

## PERBANDINGAN ALGORITMA NAIVE BAYES DAN SUPPORT VECTOR MACHINE PADA ANALISIS SENTIMEN ULASAN GOOGLE PLAY STORE

Rayyan Arifa, Ridwan Dwi Irawan, Nibras Faiq Muhammad

Teknik Informatika, Universitas Duta Bangsa

Jl. Bhayangkara No.55, Tipes, Kec. Serengan, Kota Surakarta, Jawa Tengah

rayyanarifaa123@gmail.com

### ABSTRAK

Lonjakan pemakaian aplikasi seluler memicu peningkatan drastis volume komentar pengguna di Google Play Store yang memuat opini berharga. Akan tetapi, kuantitas data yang masif membuat proses evaluasi sentimen secara manual menjadi sangat tidak efisien dan menyita waktu. Beranjak dari kendala tersebut, riset ini bertujuan untuk mengukur dan membandingkan tingkat keandalan algoritma *Support Vector Machine* (SVM) serta *Naive Bayes* dalam pengklasifikasian polaritas ulasan secara otomatis. Eksperimen dijalankan memanfaatkan korpus "Google Play Store Reviews" dari repositori Kaggle. Teks mentah difilter melewati tahapan *preprocessing* (meliputi *case folding*, *tokenization*, dan *stopword removal*), sebelum dikonversi menjadi format numerik lewat skema pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF). Tolok ukur pengujian bertumpu pada indikator akurasi, presisi, sensitivitas (*recall*), dan *F1-score*. Hasil komparasi membuktikan bahwa arsitektur SVM mendominasi dengan capaian akurasi di angka 82,07%, sukses menundukkan performa *Naive Bayes* yang tertahan pada level 80,83%.

**Kata kunci :** Analisis Sentimen, Pembelajaran Mesin, Support Vector Machine, Naive Bayes, TF-IDF, Google Play Store

### 1. PENDAHULUAN

Perkembangan pesat ekosistem digital saat ini berimbas pada tingginya tingkat adopsi aplikasi seluler oleh masyarakat luas. Seiring dengan tren tersebut, platform distribusi perangkat lunak seperti Google Play Store mengalami lonjakan volume ulasan pengguna secara drastis. Ratusan ribu komentar yang ditinggalkan oleh konsumen sejatinya bukan sekadar teks acak, melainkan rekam jejak opini, pengalaman pemakaian, serta keluhan yang bernilai krusial. Bagi para pengembang, kumpulan umpan balik ini bertindak sebagai fondasi utama untuk mengevaluasi mutu layanan, memperbaiki kendala teknis, hingga merancang fitur baru yang lebih relevan dengan kebutuhan pasar [1].

Meskipun menyimpan wawasan yang sangat berharga, ledakan kuantitas data ulasan ini justru melahirkan rintangan baru bagi pihak pengembang. Upaya untuk memilah dan mengekstraksi sentimen publik secara manual sudah tidak lagi realistis karena sangat menguras waktu dan rentan terhadap bias subjektivitas, sehingga proses evaluasi menjadi sangat tidak efisien [2].

Di ranah akademis, walaupun algoritma komputasi mulai lumrah dimanfaatkan, masih sangat minim literatur yang secara komprehensif mengomparasikan performa model klasifikasi pada korpus data teks yang bersifat universal atau lintas-kategori. Kesenjangan inilah yang menuntut adanya pendekatan terotomatisasi yang teruji keandalannya untuk membedah opini konsumen secara cepat dan akurat [3].

Berpijak pada celah permasalahan di atas, kajian ini diinisiasi dengan tujuan utama untuk mengukur sekaligus membandingkan tingkat efektivitas

algoritma komputasi dalam mengklasifikasikan polaritas ulasan pengguna di Google Play Store. Secara spesifik, penelitian ini bermaksud mengevaluasi sejauh mana perbedaan kinerja antara model *Support Vector Machine* (SVM) dan *Naive Bayes* ketika dihadapkan pada korpus teks yang bersifat general [4].

Pencapaian tujuan tersebut akan ditakar secara kuantitatif dengan bersandar pada instrumen pengukuran performa standar, yang mencakup tingkat ketepatan (*accuracy*), kepresisian (*precision*), kepekaan (*recall*), serta keseimbangan rasio (*F1-score*) [5].

Sebagai rencana penyelesaian masalah, riset ini akan mengimplementasikan teknologi analisis sentimen yang bernaung di bawah disiplin *Natural Language Processing* (NLP)[6].

Pendekatan ini difungsikan untuk menyeleksi dan memetakan teks ke dalam kelompok sentimen positif maupun negatif secara mandiri. Dalam tahap operasionalnya, penyelesaian ini akan melibatkan proses ekstraksi fitur numerik menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) guna mengoptimalkan representasi kata kunci. Setelah vektor angka terbentuk, tahapan pemilahan opini akan dieksekusi secara otomatis oleh algoritma *Naive Bayes* dan SVM. Kedua algoritma pembelajaran mesin (*machine learning*) tradisional ini dikerahkan karena rekam jejak historisnya yang terbukti kokoh, efisien dari aspek komputasi, dan sangat kapabel dalam mencerna dimensi teks berskala masif [7].

### 2. TINJAUAN PUSTAKA

Bagian ini menguraikan berbagai landasan konseptual yang menjadi pijakan riset, di antaranya

mencakup konsep analisis sentimen, penerapan model *Support Vector Machine* dan Naive Bayes, hingga teknik penentuan bobot fitur TF-IDF untuk menunjang pengelompokan opini pengguna aplikasi seluler.

## 2.1. Sentiment Analysis

Merupakan teknik komputasional dalam payung NLP untuk memilah pandangan subyektif warganet dari bentuk teks acak menjadi kelas polaritas yang terstruktur positif ataupun negatif [1].

Analisis sentimen banyak digunakan untuk memahami persepsi pengguna terhadap suatu produk atau layanan berdasarkan data teks tidak terstruktur seperti ulasan aplikasi dan komentar media sosial [2].

Dalam penelitian ini, sentiment analysis digunakan untuk mengklasifikasikan ulasan pengguna aplikasi mobile pada Google Play Store.

## 2.2. Naive Bayes

Pendekatan ini berpijak pada hukum probabilitas Teorema Bayes dengan anggapan dasar bahwa setiap atribut beroperasi secara independent [3].

