

EVALUASI EFEKTIVITAS SUPPORT VECTOR MACHINE DAN RANDOM FOREST DALAM KLASIFIKASI ULASAN PENGGUNA APLIKASI STREAMING VIDIO

Fastabiqul Khusna¹, Siti Nur'aini², Khothibul Umam³, Maya Rini Handayani⁴

^{1,2,3,4} Jurusan Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Walisongo Semarang
Jln. Prof. Dr. Hamka, Ngaliyan, Kota Semarang 50185, Jawa Tengah, Indonesia

¹2208096064@student.walisongo.ac.id*

²siti_nuraini@walisongo.ac.id

³khothibul_umam@walisongo.ac.id

⁴maya@walisongo.ac.id

Abstract

Perkembangan pesat platform streaming telah menghasilkan banyak ulasan pengguna yang dapat dimanfaatkan sebagai sumber masukan untuk pengembangan aplikasi. Penelitian ini dilakukan untuk mengevaluasi kinerja algoritma Support Vector Machine (SVM) dan Random Forest (RF) dalam mengklasifikasikan sentimen ulasan pengguna terhadap aplikasi Vidio. Sebanyak 1.000 ulasan berbahasa Indonesia dikumpulkan menggunakan teknik web scraping dan diberi label sentimen berdasarkan rating bintang, di mana rating 1–2 dikategorikan sebagai sentimen negatif dan 3–5 sebagai sentimen positif. Data ulasan diproses melalui beberapa tahap preprocessing, seperti pembersihan teks, tokenisasi, penghapusan stopword, dan stemming, sebelum dikonversi menjadi representasi numerik menggunakan metode TF-IDF. Dataset dibagi menjadi 80% data latih dan 20% data uji. Kedua model dilatih dan dievaluasi menggunakan metrik accuracy, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa performa yang lebih unggul diperoleh oleh algoritma SVM, dengan akurasi mencapai 76,11%, dibandingkan dengan RF yang memperoleh akurasi sebesar 71,67%. Selain itu, identifikasi ulasan dengan sentimen negatif juga dilakukan dengan lebih efektif oleh SVM. Temuan ini membuktikan bahwa klasifikasi sentimen ulasan aplikasi Vidio lebih tepat dilakukan menggunakan SVM, sehingga berpotensi mendukung otomatisasi analisis sentimen dan peningkatan kualitas layanan streaming. Hasil ini dapat diimplementasikan dalam sistem dashboard otomatis untuk mendeteksi keluhan pengguna secara real-time, memungkinkan pengembang Vidio meningkatkan pengalaman pengguna dengan respons yang lebih cepat dan tepat.

Kata Kunci: Text Classification, User Sentiment, Support Vector Machine, Random Forest, Vidio.

I. PENDAHULUAN

Perkembangan teknologi digital telah mengubah pola konsumsi hiburan masyarakat. Salah satu dampak dari perubahan tersebut yaitu munculnya platform streaming seperti Vidio, yang menawarkan akses cepat ke berbagai konten, termasuk film, serial televisi, dan siaran langsung [1]. Seiring meningkatnya popularitas aplikasi streaming, pengguna mengharapkan layanan berkualitas tinggi dari platform seperti Vidio yang diharapkan tidak hanya menyajikan konten yang menarik, tetapi juga memberikan pengalaman pengguna yang optimal, mulai dari navigasi antarmuka hingga kestabilan sistem dan kenyamanan saat streaming [2]. Dalam hal ini, ulasan dan komentar dari pengguna menjadi sumber wawasan penting untuk mengevaluasi performa aplikasi, ulasan tersebut sering kali mencerminkan kepuasan atau keluhan pengguna terhadap performa aplikasi, serta menawarkan masukan untuk pengembangan lebih lanjut [3]. Namun, dengan bertambahnya ulasan pengguna setiap harinya, melakukan analisis secara manual menjadi sangat tidak praktis. Metode ini tidak hanya menghabiskan waktu dan tenaga, tetapi juga berisiko menghasilkan interpretasi yang kurang konsisten [4]. Oleh karena itu, diperlukan sistem otomatis untuk mengkategorikan ulasan aplikasi streaming Vidio ke dalam sentimen positif atau

negatif. Proses klasifikasi ini memungkinkan pengembang untuk dengan cepat mengidentifikasi keluhan, memahami preferensi pengguna, serta membuat keputusan yang lebih tepat dalam pengembangan aplikasi [5].

Dengan melihat volume ulasan atau komentar yang ditinggalkan oleh pengguna aplikasi Vidio semakin hari terus bertambah, diperlukan metode klasifikasi yang cepat dan presisi. Penelitian ini mengisi gap dengan membandingkan efektivitas SVM dan RF dalam mengklasifikasikan ulasan berbahasa Indonesia pada aplikasi Vidio, sebuah platform streaming lokal yang masih jarang diteliti. Selain itu, banyak penelitian sebelumnya hanya menggunakan pelabelan sentimen biner, yang kurang menangkap ulasan ambigu seperti rating bintang 3, dan jarang membahas performa algoritma pada distribusi kelas tidak seimbang, seperti dominasi ulasan negatif dalam ulasan pengguna Vidio.

SVM dipilih karena efektif dalam menangani data teks berdimensi tinggi [6], sementara RF sebagai metode ensemble menghasilkan prediksi yang stabil dan tidak mudah mengalami *overfitting*, bahkan ketika diterapkan pada data yang kompleks dan tidak seimbang [7].

