Analisis Sentimen Ulasan Aplikasi dengan *Multinomial Naïve Bayes*, *Logistic Regression*, dan SVM

Rahelita Pasaribu[a1], Ida Ayu Gde Suwiprabayanti Putra[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jl. Raya Kampus Unud, Jimbaran, Bali, 80361, Indonesia
[1]pasaribu.2308561113@student.unud.ac.id
[2]iagsuwiprabayantiputra@unud.ac.id

Abstract

The swift uptake of mobile health applications has led to an increase in user-generated feedback, providing important insights into public satisfaction. To explore user sentiments, this study analyzes 9,848 reviews from a health-oriented application utilizing three machine learning methods: Multinomial Naïve Bayes, Logistic Regression, and Support Vector Machine (SVM). The feedbacks were classified as positive or negative. The methodology included standard preprocessing such as cleaning and stemming, feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF), and addressing class imbalance with the Synthetic Minority Oversampling Technique (SMOTE). Models were fine-tuned and verified through 5-fold cross-validation. Effectiveness was measured by accuracy, precision, recall, and F1-score. Logistic Regression and SVM reached the greatest accuracy at 92%, while Naïve Bayes trailed at 86%. Logistic Regression showed strong precision (95%) and recall (94%) for positive reviews, with SVM performing comparably. These results emphasize the capability of sentiment analysis in enhancing digital health services through information-based user feedback.

Keywords*: Sentiment Analysis, Mobile Application, TF-IDF, Logistic Regression, Naïve Bayes, SVM*

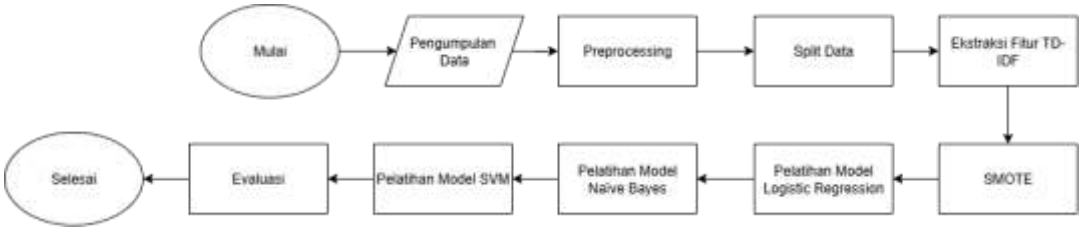
1. Pendahuluan

Pemerintah Indonesia mendorong transformasi digital di bidang kesehatan, salah satunya lewat peluncuran aplikasi mobile yang memudahkan akses masyarakat terhadap layanan dan informasi kesehatan secara online. Inisiatif ini bertujuan meningkatkan efisiensi serta jangkauan layanan publik. Namun, banyaknya pengguna dan ulasan di platform seperti Google Play Store menunjukkan perlunya evaluasi berkelanjutan untuk menilai kepuasan dan kualitas layanan. Ulasan pengguna merupakan bentuk pernyataan tertulis yang memuat opini, evaluasi, atau tanggapan terhadap suatu produk atau layanan. Isi dari ulasan ini sering kali mencerminkan tingkat kepuasan pengguna sekaligus mengungkapkan kekurangan sistem melalui masukan berupa kritik maupun saran yang membangun [1].

Untuk menjawab kebutuhan tersebut, penelitian ini menggunakan analisis sentimen sebagai pendekatan komputasi untuk mengelompokkan komentar pengguna menjadi kategori positif dan negatif. Analisis sentimen merupakan teknik pemrosesan data teks yang bertujuan memahami dan mengklasifikasikan ekspresi emosional pengguna [2]. Proses ini dilakukan dengan menerapkan algoritma machine learning seperti *Logistic Regression*, *Support Vector Machine* (SVM), dan *Naïve Bayes* yang terbukti efektif dalam pengolahan teks berskala besar. Tujuan studi ini adalah menilai persepsi pengguna terhadap aplikasi kesehatan *mobile* dan menyediakan dasar berbasis data untuk peningkatan kualitas layanan digital. Studi serupa pada aplikasi iPusnas menunjukkan dominasi sentimen positif dari pengguna, yang mencerminkan tingkat kepuasan tinggi terhadap layanannya [3]. Temuan tersebut mendukung pentingnya masukan pengguna dalam evaluasi layanan digital. Dengan pendekatan yang sama, penelitian ini membandingkan kinerja ketiga algoritma dalam klasifikasi sentimen ulasan guna memperkuat strategi pengembangan layanan kesehatan digital berbasis umpan balik pengguna.