Algoritma ini banyak digunakan dalam klasifikasi teks karena memiliki proses komputasi yang sederhana, cepat, dan cukup efektif pada data berdimensi tinggi seperti dokumen teks[8].

Pada penelitian ini, Naive Bayes digunakan untuk melakukan klasifikasi sentimen ulasan aplikasi mobile ke dalam kategori positif dan negatif.

Berikut adalah rumus dari Naive Bayes :

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (1)$$

## 2.3. Support Vector Machine

Merupakan teknik *supervised learning* yang berupaya memetakan garis batas ideal (*hyperplane*) guna memaksimalkan jarak pemisah (margin) antar-kelas klasifikasi. [4].

SVM dikenal efektif dalam menangani data teks yang memiliki dimensi tinggi dan mampu menghasilkan performa klasifikasi yang stabil [1].

Dalam penelitian ini, SVM digunakan sebagai metode pembandingan terhadap algoritma Naive Bayes pada proses klasifikasi sentimen ulasan aplikasi mobile.

Berikut adalah konsep hyperplane dari SVM :

$$f(x) = w \cdot x + b \quad (2)$$

## 2.4. TF-IDF

*Term Frequency-Inverse Document Frequency* (TF-IDF) difungsikan sebagai teknik pemberian nilai bobot pada atribut kata agar susunan teks dapat diterjemahkan menjadi format matriks angka [1].

Esensi dari mekanisme ini adalah memberikan skor signifikansi pada setiap kosakata dengan cara membandingkan rasio kehadiran kata tersebut di dalam satu dokumen spesifik terhadap total sebaran kata yang sama di seluruh populasi kumpulan data (korpus)[2].

Skema penimbang bobot ini bertugas mengonversi fitur teks menjadi nilai numerik melalui menimbang frekuensi kemunculan sebuah kata di dalam suatu kalimat berbanding terbalik dengan kemunculannya di korpus.

## 2.5. Text Preprocessing

Text preprocessing merupakan salah satu tahapan paling mendasar dalam Natural Language Processing (NLP) maupun text mining yaitu langkah mutlak untuk menyaring derau (*noise*) pada data mentah. Tahapannya mencakup *case folding* (penyeragaman huruf), *cleaning* (penghapusan tanda baca/symbol), *tokenization* (pemotongan kalimat menjadi kata-kata), serta *stopword removal* (eliminasi kata sambung tak bermakna)[9].

Tahapan text preprocessing yang umum digunakan meliputi:

- Case Folding:** Proses untuk menyeragamkan seluruh karakter huruf di dalam teks menjadi huruf kecil (lowercase) agar kata yang sama dengan variasi kapitalisasi tidak dianggap sebagai token yang berbeda oleh model.
- Cleaning:** Tahap pembersihan teks dari komponen non-kontekstual yang tidak memiliki nilai sentimen, seperti tanda baca, angka, tautan URL, simbol, dan emoji.
- Tokenization:** Proses memecah sebaris kalimat atau dokumen teks utuh menjadi potongan-potongan kata tunggal atau token individu agar dapat dianalisis secara terpisah.
- Stopword Removal:** Tahap eliminasi kata-kata umum (stopwords) yang sering muncul dalam tata bahasa namun tidak membawa informasi spesifik terkait topik atau muatan sentimen, seperti kata hubung "dan", "yang", "di", dan "ke"

## 2.6. Penelitian Terdahulu

Untuk memetakan posisi dan keaslian penelitian ini, dilakukan tinjauan terhadap beberapa studi relevan yang memanfaatkan metode klasifikasi berbasis *machine learning* untuk analisis sentimen teks.

Kajian referensi yang pertama bersumber dari publikasi Maulana dkk. (2024). Mereka merancang eksperimen perbandingan performa antara metode *Support Vector Machine* (SVM) dan Naive Bayes untuk mengelompokkan sentimen komentar konsumen pada platform investasi digital Pluang. Temuan mereka menegaskan bahwa SVM memiliki tingkat ketangguhan yang lebih superior dalam mendeteksi dan memisahkan opini audiens dibandingkan Naive Bayes. Hal yang membedakan proyek tersebut dengan riset ini adalah ruang lingkup himpunan datanya; riset ini tidak membatasi analisis pada satu perangkat lunak finansial spesifik, melainkan mengeksplorasi kumpulan ulasan dari Google Play Store secara menyeluruh dan universal (*general*).

Selanjutnya, terdapat studi dari Emiliana dkk. (2025) yang juga mengandalkan konfigurasi algoritma

serupa (SVM dan Naive Bayes) guna membedah kecenderungan opini masyarakat di jejaring sosial X (sebelumnya Twitter). Evaluasi pemodelan yang mereka kembangkan sukses menorehkan tingkat presisi yang amat baik. Pembeda paling esensial antara literatur tersebut dengan studi saat ini terletak pada asal platform korpus datanya. Karakteristik cuitan di Twitter jelas memiliki struktur bahasa, penggunaan *slang*, dan kepadatan kalimat yang sangat berlainan jika disandingkan dengan ulasan format aplikasi seluler. Oleh karena itu, pendekatan pembersihan teks (*preprocessing*) yang diimplementasikan dalam kajian ini pasti akan melalui tahapan dan penanganan yang berbeda pula.

Referensi ketiga bertumpu pada eksperimen Hariyadi dkk. (2024) yang menguji keandalan SVM serta Naive Bayes dalam mengurai pandangan pengguna terhadap aplikasi desain grafis, Canva. Luaran dari riset mereka memperlihatkan rekam jejak performa yang sangat kompetitif dan saling bersaing di antara kedua metode klasifikasi tersebut. Adapun batasan yang menjadi titik pemisah dengan penelitian ini adalah fokus subjek tinjauannya. Jika Hariyadi dkk. mengarahkan analisis pada perangkat lunak dengan fungsionalitas desain khusus, studi ini mengambil langkah yang lebih luas dengan menelaah opini dari beragam lintas-kategori aplikasi seluler.

Berpijak pada ketiga penelusuran literatur di atas, sangat jelas bahwa metode komparasi algoritma Naive Bayes yang dihadapkan dengan SVM merupakan pendekatan yang lazim dan teruji secara empiris untuk keperluan klasifikasi opini. Kendati demikian, riset yang sedang dikembangkan ini hadir untuk melengkapi celah akademik dengan membawa spektrum data yang lebih masif dan general, yakni menyoroti sentimen pengguna ekosistem Google Play Store secara keseluruhan.