Berbagai penelitian sebelumnya telah menunjukkan bahwa kedua metode ini cukup efektif. Sebagai contoh, Gunawan et al. (2023) menemukan bahwa SVM mencapai akurasi 84%,

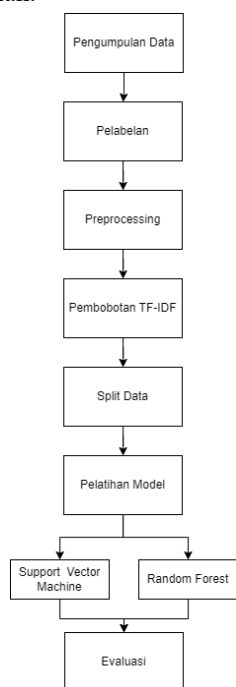
lebih tinggi dibandingkan dengan RF sebesar 80% dalam mengklasifikasikan tingkat stres berdasarkan teks dari media sosial, dengan perbedaan performa juga dipengaruhi oleh teknik transformasi fitur seperti TF-IDF dan CountVectorizer [8].

Dalam penelitian lain yang meneliti aplikasi KAI Access, performa yang lebih unggul juga ditunjukkan oleh algoritma SVM, dengan akurasi yang berhasil dicapai sebesar 97%, sedangkan Random Forest hanya mencapai 93% [9]. Perbedaan akurasi yang cukup mencolok ini menegaskan bahwa SVM mampu melakukan klasifikasi sentimen dengan tingkat presisi yang lebih tinggi. Keunggulan ini mengindikasikan bahwa SVM lebih efisien dalam menangani karakteristik data teks berbahasa alami, terutama dalam membedakan secara akurat antara opini positif dan negatif.

Mengacu pada temuan-temuan tersebut, studi ini dilakukan dengan tujuan membandingkan efektivitas algoritma SVM dan Random Forest dalam proses klasifikasi ulasan pengguna aplikasi Vidio. Diharapkan dari hasil penelitian ini, kontribusi penting dapat diberikan kepada pengembang aplikasi dalam memahami kebutuhan pengguna, serta meningkatkan kualitas dan pengalaman layanan streaming yang disediakan.

II. METODOLOGI PENELITIAN

Penelitian ini dilaksanakan dengan tujuan membandingkan kinerja algoritma Support Vector Machine (SVM) dan Random Forest (RF) dalam mengelompokkan ulasan pengguna aplikasi Vidio dari Google Play Store ke dalam dua kategori sentimen: positif dan negatif. Tahapan penelitian dilakukan melalui proses pengumpulan data, pelabelan data, pra-pemrosesan, pembobotan fitur menggunakan TF-IDF, split data, pelatihan model, serta evaluasi performa model. Gambar 1 menunjukkan tahapan alur penelitian.



Gambar 1. Alur Penelitian.

A. Pengumpulan Data.

Data berupa ulasan pengguna aplikasi streaming Vidio diperoleh secara otomatis dengan bantuan metode web scraping dengan pustaka *google-play-scraper* pada pemrograman *Python*. Dari proses ini, berhasil diperoleh sejumlah komentar dalam Bahasa Indonesia yang merefleksikan beragam tanggapan pengguna terhadap aplikasi Vidio.

B. Pelabelan Data.

Setelah ulasan terkumpul, setiap komentar diberi label sentimen berdasarkan isi komentar dan rating bintang dari pengguna. Ulasan dengan rating bintang 3 hingga 5 dan berisi dukungan dinyatakan sebagai positif, sementara komentar yang mengandung keluhan atau memiliki rating 1 hingga 2 diberi label negatif [10].

C. Preprocessing Data.

Preprocessing data bertujuan untuk menyaring elemen tidak relevan dari ulasan aplikasi streaming Vidio dan mempersiapkannya agar siap dianalisis oleh algoritma machine learning. Proses ini dirancang untuk meningkatkan kualitas data, sehingga hasil klasifikasi diperoleh secara lebih optimal dan akurat. Dalam penelitian ini, preprocessing dilakukan melalui beberapa tahap.

Tahap awal dilakukan *case folding*, yaitu seluruh huruf dalam teks diubah menjadi huruf kecil dengan tujuan sistem dapat memperlakukan kata yang sama secara konsisten, tanpa membedakan penulisan kapitalisasi, contohnya antara “Vidio” dan “vidio” [11]. Selanjutnya, dilakukan *data cleaning*, yaitu pembersihan data dari simbol, tanda baca, angka, serta elemen-elemen tidak penting seperti username atau emotikon yang sering muncul dalam teks ulasan.

Setelah data dibersihkan, tahap berikutnya adalah *tokenizing*, yaitu proses pemecahan kalimat menjadi unit kata atau token. Token ini akan menjadi satuan dasar dalam analisis selanjutnya. Kemudian, dilakukan penghapusan *stopwords*, yaitu kata-kata umum yang tidak memiliki makna sentimen, seperti “dan”, “di”, atau “yang”. Tahap terakhir adalah *stemming*, yaitu di mana kata-kata berimbuhan diubah menjadi bentuk dasarnya atau *root word*. Sebagai contoh, kata “menonton” akan mengalami proses stemming menjadi “tonton”. Proses ini penting untuk menyederhanakan representasi kata dan menghindari redundansi makna [12].

D. Pembobotan TF-IDF.

Dalam penilaian tingkat kepentingan suatu kata pada sebuah ulasan dibandingkan dengan keseluruhan kumpulan data, digunakan metode TF-IDF (Term Frequency–Inverse Document Frequency). Komponen Term Frequency (TF) menghitung berapa banyak suatu kata muncul dalam satu ulasan, sedangkan Inverse Document Frequency (IDF) digunakan untuk memberikan bobot lebih tinggi pada kata-kata yang jarang muncul di seluruh dokumen, karena kata-kata tersebut dianggap lebih bermakna atau khas [13].

E. Split Data.

Pada tahap ini, data ulasan dibagi menjadi dua bagian, yaitu data latih dan data uji. Sebanyak 80% data digunakan untuk proses pelatihan model, sedangkan 20% sisanya dimanfaatkan

untuk menguji performa model dalam memprediksi ulasan baru. Pemisahan dilakukan secara acak melalui fungsi *train_test_split* dari pustaka *scikit-learn* agar representasi data dapat terjaga secara adil dan bias dapat diminimalkan. Rasio ini dianggap ideal, karena memberikan cukup data untuk pelatihan sekaligus menyediakan data uji yang memadai untuk mengevaluasi performa model secara objektif.