2. Metode Penelitian

2.1 Kerangka Penelitian



Gambar 1. Alur Penelitian

Studi ini dimulai dengan pengumpulan data *review* dari aplikasi *mobile*. Data yang didapat kemudian dikelola melalui tahapan *pre-processing* agar siap digunakan oleh model. Selanjutnya, fitur diekstraksi dengan menggunakan metode TF-IDF. Sebelum pelatihan model, teknik SMOTE (*Synthetic Minority Oversampling Technique*) diterapkan untuk menangani ketidakseimbangan kelas. Setelahnya, data dilatih memakai tiga model *machine learning*, yaitu *Logistic Regression*, *Naïve Bayes*, dan *Support Vector Machine* (SVM). Performa setiap model dinilai dengan memanfaatkan metrik akurasi, presisi, *recall*, dan *F1-score*. Seluruh tahapan proses dilaksanakan di platform Google Colab dengan memakai bahasa pemrograman Python dalam lingkungan Jupyter Notebook.

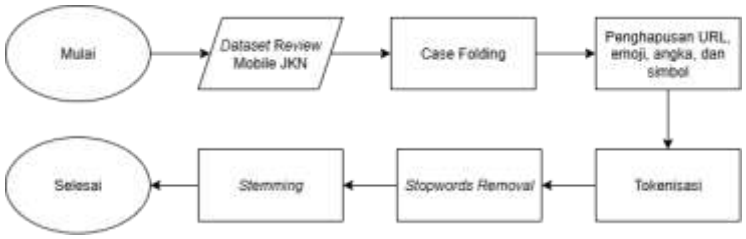
2.2 Sumber dan Deksripsi Data

Data yang digunakan dalam studi ini diambil dari sumber sekunder yang diperoleh melalui platform Kaggle. *Dataset* ini berisi sebanyak 9.848 ulasan pengguna terhadap aplikasi Mobile JKN yang diunduh dari Google Play Store. Semua data telah dikategorikan ke dalam dua jenis sentimen, yaitu positif dan negatif.

Tabel 1. Contoh *Dataset*

content	score	label
webnya bapuk. kurang satset. gk jelas	1	negatif
mantap	5	positif
sejauh ini aplikasi milik pemerintah, mobile JKN yg paling good	5	positif
kenapa mau ambil no antrean untuk kontrol”kouta poli udh habis”	2	negatif

2.3 *Text Preprocessing*



Gambar 2. Bagan Alur Tahap *Preprocessing*

Langkah-langkah praproses dimulai dengan *case folding* untuk menyelaraskan semua huruf menjadi huruf kecil. Selanjutnya, dilakukan pembersihan data, seperti menghapus URL, emoji, angka, dan simbol yang tidak relevan. Setelah itu, proses tokenisasi dilakukan untuk memecah teks menjadi kata-kata. Langkah berikutnya adalah penghapusan *stopword* untuk mengeliminasi

kata-kata umum yang tidak terlalu bermakna, seperti "dan" atau "di". Akhirnya, diterapkan *stemming* untuk mengembalikan setiap kata ke bentuk dasar, sehingga kata-kata turunan yang memiliki akar sama dapat disamakan.

2.4 Ekstraksi fitur TF-IDF

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan untuk mengevaluasi sejauh mana pentingnya suatu kata dalam dokumen tertentu dibandingkan dengan kumpulan dokumen lainnya. Pendekatan ini menggabungkan dua elemen utama, yaitu *Term Frequency* (TF), yang mengukur frekuensi kemunculan suatu kata dalam dokumen dibandingkan dengan total kata yang ada, serta *Inverse Document Frequency* (IDF), yang menilai seberapa khas kata tersebut di antara seluruh dokumen yang diperiksa [4]. Adapun rumus TF-IDF adalah sebagai berikut:

$$tf = 0{,}5 + 0{,}5 \times \frac{tf}{max(tf)} \quad (1)$$

$$idf_t = log\left(\frac{D}{df_t}\right) \quad (2)$$

$$W_{d.t} = tf_{d.t} \times idf_{d.t} \quad (3)$$

Dalam konteks perhitungan TF-IDF, *t* merujuk pada term ke-*t* dalam dokumen, *tf* adalah frekuensi kemunculan term tersebut, dan *idf* merupakan *Inverse Document Frequency*. Nilai *df* menunjukkan jumlah dokumen yang mengandung term tersebut, sementara *D* adalah dokumen ke-*d*. Hasil akhirnya berupa *W*, yaitu bobot dari dokumen ke-*d* terhadap term ke-*t*.