## 2.7. Pengukuran Kinerja

Pengukuran kinerja dari model machine learning yang telah dibangun memerlukan alat ukur evaluasi yang valid. Pengujian pada klasifikasi sentimen umumnya menggunakan instrumen *Confusion Matrix*, yaitu tabel silang yang menyajikan perbandingan antara kelas aktual (data sebenarnya) dengan kelas prediksi yang dihasilkan oleh model[10]. Matriks ini terdiri dari empat elemen representasi data, yaitu:

- True Positive (TP):** Jumlah prediksi kelas positif yang tebakannya terbukti benar.
- True Negative (TN):** Jumlah prediksi kelas negatif yang tebakannya terbukti benar.
- False Positive (FP):** Data yang aslinya negatif namun keliru ditebak sebagai kelas positif oleh algoritma (Kesalahan Tipe I).

- False Negative (FN):** Data yang aslinya positif namun keliru ditebak sebagai kelas negatif oleh algoritma (Kesalahan Tipe II).

## 2.8. Lingkungan Pengembangan dan Alat Bantu

Untuk menilai efektivitas model *machine learning* yang telah dikembangkan, diperlukan instrumen pengukuran performa yang sah. Pengujian klasifikasi umumnya bersandar pada *Confusion Matrix*, yakni sebuah tabel perbandingan silang yang mengukur tingkat kesesuaian antara data aktual (sebenarnya) dengan tebakan (prediksi) dari model. Elemen matriks ini terbagi atas empat kuadran:

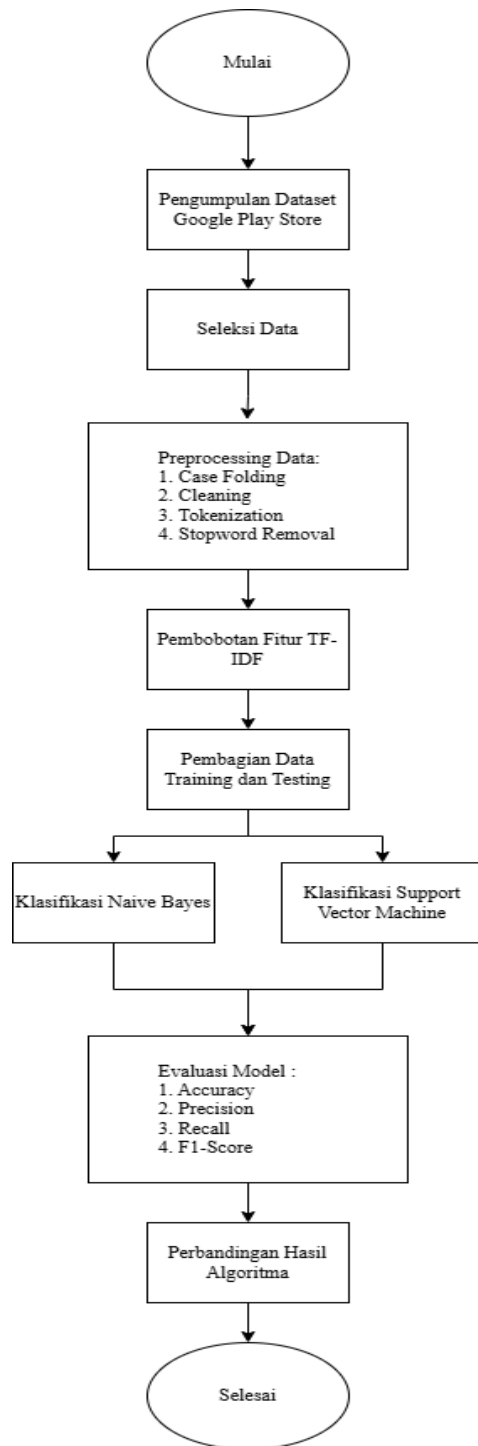
- Python:** Bahasa pemrograman tingkat tinggi (*high-level*) yang dinamis serta berorientasi objek. Bahasa ini menjadi rujukan utama (*standar industri*) di ranah *Data Science*, *Machine Learning*, dan NLP karena struktur kodenya yang ramah pengguna serta ditunjang oleh ekosistem perpustakaan (*library*) yang sangat mumpuni untuk mengolah teks.[4].
- Google Colab:** *Google Colaboratory* merupakan platform ekosistem kerja interaktif gratis bersistem *cloud* (komputasi awan) keluaran Google. Mengadopsi format *Jupyter Notebook*, pengguna dimungkinkan untuk merangkai dan mengeksekusi *script* Python langsung di *browser* tanpa instalasi perangkat lunak lokal. Keunggulan utamanya adalah ketersediaan ruang penyimpanan serta alokasi RAM yang memadai guna menunjang pelatihan data berskala masif secara efisien.[6].

## 2.9. Pembelajaran Mesin

Pembelajaran mesin adalah salah satu disiplin ilmu dalam naungan Kecerdasan Buatan (*Artificial Intelligence/AI*) yang mendesain sistem statistik dan algoritma penunjang. Melalui teknologi ini, sistem komputer sanggup mempelajari pola dari kumpulan data historis untuk meramal atau mengambil keputusan secara otonom tanpa harus diprogram secara kaku. Khusus pada pemrosesan dokumen, *machine learning* bertugas menganalisis pola penyusunan kata guna memetakan sebuah teks ke dalam klaster kategori yang sesuai[7]. Dalam konteks klasifikasi teks, *machine learning* bertugas mengenali pola kata-kata dalam suatu dokumen dan memetakannya ke dalam kategori tertentu

## 3. METODE PENELITIAN

Bagian ini menjabarkan langkah-langkah sistematis yang ditempuh selama pelaksanaan riset. Rangkaian metodologi ini mencakup tahap ekstraksi himpunan data, prapemrosesan teks, penentuan bobot kata via TF-IDF, penerapan model Naive Bayes dan SVM, hingga pengukuran kinerja akhir.



Gambar 1. Flowchart penelitian

Mengacu pada alur diagram penelitian (Gambar 1), prosedur diawali dengan menghimpun komentar pengguna dari Google Play Store, disusul dengan tahap pembersihan data (prapemrosesan). Kemudian, struktur teks dikonversi ke format angka (*numerik*) melalui algoritma TF-IDF agar dapat diolah oleh mesin pemilah (Naive Bayes dan SVM). Terakhir, kemampuan prediksi kedua model diuji komparasinya dengan bersandar pada metrik *accuracy*, *precision*, *recall*, serta *F1-score*..

