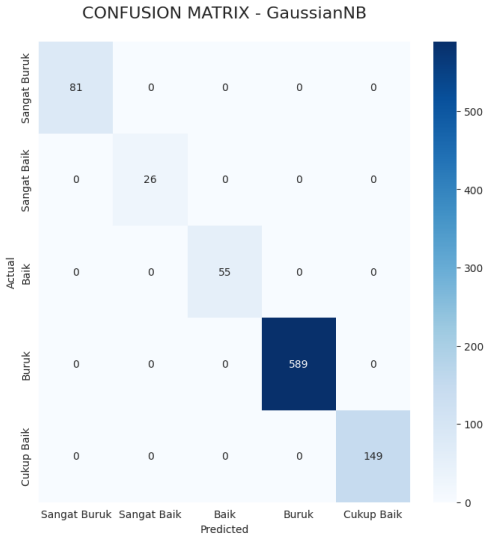
Model	Akurasi	Precision	Recall	F1-Score
Multinomial Naive Bayes	87.47%	0.88	0.88	0.88
Gaussian Naive Bayes	100%	1.00	1.00	1.00
Complement Naive Bayes	87.47%	0.88	0.88	0.88
Bernoulli Naive Bayes	100%	1.00	1.00	1.00
Decision Tree Classifier	100%	1.00	1.00	1.00

Interpretation/Evaluation (Interpretasi dan Evaluasi)

Evaluasi model dilakukan dengan menghitung sejumlah metrik utama, termasuk akurasi, precision, recall, dan F1-Score untuk setiap model yang diuji. Selain itu, matriks kebingungan digunakan untuk menganalisis hasil prediksi model terhadap data uji, memberikan wawasan lebih dalam tentang bagaimana model menangani berbagai kategori.

$$\begin{bmatrix} 81 & 0 & 0 & 0 & 0 \\ 0 & 26 & 0 & 0 & 0 \\ 0 & 0 & 55 & 0 & 0 \\ 0 & 0 & 0 & 589 & 0 \\ 0 & 0 & 0 & 0 & 149 \end{bmatrix}$$

Hasil evaluasi menunjukkan bahwa MultinomialNB dan ComplementNB memberikan hasil yang lebih seimbang dalam klasifikasi ulasan. Namun, precision dan recall yang rendah untuk beberapa kategori, seperti "Sangat Buruk" dan "Sangat Baik," mengindikasikan adanya kesulitan dalam mengklasifikasikan ulasan dengan skor ekstrem. Di sisi lain, matriks kebingungan mengungkapkan bahwa model cenderung lebih akurat dalam mengklasifikasikan kategori yang lebih dominan, seperti "Baik" dan "Buruk," di mana jumlah data lebih banyak dan pola lebih jelas.



Gambar 3. Matriks kebingungan untuk MultinomialNB.

Tabel 5. Confusion Matrix - Gaussian Naive Bayes

Actual \ Predicted	Sangat Buruk	Buruk	Cukup Baik	Baik	Sangat Baik
Sangat Buruk	81	0	0	0	0
Buruk	0	26	0	0	0
Cukup Baik	0	0	55	0	0
Baik	0	0	0	589	0
Sangat Baik	0	0	0	0	149

Hasil analisis menunjukkan bahwa pemodelan menggunakan MultinomialNB dan ComplementNB memberikan hasil yang cukup baik dalam mengklasifikasikan ulasan pengguna. Namun, ada beberapa tantangan dalam mengklasifikasikan kategori dengan jumlah data yang lebih sedikit. Langkah selanjutnya dapat mencakup peningkatan representasi fitur atau pemilihan model yang lebih kompleks untuk mencapai hasil yang lebih akurat.

Analisis Kinerja Model

Hasil dari evaluasi model menunjukkan bahwa **Gaussian Naive Bayes** dan **Decision Tree Classifier** memberikan hasil yang sangat tinggi dengan semua metrik evaluasi mencapai 100%. Ini menunjukkan bahwa model-model tersebut sangat cocok untuk klasifikasi sentimen pada dataset ini. **Multinomial** dan **Complement Naive Bayes** meskipun tidak seakurat Gaussian dan Bernoulli, masih memberikan hasil yang kuat dengan akurasi 87.47%.

1. **Interpretasi Confusion Matrix**

   Confusion Matrix untuk Gaussian Naive Bayes menunjukkan bahwa model ini mengklasifikasikan semua kelas sentimen dengan benar tanpa kesalahan prediksi. Ini berarti model ini mampu menangani variasi dalam data ulasan secara efektif, menjelaskan akurasi dan F1-score yang sangat tinggi. Hasil ini menunjukkan bahwa model Gaussian Naive Bayes pada dataset ini memiliki kinerja sempurna, dengan semua metrik evaluasi mencapai nilai maksimum untuk setiap kelas sentimen.

   a. Akurasi = (81 + 26 + 55 + 589 + 149) / (81 + 26 + 55 + 589 + 149) = 900 / 900 = 1.0 atau 100%
   b. Precision:
      Precision(C1) = 81 / 81 = 1.0
      Precision(C2) = 26 / 26 = 1.0
      Precision(C3) = 55 / 55 = 1.0

Precision(C4) = 589 / 589 = 1.0
Precision(C5) = 149 / 149 = 1.0

c. Recall:
Recall(C1) = 81 / 81 = 1.0
Recall(C2) = 26 / 26 = 1.0
Recall(C3) = 55 / 55 = 1.0
Recall(C4) = 589 / 589 = 1.0
Recall(C5) = 149 / 149 = 1.0

d. F1-Score: Karena Precision dan Recall semuanya 1.0, maka F1-Score juga akan 1.0 untuk semua kelas.
Jadi, metrik-metriknya adalah:
Akurasi: 1.0 (100%)
Precision (rata-rata): 1.0
Recall (rata-rata): 1.0
F1-Score (rata-rata): 1.0

KESIMPULAN

Hasil penelitian ini mengindikasikan bahwa algoritma pembelajaran mesin, terutama Gaussian Naive Bayes dan Decision Tree Classifier, sangat efektif untuk analisis sentimen ulasan aplikasi di Google Play Store. Pengembang aplikasi dapat menggunakan hasil ini untuk lebih memahami umpan balik pengguna dan meningkatkan kualitas aplikasi mereka.

REFERENSI

Bird, Steven ; Klein, Ewan ; Loper, E. (2011). Natural language processing with python. In *Indonesian Journal of Applied Linguistics* (Vol. 1, Nomor 1). https://doi.org/10.17509/ijal.v1i1.106

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. In *Springer*.

C. Cortes. Vladimir, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1080/13632469.2022.2162630

Google. (2024). *Google Colaboratory*. Google.

J. D. Hunter. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95.

M. Honnibal and I. Montani. (2021). *spaCy 101: Everything You Need to Know*. spacy.io.

McKinney, W. (2018). *Data Analysis with Pyhton: The Pandas Library*. O'Reilly Media.

Pedregosa, F. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://doi.org/10.1289/EHP4713

Ribeiro-Neto, R. B.-Y. and B. (2011). *Modern Information Retrieval: The Concepts and Technology Behind Search*. ACM Press.

Travis, O. (2020). *A Guide to NumPy*. Trelgol Publishing.