2.5 SMOTE (*Synthetic Minority Oversampling Technique*)

SMOTE (*Synthetic Minority Oversampling Technique*) diciptakan oleh Nitesh V. Chawla pada tahun 2002 sebagai jawaban untuk menangani ketidakseimbangan kelas pada *dataset*. Berbeda dengan *oversampling* umum yang hanya melakukan salinan pada data minoritas, SMOTE membentuk sampel sintetis baru dengan memanfaatkan pendekatan *k-nearest neighbor* (k-NN). Pada data numerik, jarak geometris diterapkan, sementara untuk data kategorikal, digunakan *Value Difference Metric* (VDM). Metode ini berkontribusi dalam membentuk data pelatihan yang lebih seimbang dan mengurangi pengaruh terhadap kelas mayoritas [5].

2.6 *Logistic Regression*

Logistic Regression merupakan teknik regresi yang dimanfaatkan untuk mengevaluasi *relation* antara variabel independen dan variabel dependen yang bersifat kategoris, seperti biner (0 dan 1) atau dikotomi (ya atau tidak). Berbeda dengan regresi berganda, metode ini dirancang khusus untuk menangani variabel dependen kualitatif. *Logistic Regression* sangat efisien dalam menyelesaikan masalah klasifikasi, terutama ketika hasil yang diinginkan berupa kategori tertentu. Model ini dinyatakan melalui persamaan (4) dan (5) [6].

$$Ln\left(\frac{p}{1-p}\right) = B_0 + B_1X \quad (4)$$

$$p = \left(\frac{e^{(B_0+B_1X)}}{1+e^{(B_0+B_1X)}}\right) \quad (5)$$

Dalam persamaan regresi logistik, B_0 merupakan konstanta, B_1 adalah koefisien dari setiap variabel prediktor, dan *p* menunjukkan probabilitas terjadinya kejadian Y = 1. Pada penelitian ini, algoritma *Logistic Regression* diterapkan dengan parameter C=100 sebagai pengatur kekuatan regularisasi, *penalty*='l2' untuk mencegah *overfitting*, *max_iter*=1000 sebagai batas iterasi maksimum, dan *class_weight='balanced'* untuk menangani ketidakseimbangan kelas. Proses *tuning hyperparameter* dilakukan menggunakan GridSearchCV dengan *5-fold cross-validation*, menggunakan variasi nilai C = [0.001, 0.01, 0.1, 1, 10, 100] dan *penalty* = ['l1', 'l2'] untuk

memperoleh kombinasi parameter terbaik berdasarkan akurasi.

2.7 *Naïve Bayes*

Naïve Bayes adalah model klasifikasi berbasis probabilistik yang terkenal dengan kesederhanaannya namun efektif, seperti dalam analisis sentimen. Algoritma ini berdasarkan pada Teorema Bayes, yang memperhitungkan peluang suatu kelas bersumber pada informasi sebelumnya. Kata "*naïve*" mencerminkan asumsi bahwa setiap fitur dianggap saling bebas secara kondisional, artinya kehadiran satu kata tidak memengaruhi kata lainnya. Dalam pemrosesan teks, NB menggunakan pendekatan *bag-of-words* (BoW) untuk mengekstraksi fitur tanpa memperhatikan urutan kata. Model ini cocok untuk data diskrit seperti frekuensi kata, karena menghitung probabilitas kelas dengan mengasumsikan setiap kata muncul secara independen dalam dokumen.

$$P\ (label|features)\ =\ \frac{P(label)\ *\ (features|label)}{P(features)} \quad (6)$$

Dalam konteks *Naïve Bayes*, *P(label|features)* menunjukkan probabilitas suatu label tertentu berdasarkan sekumpulan fitur yang diamati. *P(label)* merepresentasikan peluang awal atau prior dari label tersebut sebelum mempertimbangkan fitur. *P(features|label)* adalah kemungkinan munculnya fitur-fitur tersebut jika diketahui labelnya, sedangkan *P(features)* menunjukkan probabilitas keseluruhan dari fitur-fitur tersebut tanpa memperhatikan label.