### 3.1. Pengumpulan Data

Objek observasi dalam studi ini bertumpu pada komentar warganet terkait aplikasi seluler di Google Play Store. Korpus data ini diunduh lewat metode pencarian daring (*internet searching*) dari repositori publik Kaggle, dengan tajuk spesifik "Google Play Store Reviews Dataset". Himpunan data ini dianggap sangat representatif untuk analisis sentimen mengingat volumenya yang masif dan keragaman topik opininya. Variabel utama yang diekstraksi adalah teks ulasan beserta skor bintang (*rating*)

Sebelum diolah, data mentah ini harus melewati tahapan penyaringan (*filtering*) guna mengeliminasi baris kosong, entri ganda (duplikat), dan anomali data lain yang tidak relevan. Pasca-penyaringan, kumpulan ulasan dipartisi ke dalam dua kelompok polaritas: sentimen positif dan negatif, yang klasifikasinya merujuk pada penilaian bintang dari tiap pengguna. Korpus bersih inilah yang nantinya akan disalurkan ke tahap prapemrosesan, perhitungan bobot TF-IDF, serta skenario pelatihan dan pengujian klasifikasi *machine learning*.

### 3.2. Preprocessing Data

Langkah prapemrosesan diimplementasikan untuk menyaring teks mentah agar lebih terstruktur dan siap ditelaah oleh mesin komputasi. Tujuan utamanya adalah mendongkrak mutu data sekaligus mereduksi kemunculan kosakata tak bermakna yang berpotensi mendegradasi kualitas pengelompokan sentimen. Urutan pembersihan yang diaplikasikan meliputi:

- Case Folding**  
Menyeragamkan keseluruhan karakter huruf kapital dalam kalimat menjadi format huruf kecil (*lowercase*) supaya model mesin tidak keliru menganggap kata yang sama sebagai entitas berbeda hanya karena variasi penulisan.
- Cleaning**  
Mengeliminasi berbagai macam derau (*noise*) tekstual yang nirsentimen, di antaranya tanda baca, deret angka, simbol, *emoji*, dan tautan URL, agar sistem bisa berfokus penuh pada kata-kata bermuatan opini.
- Tokenization**  
Memenggal satu kalimat utuh menjadi kepingan-kepingan kata dasar (*token*) sehingga mempermudah mesin untuk mengevaluasi setiap kata secara terpisah
- Stopword Removal**  
Menyeleksi dan menghapus perbendaharaan kata hubung baku (seperti "di", "yang", "dan", "ke") yang frekuensi kemunculannya terlampau tinggi namun nirfaedah dalam mengindikasikan muatan opini. Penghapusan ini membuat dimensi ruang data menjadi lebih ringkas.

Keluaran dari tahapan ini (seperti diilustrasikan pada Tabel 1) berupa teks bersih yang akan dikonversi nilai fiturnya melalui metode TF-IDF.

Tabel 1. Tabel Keluaran Preprocessing Data

| Tahapan          | Hasil                               |
|------------------|-------------------------------------|
| Data Asli        | Aplikasi ini bagus sekali wow!!!    |
| Case Folding     | aplikasi ini bagus sekali wow       |
| Cleaning         | aplikasi ini bagus sekali wow       |
| Tokenization     | [aplikasi, ini, bagus, sekali, wow] |
| Stopword Removal | [aplikasi, bagus]                   |

Tabel 1 menyajikan contoh penerapan tahapan *preprocessing* pada data teks. Proses ini dimulai dari teks asli yang kemudian melewati tahap *case folding*, *cleaning*, dan *tokenization*. Tahap terakhir bertujuan bertujuan untuk menghapuskannya kata yang tidak bermakna agar inti dari ulasan tersebut lebih mudah dianalisis.

### 3.3. Pembobotan Fitur TF-IDF

Pasca-prapemrosesan, korpus teks bersih wajib dimodifikasi menjadi representasi matematis (*numerik*) lewat pendekatan TF-IDF. Mekanisme ini bekerja dengan mengalkulasi seberapa dominan sebuah kata (*token*) di satu dokumen tersebut, berbanding terbalik dengan frekuensi eksistensinya di seluruh kumpulan korpus atau dokumen. Tujuan kalkulasi ini adalah menegaskan signifikansi kata kunci spesifik pembentuk sentimen agar lebih mudah diidentifikasi oleh sistem.

Metode ini dibentuk oleh dua parameter inti, yakni *Term Frequency* (TF) untuk menakar kuantitas kemunculan kata, dan *Inverse Document Frequency* (IDF) untuk mengevaluasi derajat bobot kata tersebut berdasarkan sebaran dokumennya. Perkalian antara nilai TF dan IDF inilah yang melahirkan wujud numerik teks. Adapun formulasi matematika dasar TF-IDF dinyatakan sebagai berikut:

$$TF(t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \quad (3)$$

Merujuk pada formulasi *Term Frequency* (TF) di atas, variabel  $f_{t,d}$  mengindikasikan seberapa sering sebuah token atau term  $t$  muncul di dalam suatu dokumen tunggal  $d$ . Selanjutnya, persamaan matematis untuk *Inverse Document Frequency* (IDF) dirumuskan sebagai berikut:

$$IDF(t) = \log \frac{N}{df(t)} \quad (4)$$

Berdasarkan persamaan *Inverse Document Frequency* (IDF) sebelumnya, variabel  $N$  merepresentasikan total populasi dokumen yang dianalisis, sedangkan  $df(t)$  menunjukkan seberapa banyak dokumen yang memuat kata (*term*)  $t$ . Dengan menggabungkan pemahaman dari kedua komponen TF dan IDF tersebut, maka kalkulasi akhir untuk metode TF-IDF dapat diformulasikan seperti di bawah ini:

$$TFIDF(t, d) = TF(t, d) \times IDF(t) \quad (5)$$

Tabel 2. Contoh Hasil TF-IDF

| Kata   | TF-IDF |
|--------|--------|
| bagus  | 0,72   |
| lambat | 0,65   |
| error  | 0,81   |

Tabel di atas mengilustrasikan hasil kalkulasi bobot TF-IDF untuk sejumlah token uji. Skor matematis ini pada dasarnya berfungsi sebagai indikator untuk menakar seberapa vital peran sebuah kata di dalam suatu kalimat ulasan. Merujuk pada data tersebut, token "error" memegang bobot puncak di angka 0,81, yang mengartikan bahwa kata ini sangat mendominasi dan memiliki pengaruh terkuat dalam struktur teks. Di posisi berikutnya, token "bagus" mencatatkan skor 0,72, disusul oleh token "lambat" dengan skor 0,65. Kesimpulannya, signifikansi sebuah kata terhadap makna keseluruhan teks berbanding lurus dengan besaran nilai TF-IDF yang dimilikinya.