F. Klasifikasi atau Pelatihan Model.

Model SVM dan RF dilatih menggunakan 80% data ulasan aplikasi streaming Vidio sebagai data latih dan diuji dengan 20% sisanya sebagai data uji. Pemilihan kernel linear pada algoritma Support Vector Machine (SVM) dilakukan karena data teks yang telah diubah ke dalam bentuk TF-IDF bersifat sparsity tinggi, sehingga kernel non-linear tidak memberikan peningkatan performa yang signifikan. Parameter regularisasi C ditetapkan sebesar 1.0 untuk menjaga keseimbangan antara margin maksimal dan kesalahan klasifikasi. SVM secara teori dikenal efektif untuk klasifikasi dalam ruang fitur berdimensi tinggi [14].

Pada Random Forest, digunakan 100 pohon keputusan ($n_{estimators} = 100$) dengan kedalaman maksimum 30 ($max_depth = 30$) untuk menjaga keseimbangan antara akurasi dan kompleksitas model serta mencegah overfitting, dan ($random_state=42$) untuk konsistensi. Random Forest dikenal memiliki ketahanan terhadap overfitting dan keunggulan dalam melakukan klasifikasi berbasis ensemble learning [15].

Setelah data diolah dan dipisahkan menjadi data latih dan uji, model klasifikasi dilatih menggunakan algoritma Support Vector Machine (SVM) dan Random Forest (RF) untuk mengevaluasi efektivitas klasifikasi ulasan aplikasi streaming Vidio. Kedua algoritma ini dipilih karena keunggulannya dalam analisis sentimen teks. Pada algoritma SVM, pencarian batas pemisah (*hyperplane*) optimal dilakukan untuk memisahkan kelas data secara maksimal, sehingga sangat sesuai untuk data berdimensi tinggi seperti teks, terutama ketika menggunakan kernel linear [16]. Sementara itu, RF sebagai metode ensemble yang mengintegrasikan sejumlah pohon keputusan. Metode ini membangun beberapa pohon selama proses pelatihan dan menggabungkan hasilnya untuk meningkatkan akurasi serta mengurangi risiko *overfitting*. Keunggulan RF terletak pada kemampuannya dalam mengelola data yang kompleks tanpa memerlukan banyak penyesuaian parameter awal, sehingga dihasilkan klasifikasi yang lebih stabil dan andal.

G. Evaluasi.

Untuk mengevaluasi efektivitas Support Vector Machine (SVM) dan Random Forest (RF) dalam mengklasifikasi ulasan aplikasi streaming Vidio, dataset dipisahkan menjadi data latih (80%) dan data uji (20%) menggunakan metode *train-test split*. Data latih digunakan untuk proses pelatihan model, sedangkan data uji menilai kemampuan model memprediksi ulasan baru yang belum dikenali. Evaluasi performa model dilakukan menggunakan empat metrik standar dalam klasifikasi teks, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score* [15]. Keempat metrik ini dipilih karena mampu memberikan gambaran yang komprehensif terhadap kinerja model, khususnya dalam kondisi distribusi kelas yang tidak seimbang, seperti dominasi ulasan positif atau negatif yang umum dijumpai dalam analisis sentimen.

Seluruh proses pengolahan data, pelatihan model, dan evaluasi dilakukan menggunakan bahasa pemrograman Python versi 3.9.12 yang dijalankan di *Jupyter Notebook* pada sistem operasi *Windows 11*. Berbagai pustaka digunakan untuk menunjang eksperimen ini, antara lain *pandas* dan *numpy* untuk manipulasi data dan perhitungan numerik, *matplotlib* dan *seaborn* untuk visualisasi data, serta *re* untuk proses pembersihan teks. Proses representasi teks menggunakan *TfidfVectorizer* dari pustaka *scikit-learn* versi 1.2.2, yang juga digunakan untuk pembagian data, pelatihan model klasifikasi (SVM dan Random Forest), dan evaluasi performa. Untuk pengolahan bahasa alami, digunakan *nltk* versi 3.8.1 terutama untuk tokenisasi dan stopword removal serta *Sastrawi* versi 1.0.1 untuk proses stemming dalam Bahasa Indonesia. Pengambilan data dilakukan menggunakan pustaka *google-play-scraper* versi 1.2.0. Lingkungan eksperimen ini dirancang agar proses dapat direplikasi secara konsisten oleh peneliti lain

III. HASIL DAN PEMBAHASAN

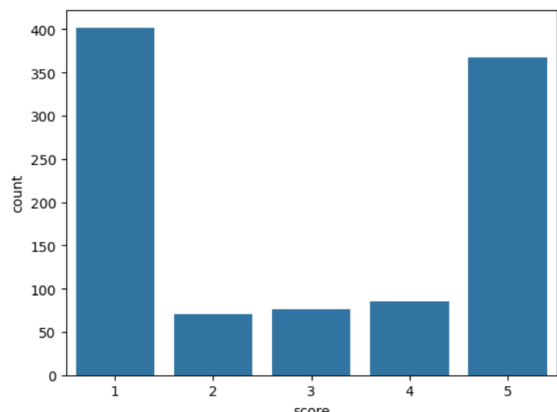
Bagian ini menguraikan setiap tahapan dalam pelaksanaan penelitian, sesuai dengan metodologi yang telah dijelaskan sebelumnya.