Multinomial Naïve Bayes merupakan varian dari NB yang umum digunakan dalam *text classification*, seperti analisis sentimen. Algoritma ini dirancang untuk menangani fitur berbasis frekuensi kata, seperti yang diperoleh melalui bag-of-words atau TF-IDF. Model ini mengasumsikan bahwa setiap fitur bersifat independen secara kondisional terhadap fitur lainnya dalam satu kelas, serta mengikuti distribusi multinomial. Probabilitas suatu dokumen dikalkulasi berdasarkan frekuensi kemunculan kata dalam dokumen, dan *class* dengan nilai kemungkinan terbesar dipilih sebagai hasil klasifikasi [7]. Dalam penelitian ini, *Multinomial Naïve Bayes* digunakan dengan parameter *alpha*=0.1 sebagai teknik *smoothing* untuk menghindari probabilitas nol. Proses *tuning hyperparameter* dilakukan menggunakan GridSearchCV dengan *5-fold cross-validation*, dengan rentang *alpha* mulai dari 0.1 hingga 10.0 untuk menemukan konfigurasi terbaik berdasarkan akurasi.

2.8 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah algoritma pembelajaran mesin yang digunakan untuk mengklasifikasikan data dengan label yang telah ditetapkan. SVM mampu mengidentifikasi pola sentimen dan emosi pengguna terhadap aplikasi berdasarkan isu yang dihadapi [8]. Dalam penelitian ini, SVM diterapkan dengan berbagai kombinasi *hyperparameter* untuk memperoleh konfigurasi terbaik. Parameter yang disesuaikan meliputi C = [0.1, 1, 10, 100] sebagai pengatur regularisasi, *gamma* = [1, 0.1, 0.01, 0.001] untuk mengontrol jangkauan pengaruh data latih pada kernel RBF, serta kernel = ['rbf', 'linear'] untuk menentukan fungsi pemisah data. Proses optimisasi dilakukan menggunakan GridSearchCV dengan *5-fold cross-validation* untuk mendapatkan kombinasi parameter yang menghasilkan akurasi tertinggi.

2.9 *Confusion Matrix*

Sesudah tahap klasifikasi, kinerja model dievaluasi dengan bantuan *confusion matrix*. *Confusion matrix* merupakan metode analisis yang sering diaplikasikan untuk menilai ketepatan model klasifikasi, baik dalam situasi biner maupun *multiclass* [9]. Dari matriks ini, dapat dihitung empat metrik utama: ketepatan, presisi, *recall*, dan skor F1. Berikut adalah rumus dari masing-masing metrik evaluasi berdasarkan *confusion matrix*.

$$Accuracy\ =\ \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

$$Precision\ =\ \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (10)$$

3. Hasil dan Diskusi

3.1. *Preprocessing*

Proses pengolahan awal dilaksanakan pada kumpulan data yang terdiri dari 6.541 entri berlabel positif dan 3.307 entri negatif. Langkah ini dimulai dengan pengubahan huruf menjadi huruf kecil secara menyeluruh, diikuti dengan penghapusan elemen yang tidak relevan seperti URL, simbol, emoji, dan angka. Berikutnya, dilakukan pemecahan teks menjadi token melalui tokenisasi, kemudian penghapusan kata umum dengan menggunakan *library* Sastrawi untuk menyingkirkan kata-kata yang tidak memberikan kontribusi pada analisis. Tahap terakhir adalah pengembalian kata ke bentuk dasarnya menggunakan Sastrawi. Contoh hasil pengolahan awal dapat dilihat pada Tabel 2.

Tabel 2. Contoh hasil *preprocessing*

	content	**label**
Data Awal	aplikasi apa ini saya verifikasi sudah 6x Gonta ganti nomor telfon masa tidak ada muncul notifikasinya, parah benar	negatif
Case Folding	aplikasi apa ini saya verifikasi sudah 6x gonta ganti nomor telfon masa tidak ada muncul notifikasinya, parah benar	negatif
Hapus URL, simbol, emoji, dan angka	aplikasi apa ini saya verifikasi sudah x gonta ganti nomor telfon masa tidak ada muncul notifikasinya parah benar	negatif
Tokenization	['aplikasi', 'apa', 'ini', 'saya', 'verifikasi', 'sudah', 'x', 'gonta', 'ganti', 'nomor', 'telfon', 'masa', 'tidak', 'ada', 'muncul', 'notifikasinya', 'parah', 'benar']	negatif
Stopwords Removal	['aplikasi', 'apa', 'verifikasi', 'x', 'gonta', 'ganti', 'nomor', 'telfon', 'masa', 'muncul', 'notifikasinya', 'parah', 'benar']	negatif
Stemming	['aplikasi', 'apa', 'verifikasi', 'x', 'gonta', 'ganti', 'nomor', 'telfon', 'masa', 'muncul', 'notifikasi', 'parah', 'benar']	negatif