### 3.4. Klasifikasi Sentimen

Agenda inti klasifikasi dieksekusi begitu nilai numerik TF-IDF rampung dibentuk. Dua algoritma *machine learning*, yakni Naive Bayes serta *Support Vector Machine* (SVM), dipercayakan untuk menyeleksi dan memisahkan opini audiens ke kluster positif maupun negatif.

Akan tetapi, sebelum pelatihan dilangsungkan, himpunan dataset wajib dipartisi menjadi dua segmen: data latih (*training*) dan data uji (*testing*). Data latih difungsikan sebagai materi agar mesin bisa mengenali corak sentimen teks, sedangkan data uji disiapkan sebagai sarana validasi guna mengukur tingkat kecerdasan mesin pada variasi data baru. Strategi pemisahan ini vital untuk menggaransi kemampuan adaptasi (*generalisasi*) model prediksi.

Secara mekanis komputasi, Naive Bayes memecahkan probabilitas statistik masing-masing kelas sentimen berbekal ekstraksi fitur kata, sebuah keunggulan yang membuatnya gesit di dimensi tinggi. Sebaliknya, SVM merancang batas bidang pemisah (*hyperplane*) paling optimal demi membelah data menjadi margin terlebar, yang berdampak pada stabilitas hasil. Pelatihan algoritma dijejali dengan data latih bervektor TF-IDF, lalu dievaluasi kinerjanya menggunakan data uji untuk menentukan algoritma yang paling tangguh.

Proses klasifikasi dilakukan dengan melatih setiap algoritma memakai data training yang telah ditunjukkan dalam bentuk vektor TF-IDF. Setelah proses pelatihan selesai, model digunakan untuk memprediksi sentimen pada data testing. Hasil prediksi dari kedua algoritma kemudian dibandingkan untuk mengetahui metode yang memiliki performa terbaik dalam klasifikasi sentimen ulasan aplikasi mobile.

### 3.5. Evaluasi Model

Sebagai pamungkas, kapabilitas SVM dan Naive Bayes wajib dikuantifikasi setelah mereka menuntaskan fase prediksi (*testing*). Tolok ukur yang diadopsi adalah *Confusion Matrix* (Matriks Kebingungan). Matriks ini berfungsi menyingkap seberapa sinkron ramalan algoritma dibandingkan dengan kondisi data sebenarnya. Terdapat empat variabel pengukur di matriks ini: *True Positive* (TP),

True Negative (TN), False Positive (FP), serta False Negative (FN).

Dari keempat variabel penyusun tersebut, diturunkanlah empat parameter evaluasi untuk mengomparasi model:

- Akurasi (Accuracy): Rasio ketepatan tebakan algoritma secara keseluruhan.
- Presisi (Precision): Tingkat kecermatan model saat menunjuk sebuah kelas sentimen spesifik
- Recall: Tolok ukur sensitivitas sistem dalam menemukan seluruh anggota populasi pada suatu kategori
- F1-Score: Nilai ekuilibrium yang menyeimbangkan tarikan metrik presisi dan recall.

Berikut adalah rumus Evaluasi Model :

Accuracy :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Precision :

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

Recall :

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

F1-Score :

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

## 4. HASIL DAN PEMBAHASAN

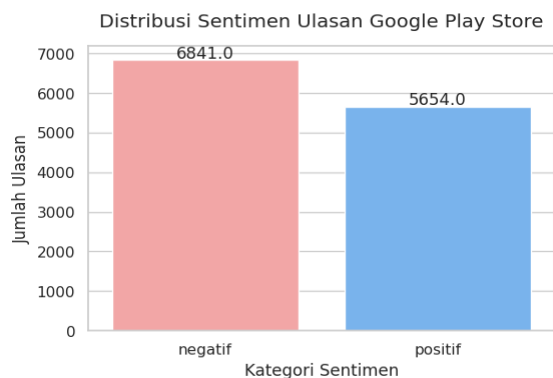
### 4.1. Hasil untuk Pengumpulan Data

Penelitian ini menggunakan dataset sekunder yaitu adalah ulasan pengguna pada berbagai macam aplikasi di Google Play Store, dimana diperoleh dari platform Kaggle dengan nama Google Play Store Reviews Dataset. Berdasarkan proses pengumpulan data yang telah dilakukan, diperoleh total sebanyak 12.495 data ulasan.

Tahap selanjutnya adalah memberikan label sentimen pada dataset tersebut. Pelabelan dilakukan berdasarkan nilai rating (skor 1 hingga 5) yang diberikan pengguna. Hasil dari pengelompokan sentimen tersebut disajikan pada Tabel 3.

Tabel 3. Hasil Pengumpulan Data Sentimen

| Jenis Data       | Jumlah |
|------------------|--------|
| Sentimen Positif | 5.654  |
| Sentimen Negatif | 6.841  |
| Total Data       | 12.495 |



Gambar 2. Distribusi Sentimen Ulasan

Penjelasan dari grafik diatas yaitu ulasan dengan nilai (rating) pada 4 dan 5 dipisah menjadi sentimen positif, sedangkan ulasan dengan nilai (rating) pada 1,2, dan 3 dipisah menjadi sentiment negatif. Total data merupakan hasil dari pengelompokkan kedua sentiment tersebut.