Langkah awal dilakukan dengan **pengumpulan data** berupa ulasan pengguna aplikasi streaming Vidio yang diperoleh melalui Google Play Store menggunakan teknik *web scraping*. Proses ini memanfaatkan pustaka *google-play-scraper* pada bahasa pemrograman Python untuk mengakses serta mengambil data ulasan dari pengguna aplikasi Vidio. [17]. Teknik ini dipilih karena efisiensinya, kemudahan penggunaan, dan kemampuan menyaring data sesuai kriteria tertentu, mendukung evaluasi efektivitas klasifikasi sentimen.

Sebanyak 1000 ulasan terbaru dalam bahasa Indonesia berhasil dihipung untuk mengevaluasi pengalaman pengguna aplikasi streaming Vidio. Data yang diperoleh mencakup teks ulasan dan rating bintang (1 hingga 5) yang diberikan pengguna terhadap aplikasi Vidio. Data meliputi teks ulasan dan rating bintang (skala 1 hingga 5). Untuk memastikan ulasan mencerminkan persepsi dan pengalaman pengguna, pengambilan ulasan difokuskan pada kategori "terbaru" (*newest*), dengan parameter bahasa (*lang='id'*) dan wilayah (*country='id'*). Hasil data mentah disajikan pada Gambar 2 dan Gambar 3 merupakan hasil gambar yang menampilkan jumlah data mentah dalam setiap ratingnya.

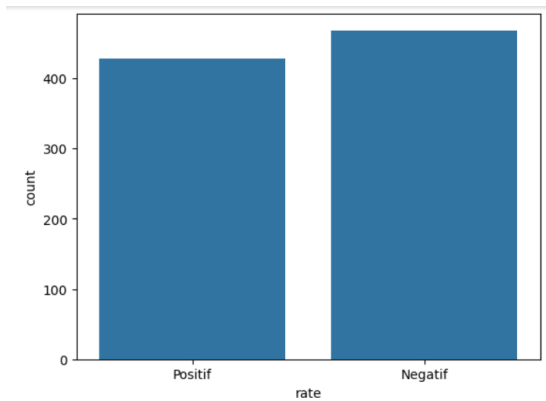
| | content | score |
|---|---|-------|
| 0 | bagus cuma lagi live di jeda sama iklan mengga... | 5 |
| 1 | Mau nonton bola ternyata harus langganan dulu ... | 1 |
| 2 | kaga nyesel download, ragu rugi | 5 |
| 3 | kurangin iklannya donk | 5 |
| 4 | aplikasi bobrok versi gratis kalau baru di buk... | 1 |

Gambar 1. Hasil scraping data



Gambar 2. Data mentah ulasan positif negatif

Setelah data ulasan berhasil dikumpulkan, tahap berikutnya adalah **pelabelan data**, yakni proses pengelompokan setiap ulasan ke dalam kategori sentimen positif atau negatif, sebagaimana ditunjukkan pada Gambar 4. Dalam penelitian ini, pelabelan dilakukan secara semi-otomatis dengan memanfaatkan rating bintang dari Google Play Store. Ulasan yang memiliki rating 1 hingga 2 dikategorikan sebagai negatif, mencerminkan ketidakpuasan pengguna, sedangkan ulasan dengan rating 3 hingga 5 dikategorikan sebagai sentimen positif, mencerminkan kepuasan pengguna.



Gambar 3. Hasil Pelabelan

Ketika data ulasan yang telah diberi label selesai, selanjutnya diproses melalui tahapan **preprocessing** untuk mempersiapkannya menjadi format yang dapat dibaca dan diproses oleh model klasifikasi. Tahap preprocessing adalah sebagai berikut:

Case folding dilakukan untuk mengubah seluruh huruf dalam teks ulasan menjadi huruf kecil untuk menjaga konsistensi dan menghindari perbedaan perlakuan terhadap kata yang sama dengan variasi kapitalisasi (misalnya, “Vidio” dan “vidio” diperlakukan sama). Proses ini menggunakan pustaka Python seperti `str.lower()` untuk menyeragamkan teks. Hasil dari proses ini ditampilkan pada Tabel 1.

Tabel 1. Hasil case folding

| Content | Case Folding |
|---|---|
| bagus cuma lagi live di jeda sama iklan mengga... | bagus cuma lagi live di jeda sama iklan mengga... |
| Mau nonton bola ternyata harus langganan dulu ... | mau nonton bola ternyata harus langganan dulu ... |
| kaga nyesel download, ragu rugi | kaga nyesel download, ragu rugi |

kurangin iklannya donk
aplikasi bobrok versi gratis
kalau baru di buk...

kurangin iklannya donk
aplikasi bobrok versi gratis kalau
baru di buk...

Data Cleaning. Selanjutnya adalah tahapan data cleaning yaitu menghilangkan elemen tidak relevan seperti tanda baca, simbol, angka, atau karakter dari ulasan aplikasi streaming Vidio untuk menghasilkan teks yang lebih bersih dan fokus pada kata-kata bermakna. Menggunakan pustaka re dengan ekspresi reguler seperti `[^a-zA-Z\s]`, proses ini memfilter teks hasil case folding untuk menghilangkan noise seperti tanda elipsis atau spasi berlebih. Hasil dari case folding tertera pada Tabel 2.

Tabel 2. Hasil data cleaning

| Content | Data cleaning |
|--|--|
| bagus cuma lagi live di jeda sama iklan mengga... | bagus cuma lagi live di jeda sama iklan mengga... |
| Mau nonton bola ternyata harus langganan dulu ... | mau nonton bola ternyata harus langganan dulu ... |
| kaga nyesel download, ragu rugi | kaga nyesel download ragu rugi |
| kurangin iklannya donk aplikasi bobrok versi gratis kalau baru di buk... | kurangin iklannya donk aplikasi bobrok versi gratis kalau baru di buk... |

Tokenizing. Langkah ini merupakan memecah teks ulasan aplikasi streaming Vidio menjadi unit-unit kata (token) berdasarkan spasi, menghasilkan daftar kata yang dapat dianalisis secara individual oleh model klasifikasi. Pada tahap ini menggunakan fungsi `word_tokenize` dari pustaka `nltk` untuk menguraikan teks hasil data cleaning, memungkinkan model SVM dan RF mendeteksi pola sentimen dengan lebih akurat. Tabel 3 adalah hasil dari tahapan tokenizing.