3.2. Perhitungan Ekstraksi Fitur TF-IDF

Setelah melewati tahap pra-pemrosesan, teks diubah menjadi bentuk numerik menggunakan pendekatan TF-IDF (*Term Frequency-Inverse Document Frequency*). Berbeda dengan *CountVectorizer* yang hanya mencatat jumlah kemunculan kata, TF-IDF menghitung bobot kata berdasarkan frekuensi kemunculan dalam dokumen dan keberadaannya di seluruh kumpulan data, sehingga memberikan nilai lebih pada kata-kata tertentu yang jarang muncul di dokumen lain. Ekstraksi fitur dilakukan menggunakan *TfidfVectorizer* dari *library scikit-learn*. Data latih diproses dengan fit_transform() untuk membentuk kosakata sekaligus menghitung bobot, sementara data uji diproses dengan *transform()* menggunakan kosakata yang sama.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

```
vectorizer = TfidfVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

Gambar 3. Implementasi TF-IDF

3.3. Implementasi SMOTE (*Synthetic Minority Oversampling Technique*)

Setelah penerapan teknik SMOTE pada *dataset*, distribusi kelas menjadi seimbang dengan jumlah yang sama untuk kedua kategori sentimen, yaitu 5.233 data untuk sentimen positif dan 5.233 data untuk sentimen negatif.

```
smote = SMOTE(sampling_strategy='auto',
random_state=42)
X_train_smote, y_train_smote =
smote.fit_resample(X_train_vec, y_train)
class_distribution = y_train_smote.value_counts()
print(class_distribution)
```

Gambar 4. Implementasi SMOTE

3.4. Pemodelan

Setelah fitur diperoleh melalui TF-IDF, langkah berikutnya adalah membangun model menggunakan tiga algoritma klasifikasi pembelajaran mesin: *Logistic Regression*, Multinomial Naïve Bayes, dan *Support Vector Machine* (SVM). Ketiga algoritma ini dipilih karena sudah terbukti efektif dalam klasifikasi teks berdasarkan representasi vektor TF-IDF. Sebelum pelatihan model dilakukan, data dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian, seperti yang terlihat pada Gambar 4.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(df['preprocessed'], df['label'],
test_size=0.2, random_state=42, stratify=df['label'])
print('Jumlah data latih:', len(X_train))
print('Jumlah data uji:', len(X_test))
print('Data berhasil di split.')
```

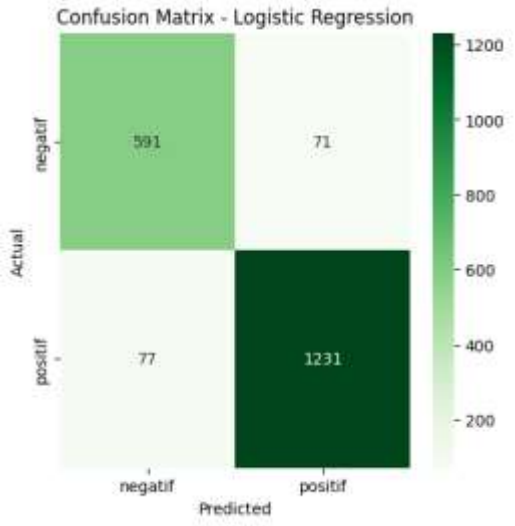
Gambar 5. Pembagian data *training* dan data *testing*

a. *Logistic Regression*

Pembangunan model *Logistic Regression* dilaksanakan menggunakan *library scikit-learn* dengan melakukan penyetelan parameter optimal melalui teknik *grid search*. Proses pencarian parameter terbaik dilakukan pada parameter C dengan rentang nilai [0.001, 0.01, 0.1, 1, 10, 100] dan parameter *penalty* dengan pilihan ['l1', 'l2'] menggunakan GridSearchCV dengan *cross-validation 5-fold*. Hasil pencarian parameter optimal menunjukkan bahwa kombinasi terbaik adalah C=100 dan *penalty*='l2' dengan *max_iter*=1000. Parameter C mengatur kekuatan regularisasi dimana nilai yang lebih besar berarti regularisasi yang lebih lemah, sedangkan penalty L2 menerapkan regularisasi *ridge*. Model kemudian dilatih menggunakan dataset latih yang telah melalui teknik SMOTE dan transformasi fitur TF-IDF.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Tahap evaluasi model *Logistic Regression* dilakukan dengan menguji kemampuan prediksi pada dataset uji yang telah diubah ke dalam format vektor TF-IDF. Output prediksi model dibandingkan dengan label yang sesungguhnya untuk menghasilkan *confusion matrix*.