Gambar tersebut merupakan sebuah diagram bar chart yang memvisualisasikan perbandingan jumlah ulasan berdasarkan kategori sentimennya yang sesuai dengan dataset Google Play Store Reviews Dataset yang diperoleh dari Kaggle melalui

### 4.2. Hasil Preprocessing Data

Komentar mentah hasil unduhan umumnya masih dipenuhi oleh derau (*noise*), seperti variasi kapitalisasi yang berantakan, kemunculan simbol, angka, hingga tanda baca yang tak bermakna. Oleh sebab itu, eksekusi pembersihan awal (prapemrosesan) mutlak diperlukan sebelum data teks diserahkan ke mesin pengklasifikasi. Dengan bantuan eksekusi *script* di bahasa pemrograman Python, empat prosedur pembersihan diaplikasikan secara berurutan: *Case Folding*, *Cleaning*, *Tokenization*, dan diakhiri dengan *Stopword Removal*.

Tabel 4. Contoh Preprocessing pada Dataset

| Tahapan          | Hasil                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------|
| Data Asli        | Terrible experience!!! The loading time is extremely slow on my android device.                                  |
| Case Folding     | terrible experience!!! the loading time is extremely slow on my android device.                                  |
| Cleaning         | terrible experience the loading time is extremely slow on my android device                                      |
| Tokenization     | ['terrible', 'experience', 'the', 'loading', 'time', 'is', 'extremely', 'slow', 'on', 'my', 'android', 'device'] |
| Stopword Removal | terrible experience loading time extremely slow android device                                                   |

Transformasi kalimat kotor menjadi kepingan kata bersih tergambar dengan jelas pada Tabel 4. Pada langkah *Case Folding*, semua abjad dikonversi paksa ke format huruf kecil. Selanjutnya, fase *Cleaning* bertugas menghilangkan angka (seperti "910") beserta berbagai tanda baca kurung dan titik. Lewat *Tokenization*, kalimat tersebut diiris menjadi *array* kata tunggal. Sebagai penutup, tahap *Stopword Removal* menyaring kata-kata penghubung bahasa Inggris (misalnya "very", "for", "the", "to") agar tersisa kosakata inti penentu polaritas sentimen..

### 4.3. Hasil Pembobotan TF-IDF

Segera setelah korpus teks dibersihkan, data tersebut wajib ditransformasikan ke dalam susunan angka (*numerik*) agar sistem algoritma *machine learning* mampu mencernanya secara kalkulatif. Pendekatan TF-IDF dipercaya untuk memberi nilai bobot spesifik pada tiap-tiap token. Sebuah kata akan mendapatkan skor penimbang yang besar apabila kata

itu sangat mendominasi suatu ulasan, tetapi jarang terlihat di keseluruhan populasi dokumen. Matriks angka bobot inilah yang nantinya disuntikkan sebagai data masukan (*input feature*) bagi model Naive Bayes (NB) dan juga *Support Vector Machine* (SVM).

#### 4.4. Hasil Pengujian Model Klasifikasi

Guna memvalidasi kinerja algoritma komputasi, himpunan 12.495 komentar tersebut dipartisi menjadi segmen data latih (*training*) serta data uji (*testing*). Proporsi pemisahan ditetapkan memakai rasio 80:20. Skema ini secara otomatis memproduksi 9.996 sampel untuk arena pelatihan model, serta menyediakan 2.499 sampel sisanya untuk evaluasi performa prediksi.

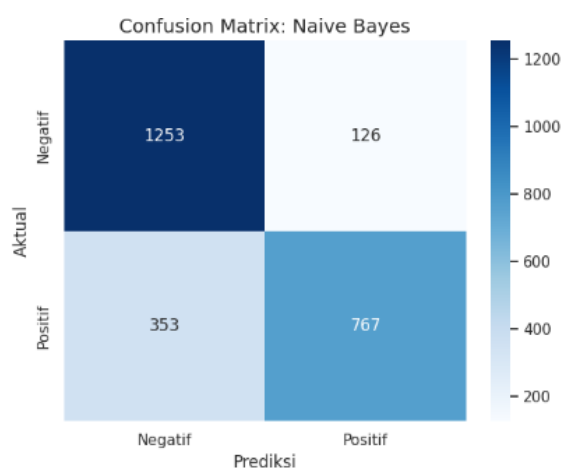
##### 4.4.1. Hasil Klasifikasi Algoritma Naive Bayes

Pemodelan klasifikasi Naive Bayes (jenis Multinomial) disimulasikan memakai data latih, dan kemampuan tebakan dipes secara aktual memakai data uji. Rangkuman kemampuan prediksi algoritma probabilistik ini direpresentasikan melalui *Confusion Matrix* di bawah ini:

Tabel 5. Confusion Matrix Algoritma Naive Bayes

|                | Prediksi Negatif      | Prediksi Positif     |
|----------------|-----------------------|----------------------|
| Aktual Negatif | 1.254 (True Negative) | 125 (False Positive) |
| Aktual Positif | 354 (False Negative)  | 766 (True Positive)  |

Merujuk pada silang data dari matriks di atas, Naive Bayes secara presisi sanggup menebak 1.254 kelas negatif dan melabeli kelas positif secara akurat sebanyak 766 entri. Sayangnya, algoritma ini masih terbentur pada kesalahan prediksi yang cukup terlihat, di mana terdapat 353 ulasan yang sejatinya positif justru keliru dideteksi sebagai sentimen negatif (*False Negative*).



Gambar 3. Heatmap Naive Bayes

Gambar *confusion matrix* di atas memaparkan capaian pengujian dari algoritma Naive Bayes. Secara visual, tingginya tingkat tebakan benar ditunjukkan oleh pekatnya warna biru di sepanjang diagonal utama.

Algoritma ini secara presisi mampu mengelompokkan 1.253 ulasan ke dalam sentimen aktual negatif dan 767 ulasan ke dalam sentimen aktual positif. Walaupun demikian, masih terdapat celah ketidakakuratan yang direpresentasikan oleh gradasi warna terang di luar area diagonal. Rinciannya, 126 sampel bersentimen negatif salah dideteksi sebagai ulasan positif (*false positive*), sedangkan 353 sampel positif keliru ditebak menjadi ulasan negatif (*false negative*). Secara umum, Naive Bayes menampilkan mutu klasifikasi yang memuaskan, di mana model ini terbukti lebih superior dan peka dalam mendeteksi opini bernada negatif ketimbang opini positif.