Tabel 3. Hasil tokenizing

| Content | Tokenizing |
|--|---|
| bagus cuma lagi live di jeda sama iklan mengga... | 'bagus', 'cuma', 'lagi', 'live', 'di', 'jeda', 'sama', 'iklan', 'menggangu' |
| Mau nonton bola ternyata harus langganan dulu ... | 'mau', 'nonton', 'bola', 'ternyata', 'harus', 'langganan', 'dulu' |
| kaga nyesel download, ragu rugi | 'kaga', 'nyesel', 'download', 'ragu', 'rugi' |
| kurangin iklannya donk aplikasi bobrok versi gratis kalau baru di buk... | 'kurangin', 'iklannya', 'donk', 'aplikasi', 'bobrok', 'versi', 'gratis', 'kalau', 'baru', |

Langkah berikutnya yaitu **Stopwords**. Yaitu menghilangkan kata-kata umum seperti “di”, “sama”, “cuma”, atau “dulu” dari daftar token ulasan aplikasi streaming Vidio karena tidak memiliki makna sentimen signifikan. Menggunakan daftar stopwords bahasa Indonesia dari pustaka `nltk`, diperluas dengan kata seperti “lagi” atau “kalau”, proses ini mengurangi dimensi data dan memfokuskan analisis pada kata-kata bermakna. Tabel 4 merupakan hasil dari tahapan stopwords.

Tabel 4. Hasil stopwords

| Case Folding | Stopword |
|---|---|
| 'bagus', 'cuma', 'lagi', 'live', 'di', 'jeda', 'sama', 'iklan', 'menggangu' | 'bagus', 'live', 'jeda', 'iklan', 'menggangu' |
| 'mau', 'nonton', 'bola', 'ternyata', 'harus', 'langganan', 'dulu' | 'nonton', 'bola', 'langganan' |
| 'kaga', 'nyesel', 'download', 'ragu', 'rugi' | 'kaga', 'nyesel', 'download', 'ragu', 'rugi' |
| 'kurangin', 'iklannya', 'donk' | 'kurangin', 'iklannya', 'donk' |

| | |
|---|---|
| 'aplikasi', 'bobrok', 'versi', 'gratis', 'kalau', 'baru', 'di', 'buka', 'ngelag', 'karna' | 'aplikasi', 'bobrok', 'versi', 'gratis', 'buka', 'ngelag', 'karna' |
|---|---|

Lalu tahap akhir yaitu **Stemming**. Stemming adalah mengubah kata dalam ulasan aplikasi streaming Vidio ke bentuk dasarnya dengan menghapus imbuhan, seperti “meng-” atau “-an”, menggunakan pustaka Sastrawi untuk bahasa Indonesia. Misalnya, “mengganggu” menjadi “ganggu”. Tabel 5 merupakan hasil dari tahapan stemming. Proses ini menyederhanakan variasi kata, mengurangi redundansi, dan meningkatkan akurasi klasifikasi sentimen, mendukung evaluasi efektivitas SVM dan RF.

Tabel 5. Hasil stemming

| Stopword | Stemming |
|---|--|
| bagus', 'live', 'jeda', 'iklan', 'mengganggu' | 'bagus', 'live', 'jeda', 'iklan', 'ganggu' |
| 'nonton', 'bola', 'langganan' | 'nonton', 'bola', 'langgan' |
| 'kaga', 'nyesel', 'download', 'ragu', 'rugi' | 'kaga', 'nyesel', 'download', 'ragu', 'rugi' |
| 'kurangin', 'iklannya', 'donk' | 'rangin', 'iklan', 'donk' |
| 'aplikasi', 'bobrok', 'versi', 'gratis', 'buka', 'ngelag', 'karna' | 'aplikasi', 'bobrok', 'vers', 'gratis', 'buka', 'ngelag', 'karna' |

Hasil akhir **preprocessing**, setelah melwati seluruh tahapan akan menghasilkan teks yang ringkas dan terfokus, seperti ditampilkan pada Tabel 6.

Tabel 6. Hasil preprocessing

| Content | Content Cleaned |
|--|---|
| bagus cuma lagi live di jeda sama iklan mengga... | bagus live jeda iklan ganggu |
| Mau nonton bola ternyata harus langganan dulu ... | nonton bola langgan |
| kaga nyesel download, ragu rugi | kaga nyesel download ragu rugi |
| kurangin iklannya donk | rangin iklan donk |
| aplikasi bobrok versi gratis kalau baru di buk... | aplikasi bobrok versi gratis buka ngelag karna |

Setelah preprocessing selesai, tahap selanjutnya adalah menerapkan **pembobotan TF-IDF** ulasan pengguna aplikasi streaming Vidio. Teknik ini digunakan untuk menghitung bobot kata berdasarkan frekuensi kemunculannya dalam ulasan tertentu (Term Frequency) serta tingkat kelangkaannya pada keseluruhan kumpulan ulasan (Inverse Document Frequency). Dengan metode ini, setiap kata dalam ulasan akan diberikan bobot berdasarkan tingkat kepentingannya; kata yang sering muncul tetapi kurang informatif akan memiliki bobot rendah, sementara kata-kata yang jarang tetapi unik akan mendapatkan bobot lebih tinggi. Transformasi ini menghasilkan vektor numerik berdimensi tetap untuk setiap ulasan, yang kemudian dapat digunakan sebagai input untuk algoritma klasifikasi. Proses ini dilaksanakan dengan bantuan pustaka TfidfVectorizer dari scikit-learn dalam Python.