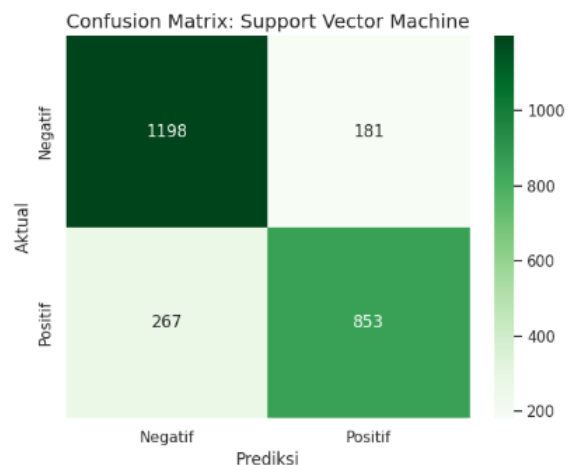
##### 4.4.2. Hasil Klasifikasi Algoritma Support Vector Machine

Pada percobaan kedua, metode *Support Vector Machine* (SVM) dieksekusi dengan konfigurasi spesifik menggunakan kernel linier. Output validasi dari algoritma ini dituangkan dalam bentuk *Confusion Matrix* di Tabel 6:

Tabel 6. Confusion Matrix Algoritma SVM

|                | Prediksi Negatif      | Prediksi Positif     |
|----------------|-----------------------|----------------------|
| Aktual Negatif | 1.198 (True Negative) | 181 (False Positive) |
| Aktual Positif | 265 (False Negative)  | 855 (True Positive)  |

Matriks di atas mendemonstrasikan kapabilitas SVM yang berhasil memetakan 1.198 ulasan negatif dan 855 ulasan positif sesuai dengan data sesungguhnya. Secara komparatif, SVM berhasil menekan angka kegagalan *False Negative* ke titik yang jauh lebih minim ketimbang model Naive Bayes, yakni dengan hanya melakukan 265 tebakan meleset pada sentimen positif.



Gambar 4. Heatmap Support Vector Machine

Pencapaian dari algoritma SVM tergambar dengan jelas melalui pemetaan matriks kebingungan di atas. Secara umum, model ini menunjukkan dominasi tebakan yang presisi, yang direpresentasikan oleh blok hijau tua di sepanjang area diagonal utamanya. Mesin

klasifikasi ini secara akurat mampu melabeli 1.198 sampel ulasan bernada negatif serta 853 sampel ulasan positif. Meski demikian, tetap ditemukan beberapa tebakan yang meleset, sebagaimana diilustrasikan oleh sel-sel berwarna pudar di luar jalur diagonal. Rinciannya, ada 181 entri bersentimen negatif yang salah diidentifikasi sebagai kelas positif (*false positive*), berbarengan dengan 267 entri positif yang gagal dikenali dan justru ditebak menjadi negatif (*false negative*). Sebagai konklusi, algoritma SVM mendemonstrasikan kapabilitas prediksi yang amat tangguh. Rasio tebakan benarnya terbukti jauh

mengungguli angka kesalahannya, sehingga mengonfirmasi kelayakan model ini dalam membedah sentimen di kedua kelas secara optimal.

#### 4.5. Pembahasan dan Perbandingan Performa Algoritma

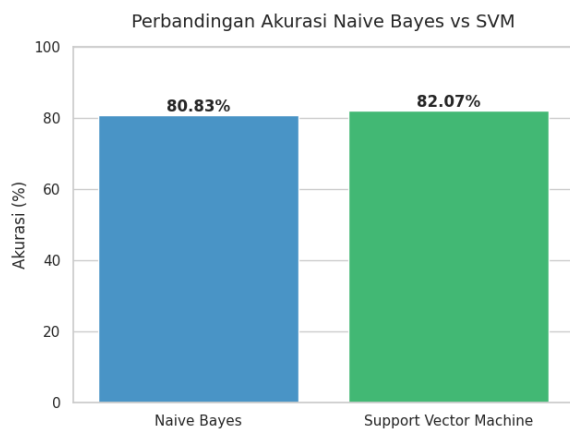
Pasca-kalkulasi tebakan dari matriks kebingungan untuk masing-masing model, agenda penghujung adalah mengomparasikan profil kinerja kedua metode berlandaskan empat parameter esensial: Akurasi, Presisi, *Recall*, dan *F1-Score*. Rangkuman angka perbandingan ini pada Tabel 7.

Tabel 7. Perbandingan Performa Algoritma

| Algoritma | Accuracy | Precision                   | Recall                      | F1-Score                    |
|-----------|----------|-----------------------------|-----------------------------|-----------------------------|
| NB        | 80,83%   | 85,97% (Pos) / 77,98% (Neg) | 68,39% (Pos) / 90,93% (Neg) | 76,18% (Pos) / 83,96% (Neg) |
| SVM       | 82,07%   | 82,52% (Pos) / 81,88% (Neg) | 76,33% (Pos) / 86,87% (Neg) | 79,31% (Pos) / 84,30% (Neg) |

Data hasil uji coba menegaskan fakta bahwa algoritma SVM bertengger di posisi superior dengan pencapaian metrik Akurasi di level 82,07%, menumbangkan catatan Naive Bayes yang tertahan di level 80,83%.

Apabila diamati secara spesifik dari variabel *Recall* untuk ulasan bersentimen positif, SVM terbukti jauh lebih tajam dengan skor penangkapan sebesar 76,33%, mengalahkan kinerja Naive Bayes (68,39%). Lonjakan angka ini menjadi indikasi kuat bahwa mekanisme SVM memiliki kepekaan analitis yang jauh lebih tinggi dalam menjaring dan mengidentifikasi kosakata bermuatan positif dari pengguna aplikasi.



Gambar 5. Grafik Akurasi NB dengan SVM

Sekalipun Naive Bayes tampak sedikit lebih unggul pada metrik presisi di kelas positif (85,97%) dan *Recall* di kelas negatif (90,93%), sistem klasifikasi SVM pada hakikatnya menawarkan performa keseimbangan (*stabilitas*) prediksi antar-kelas yang jauh lebih solid dan tak timpang. Kesolidan ekuilibrium ini dibuktikan oleh sebaran nilai persentase *F1-Score* SVM yang sangat merata dan proporsional pada dua polaritas sentimen (yakni 79,31% untuk positif dan 84,30% untuk negatif).