Setelah data berhasil diubah menjadi vektor TF-IDF, tahapan keempat **split data** yaitu membagi data menjadi data latih dan data uji untuk mengevaluasi performa model dalam mengklasifikasi ulasan aplikasi streaming Vidio. Pembagian data dilakukan menggunakan teknik train-test split menggunakan rasio 80:20, yaitu 80% data dialokasikan untuk pelatihan model dan 20% untuk uji kemampuan model dalam menggeneralisasi data baru. Teknik ini memungkinkan model

mempelajari pola dari data latih, kemudian diuji dengan data yang belum pernah dikenal untuk mengukur tingkat generalisasinya. Dengan hasil data uji total 180 ulasan data.

Setelah data latih dan uji disiapkan, tahapan selanjutnya adalah **pelatihan model** yaitu algoritma Support Vector Machine (SVM) dan Random Forest (RF) untuk ulasan aplikasi streaming Vidio. Pemilihan kedua algoritma ini didasarkan pada performa mereka yang sudah terbukti efektif dalam tugas-tugas klasifikasi teks, khususnya analisis sentimen. Model dilatih menggunakan 80% data ulasan aplikasi streaming Vidio sebagai data latih dan diuji dengan 20% sisanya sebagai data uji. Proses pelatihan dan pengujian ini diimplementasikan dengan pustaka scikit-learn pada Python untuk memastikan efisiensi dan konsistensi.

Tahap terakhir dalam penelitian ini adalah **evaluasi** performa model. Tahap ini sangat penting untuk menilai efektivitas algoritma klasifikasi yang digunakan, yaitu SVM dan Random forest dalam mengklasifikasi sentimen ulasan pengguna aplikasi streaming Vidio. Evaluasi kinerja dilakukan menggunakan empat metrik utama dalam klasifikasi teks, yakni *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, yang semuanya dihitung berdasarkan Confusion Matrix. Confusion Matrix ini mencakup empat komponen penting: *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*, yang masing-masing menunjukkan tingkat keberhasilan atau kesalahan model dalam memprediksi sentimen positif maupun negatif.

Hasil evaluasi pada data uji dari model SVM ditampilkan pada Gambar 5, sedangkan hasil dari model Random Forest ditampilkan pada Gambar 6.

| | | | | |
|---------------------|-----------|--------|----------|---------|
| Evaluasi Model SVM: | | | | |
| | precision | recall | f1-score | support |
| Negatif | 0.80 | 0.76 | 0.78 | 99 |
| Positif | 0.72 | 0.77 | 0.74 | 81 |
| accuracy | | | 0.76 | 180 |
| macro avg | 0.76 | 0.76 | 0.76 | 180 |
| weighted avg | 0.76 | 0.76 | 0.76 | 180 |

Confusion Matrix:
[[75 24]
[19 62]]
Akurasi SVM: 0.7611111111111111

Gambar 4. Evaluasi Model SVM

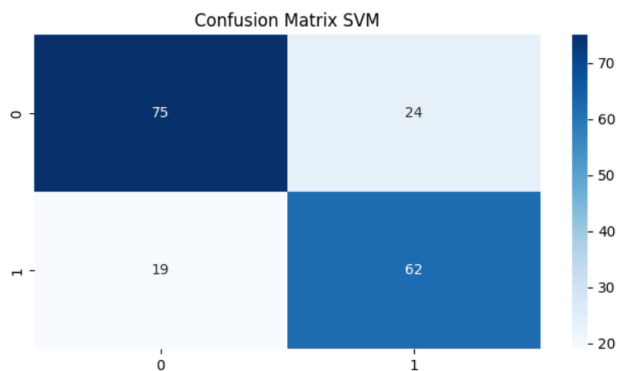
| | | | | |
|-------------------------------|-----------|--------|----------|---------|
| Evaluasi Model Random Forest: | | | | |
| | precision | recall | f1-score | support |
| Negatif | 0.78 | 0.68 | 0.72 | 99 |
| Positif | 0.66 | 0.77 | 0.71 | 81 |
| accuracy | | | 0.72 | 180 |
| macro avg | 0.72 | 0.72 | 0.72 | 180 |
| weighted avg | 0.73 | 0.72 | 0.72 | 180 |

Confusion Matrix:
[[67 32]
[19 62]]
Akurasi Random Forest: 0.7166666666666667

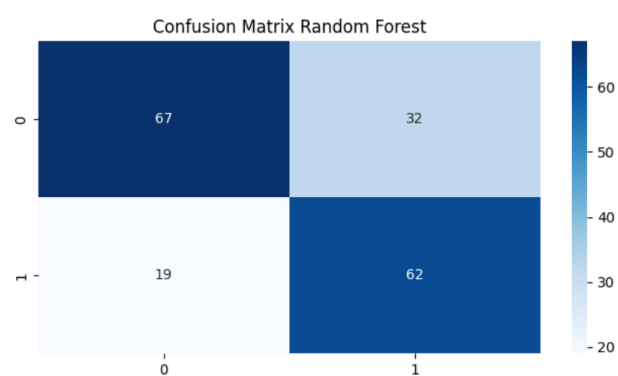
Gambar 5. Evaluasi Model Random Forest

SVM menunjukkan keunggulan dibandingkan RF pada semua indikator evaluasi. *Accuracy* yang dicapai oleh SVM sebesar 76,11%, lebih tinggi dari Random Forest yang memperoleh akurasi 71,67%. Begitu pula dengan nilai *f1-score* SVM yang konsisten lebih baik di kedua kelas, yaitu 0.78 untuk

kelas negatif dan 0.74 untuk kelas positif, dibandingkan dengan Random Forest yang memperoleh 0.72 dan 0.71 pada kelas yang sama.



Gambar 6. Confusion Matrix Metode SVM



Gambar 7. Confusion Matrix Metode Random Forest

Hasil *confusion matrix* pada Gambar 7 dan Gambar 8 dari masing-masing model memberikan gambaran detail tentang kemampuan klasifikasi terhadap dua kelas sentimen. Pada model SVM, sebanyak 75 ulasan yang berhasil diklasifikasikan dengan benar sebagai sentimen negatif, sementara 62 ulasan yang tepat dikenali sebagai sentimen positif. Namun, masih ditemukan kesalahan klasifikasi, yakni 24 ulasan negatif yang salah diprediksi sebagai positif, serta 19 ulasan positif yang keliru diklasifikasikan sebagai negatif. Sementara itu, model Random Forest hanya mampu mengklasifikasikan 67 ulasan negatif secara benar, dengan jumlah kesalahan pada kelas negatif yang lebih tinggi yaitu 32 prediksi salah. Untuk kelas positif, hasilnya serupa dengan SVM, yakni 62 ulasan berhasil diklasifikasikan dengan benar, dan 19 ulasan positif salah diprediksi menjadi negatif. Dari sini dapat disimpulkan bahwa SVM lebih unggul dalam mengidentifikasi ulasan negatif dan memiliki kesalahan klasifikasi yang lebih rendah secara keseluruhan dibandingkan Random Forest.

Hasil pengujian menunjukkan bahwa algoritma SVM memiliki akurasi yang lebih tinggi dibandingkan Random Forest. Hal ini disebabkan oleh kemampuannya dalam membangun hyperplane optimal yang dapat memisahkan kelas ulasan berdasarkan representasi vektor TF-IDF. Sifat sparsity dari data teks sangat sesuai dengan pendekatan kernel linear dalam SVM, yang secara teoretis memang unggul untuk klasifikasi data berdimensi tinggi. Selain sifat sparsity data teks, distribusi kelas dalam dataset juga memengaruhi performa kedua algoritma. Dari 1.000 ulasan yang dikumpulkan, sekitar 60% ulasan dikategorikan sebagai negatif (rating 1–2) dan 40%

sebagai positif (rating 3–5), menunjukkan ketidakseimbangan kelas, SVM lebih unggul dalam menangani distribusi kelas yang tidak seimbang karena kemampuannya dalam memaksimalkan margin pemisah antar kelas, sementara Random Forest cenderung bias ke kelas mayoritas (negatif), menghasilkan 32 kesalahan klasifikasi negatif sehingga menghasilkan lebih banyak kesalahan klasifikasi pada kelas positif berbanding 24 pada SVM. Temuan ini konsisten dengan literatur yang menyatakan bahwa SVM lebih robust terhadap ketidakseimbangan kelas dibandingkan metode ensemble seperti Random Forest [15].

Selain itu, temuan ini sejalan dengan hasil studi terdahulu seperti Gunawan et al. (2023) dan Sidauruk et al. (2023), yang menunjukkan bahwa SVM konsisten memberikan performa lebih tinggi dalam klasifikasi opini pengguna berbasis teks. Kesalahan klasifikasi, seperti 24 ulasan negatif yang salah diprediksi sebagai positif oleh SVM, kemungkinan besar disebabkan oleh ulasan yang bersifat ambigu atau netral, seperti yang umumnya terjadi pada ulasan dengan rating bintang 3. Untuk mengatasi permasalahan ini, disarankan penggunaan pendekatan analisis sentimen multi-kelas (positif, netral, negatif), serta validasi manual pada sebagian data untuk memastikan kualitas pelabelan yang lebih akurat. Sebagai contoh, ulasan seperti "aplikasi bagus tapi iklan terlalu banyak" dengan rating 3 sering kali ambigu karena mengandung elemen positif ("bagus") dan negatif ("iklan terlalu banyak"), sehingga sulit diklasifikasikan secara biner sebagai positif atau negatif.

IV. KESIMPULAN

Dalam penelitian ini, sebanyak 1.000 ulasan pengguna aplikasi streaming Vidio dari Google Play Store berhasil dikumpulkan dan digunakan sebagai dataset untuk klasifikasi sentimen. Pelabelan dilakukan berdasarkan rating bintang, dengan kategori negatif (1–2) dan positif (3–5), menunjukkan dominasi sentimen negatif. Hasil evaluasi menunjukkan bahwa algoritma Support Vector Machine (SVM) memberikan performa terbaik dengan akurasi sebesar 76,11% dan F1-score masing-masing 0,78 (negatif) dan 0,74 (positif), mengungguli algoritma Random Forest yang hanya mencapai akurasi 71,67%. Temuan ini menegaskan bahwa SVM lebih efektif dalam menangani data teks ulasan berbahasa Indonesia, terutama dalam mendeteksi keluhan pengguna. Lebih dari itu, hasil ini memberikan kontribusi penting dalam pengembangan sistem klasifikasi otomatis berbasis pembelajaran mesin untuk analisis sentimen, khususnya pada aplikasi streaming lokal yang belum banyak diteliti. Kesimpulan ini juga menunjukkan potensi implementasi nyata, seperti pengembangan dashboard cerdas untuk memantau keluhan pengguna secara real-time, yang pada akhirnya dapat membantu pengembang Vidio meningkatkan kualitas layanan dan pengalaman pengguna. Meski demikian, studi ini memiliki keterbatasan, yaitu penggunaan skema pelabelan biner yang tidak mengakomodasi ulasan netral atau ambigu. Oleh karena itu, penelitian lanjutan disarankan untuk mengadopsi pendekatan multi-kelas dan mengeksplorasi algoritma lain seperti BERT atau model deep learning berbasis transformer guna memperoleh hasil klasifikasi yang lebih presisi dan adaptif.