Dominasi metode *Support Vector Machine* ini sejatinya beresonansi lurus dengan karakteristik teoritisnya yang sangat kokoh (*robust*) manakala diperhadapkan pada problem pengelompokan korpus teks berdimensi sangat lebar dan kompleks, seperti halnya vektor pembobotan hasil TF-IDF. Sistem mesin ini terbukti sangat piawai memformulasikan jarak pemisah maksimum (*hyperplane*) paling optimal guna mendemokratisasi pemisahan area kelas ulasan positif dan negatif secara presisi, menenggelamkan keandalan algoritma probabilitas tebakan independen seperti Naive Bayes.

#### 5. KESIMPULAN DAN SARAN

Merujuk pada serangkaian evaluasi terhadap 12.495 komentar pengguna aplikasi seluler dari Google Play Store, riset ini membuktikan bahwa metode *Support Vector Machine* (SVM) secara umum mendemonstrasikan keandalan klasifikasi sentimen yang lebih superior daripada algoritma Naive Bayes. Penilaian kinerja mesin yang diselaraskan dengan sasaran penelitian ini membuahkan temuan sebagai berikut:

- Accuracy: Algoritma SVM mengungguli Naive Bayes dengan tingkat akurasi sebesar 82,07%, berbanding 80,83%.
- Precision & Recall: Meskipun Naive Bayes memiliki nilai precision sedikit lebih tinggi pada kelas positif (85,97%) dan recall lebih tinggi pada kelas negatif (90,93%), algoritma SVM mampu mendeteksi pola ulasan kelas positif (recall positif) jauh lebih baik sebesar 76,33% dibandingkan Naive Bayes yang hanya 68,39%.
- F1-Score: SVM menunjukkan performa prediksi yang jauh lebih stabil dan berimbang antar-kelas, dibuktikan dengan nilai rata-rata F1-score yang lebih proporsional (79,31% untuk positif dan 84,30% untuk negatif).

Dominasi performa SVM ini pada dasarnya ditopang oleh kapasitasnya yang amat konsisten dalam merumuskan batas pemisah (*hyperplane*) paling ideal. Skema pemetaan ini sangat efektif membelah label

positif dan negatif, bahkan ketika sistem dihadapkan pada ekstraksi fitur teks berdimensi teramat luas yang merupakan hasil kalkulasi TF-IDF.

Berpijak pada konklusi riset tersebut, terdapat beberapa rekomendasi krusial bagi pengembangan studi di masa mendatang. Peneliti selanjutnya sangat disarankan untuk memperlebar spektrum komparasi metode klasifikasinya. Diharapkan, telaah ke depan tidak lagi sebatas mengadopsi model *machine learning* konvensional, melainkan turut mengimplementasikan dan membandingkannya dengan arsitektur *Deep Learning*. Pendekatan berbasis jaringan saraf tiruan tersebut diyakini mampu mendedah dan mengklasifikasikan pandangan warganet dengan potensi akurasi yang jauh lebih presisi dan maksimal.

#### DAFTAR PUSTAKA

- [1] B. A. Maulana, M. J. Fahmi, A. M. Imran, and N. Hidayati, "Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM)," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 375–384, 2024, doi: 10.57152/malcom.v4i2.1206.
- [2] N. A. Pasaribu and Sriani, "The Shopee Application User Reviews Sentiment Analysis Employing Naïve Bayes Algorithm," *Int. J. Softw. Eng. Comput. Sci.*, vol. 3, no. 3, pp. 194–204, Dec. 2023, doi: 10.35870/ijsecs.v3i3.1699.
- [3] A. Supian, B. Tri Revaldo, N. Marhadi, L. Efrizoni, and R. Rahmadden, "PERBANDINGAN KINERJA NAÏVE BAYES DAN SVM PADA ANALISIS SENTIMEN TWITTER IBUKOTA NUSANTARA," *J. Ilm. Inform.*, vol. 12, no. 01, pp. 15–21, Mar. 2024, doi: 10.33884/jif.v12i01.8721.
- [4] A. J. Tobing and A. Febriandirza, "Analisis Sentimen Aplikasi Mobile Banking Bca Pada Ulasan Pengguna di Google Play Store Menggunakan Metode Naive Bayes," *J. Inf. Syst. Res.*, vol. 5, no. 4, pp. 998–1005, Jul. 2024, doi: 10.47065/josh.v5i4.5485.
- [5] Rahel Lina Simanjuntak, Theresia Romauli Siagian, Vina Anggriani, and Arnita Arnita, "Analisis Sentimen Ulasan Pada Aplikasi E-Commerce Shopee Dengan Menggunakan Algoritma Naïve Bayes," *J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 3, no. 3, pp. 23–39, Nov. 2023, doi: 10.55606/teknik.v3i3.2411.
- [6] D. Ananda and R. R. Suryono, "Analisis Sentimen Publik Terhadap Pengungsi Rohingya di Indonesia dengan Metode Support Vector Machine dan Naïve Bayes," *J. MEDIA Inform. BUDIDARMA*, vol. 8, no. 2, p. 748, Apr. 2024, doi: 10.30865/mib.v8i2.7517.
- [7] H. Hariyadi, D. Firdo, and M. H. Al Rafi, "Implementasi Algoritma Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Ulasan Aplikasi Canva," *J. Minfo Polgan*, vol. 13, no. 1, pp. 261–269, Mar. 2024, doi: 10.33395/jmp.v13i1.13568.
- [8] S. A. S. Mola, D. L. B. Baun, I. O. Nunes, and M. M. A. R. Sani, "ANALISIS SENTIMEN APLIKASI HALO BCA DI GOOGLE PLAY STORE MENGGUNAKAN METODE NAIVE BAYES, SUPPORT VECTOR MACHINE DAN RANDOM FOREST," *HOAQ (High Educ. Organ. Arch. Qual. J. Teknol. Inf.)*, vol. 15, no. 2, pp. 69–79, Dec. 2024, doi: 10.52972/hoaq.vol15no2.p69-79.
- [9] I. S. K. Idris, Y. A. Mustofa, and I. A. Salihi, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 32–35, Jan. 2023, doi: 10.37905/jjee.v5i1.16830.
- [10] M. R. Emiliana, M. R. I. Fata, and M. G. Adam, "Perbandingan Metode Naïve Bayes dan Support Vector Machine untuk Analisis Sentimen pada Ulasan Pengguna Aplikasi X," *J. Pustaka Data (Pusat Akses Kaji. Database, Anal. Teknol. dan Arsit. Komputer)*, vol. 5, no. 2, pp. 331–341, Dec. 2025, doi: 10.55382/jurnalpustakadata.v5i2.1460.