V. SARAN

Untuk penelitian selanjutnya, disarankan agar pengujian dilakukan menggunakan algoritma lain, seperti model *deep learning* (misalnya, BERT), guna meningkatkan akurasi klasifikasi, serta memperluas analisis dengan tools machine learning lainnya yang mendukung pemrosesan teks berbahasa Indonesia. Analisis multi-kelas (positif, netral, negatif) dan validasi manual pada ulasan ambigu (rating 3) disarankan untuk meningkatkan presisi. Penelitian ini diharapkan dapat memberikan masukan berharga bagi pengembang aplikasi streaming dalam memahami kebutuhan dan preferensi pengguna, serta mendorong peningkatan kualitas layanan di masa yang akan datang.

REFERENSI

- [1] A. B. Pranata and F. Abdillah, Allif Rizki Irwiensyah, "Analisis Sentimen Ulasan Pengguna Aplikasi Netflix Pada Google Play Menggunakan Algoritma Naïve Bayes," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 6, pp. 3091–3098, 2024.
- [2] G. Kanugrahan, V. Hafizh, C. Putra, and Y. Ramdhani, "Analisis Sentimen Aplikasi Gojek Menggunakan SVM, Random Forest dan Decision Tree," *J. Infortech*, vol. 6, no. 2, pp. 171–178, 2024.
- [3] A. Ameliaa, L. N. Hayati, and H. Darwis, "Analisis Sentimen Masyarakat Terhadap Sistem Pembayaran Mypertamina dengan Metode Random Forest, SVM, dan Naïve Bayes," *LINIER Lit. Inform. dan Komput.*, vol. 1, no. 1, pp. 28–44, 2024.
- [4] U. Kulsum, M. Jajuli, and N. Sulistiyowati, "Analisis Sentimen Aplikasi WETV di Google Play Store Menggunakan Algoritma Support Vector Machine," *J. Appl. Informatics Comput.*, vol. 6, no. 2, pp. 205–212, 2022.
- [5] Anisa Ma'û Luthfi and Fatkhurokhman Fauzi, "Perbandingan Klasifikasi Random Forest, Support Vector Machines, dan LGBM Pada Klasifikasi Kualitas Udara di Jakarta," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 9, no. 2, pp. 99–108, 2024, doi: 10.32528/justindo.v9i2.1912.
- [6] M. F. Al-shufi and A. Erfina, "Sentimen Analisis Mengenai Aplikasi Streaming Film Menggunakan Algoritma Support Vector Machine Di Play Store," *Sismatik*, pp. 156–162, 2021.
- [7] S. Nur Adhan, G. N. A. Wibawa, D. C. Arisona, I. Yahya, and R. Ruslan, "Analisis Sentimen Ulasan Aplikasi Wattpad Di Google Play Store Dengan Metode Random Forest," *AnoaTIK J. Teknol. Inf. dan Komput.*, vol. 2, no. 1, pp. 6–15, 2024, doi: 10.33772/anoatik.v2i1.32.
- [8] N. Fathirachman Mahing, A. Lazuardi Gunawan, A. Foresta Azhar Zen, F. Abdurrachman Bachtiar, and S. Agung Wicaksono, "Klasifikasi Tingkat Stress dari Data Berbentuk Teks dengan Menggunakan Algoritma Support Vector Machine (SVM) dan Random Forest," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 7, pp. 1527–1536, 2023, doi: 10.25126/jtiik.1078010.
- [9] N. Sidauruk, N. Riza, and R. N. Siti Fatonah, "Penggunaan Metode Svm Dan Random Forest Untuk Analisis Sentimen Ulasan Pengguna Terhadap Kai Access Di Google Playstore," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 3, pp. 1901–1906, 2023, doi: 10.36040/jati.v7i3.6899.
- [10] A. S. A. Rafsanjani, D. L. Fithri, and Supriyono, "Analisis Sentimen Ulasan Pengguna Aplikasi KitaLulus pada Google Play Store dengan menggunakan algoritma Support Vector Machine (SVM)," *Sist. J. Sist. Inf.*, vol. 14, no. 5, pp. 2519–2530, 2025.
- [11] M. Apriliyani, M. I. Musyaffaq, S. Nur'Aini, M. R. Handayani, and K. Umam, "Implementasi analisis sentimen pada ulasan aplikasi Duolingo di Google Playstore menggunakan algoritma Naïve Bayes," *AITI*, vol. 21, no. 2, pp. 298–311, Sep. 2024, doi: 10.24246/aiti.v21i2.298-311.
- [12] S. Rohimah, M. Afdal, and R. Novita, "Analisis Sentimen Traveloka Berdasarkan Ulasan Google Play Store Menggunakan Algoritma Support Vector Machine dan Random Forest," vol. 6, no. 3, pp. 1709–1716, 2024, doi: 10.47065/bits.v6i3.6300.
- [13] R. Ramadhan, Y. A. Sari, and P. P. Adikara, "Perbandingan Pembobotan Term Frequency-Inverse Document Frequency dan Term Frequency-Relevance Frequency terhadap Fitur N-Gram pada Analisis Sentimen," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 11, pp. 5075–5079, 2021, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10173>
- [14] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
- [15] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [16] R. B. Gumilar, Y. Cahyana, C. E. Sukmawati, and A. M. Siregar, "Analisa Perbandingan Algoritma Support Vector Machine dan K- Nearest Neighbors Terhadap Ulasan Aplikasi Vidio," *J. Inf. Syst. Res.*, vol. 5, no. 4, pp. 1188–1195, 2024, doi: 10.47065/josh.v5i4.5640.
- [17] A. M. Ndapamuri, D. Manongga, and A. Iriani, "Analisis Sentimen Ulasan Aplikasi Tripadvisor Dengan Metode Support Vector Machine, K-Nearest Neighbor, Dan Naïve Bayes," *INOVTEK Polbeng - Seri Inform.*, vol. 8, no. 1, p. 127, 2023, doi: 10.35314/isi.v8i1.3260.