Gambar 6. *Confustion matrix* model *Logistic Regression*

Analisis matriks kebingungan menunjukkan bahwa model *Logistic Regression* berhasil mengklasifikasikan 591 data dengan sentimen negatif dengan akurasi yang baik, meskipun ada 71 data negatif yang keliru diprediksi sebagai positif. Untuk sentimen positif, model ini mampu mengidentifikasi 1.231 data dengan tepat, sementara 77 data lainnya salah dikategorikan sebagai negatif. Dari hasil ini, metrik evaluasi seperti ketepatan, presisi, *recall*, dan skor F1 dapat dihitung untuk menilai performa keseluruhan model.

```
===== Kinerja Logistic Regression =====
Parameter Logistic Regression:  {'C': 100, 'penalty': 'l2'}
Akurasi: 0.9248730964467005
Classification Report:
              precision    recall  f1-score   support

     negatif       0.88      0.89      0.89       662
     positif       0.95      0.94      0.94      1308

    accuracy                           0.92      1970
   macro avg       0.92      0.92      0.92      1970
weighted avg       0.93      0.92      0.92      1970
```

Gambar 7. Evaluasi model *Logistic Regression*

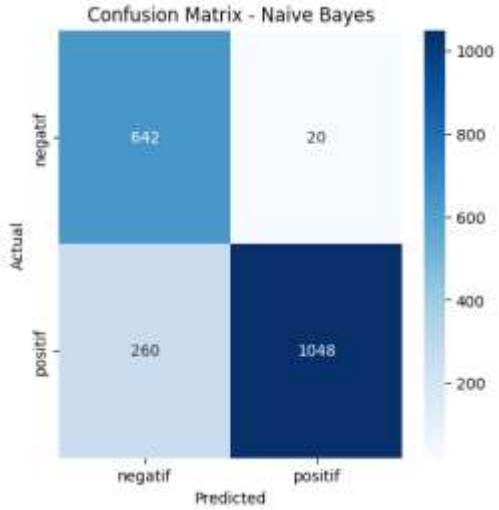
Hasil evaluasi menyeluruh model *Logistic Regression* dengan fitur TF-IDF menunjukkan kinerja yang sangat memuaskan dengan tingkat akurasi mencapai 92%. Untuk klasifikasi sentimen negatif, model mencapai *precision* 88%, *recall* 89%, dan *F1-score* 89%. Sedangkan pada klasifikasi sentimen positif, model memperoleh *precision* 95%, *recall* 94%, dan *F1-score* 94%.

b. *Naïve Bayes*

Model *Multinomial Naïve Bayes* diimplementasikan menggunakan *library scikit-learn* dengan optimasi *hyperparameter alpha* melalui GridSearchCV dan *5-fold cross-validation*. Nilai *alpha* yang diuji meliputi 0.1, 0.5, 1.0, 2.0, 5.0, dan 10.0, dengan hasil terbaik pada *alpha*=0.1 sebagai parameter *smoothing* untuk menangani kata yang tidak muncul di data latih. Model dilatih menggunakan data hasil SMOTE dan representasi fitur TF-IDF. Setelah

p-ISSN: 2986-3929
e-ISSN: 3032-1948

pelatihan, evaluasi dilakukan pada data uji yang telah melalui proses ekstraksi TF-IDF. Hasil prediksi dibandingkan dengan label aktual untuk menghasilkan *confusion matrix* sebagai dasar evaluasi kinerja model.



Gambar 8. *Confusion matrix* model *Multinomial Naïve Bayes*

Berdasarkan analisis matriks kebingungan, model Multinomial Naïve Bayes telah berhasil mengategorikan 642 data dengan sentimen negatif dengan akurasi yang tinggi, meskipun terdapat 20 data negatif yang salah dikenali sebagai positif. Untuk sentimen positif, model sukses mengidentifikasi 1.048 data dengan tepat, sedangkan 260 data lainnya salah klasifikasi sebagai negatif. Dari hasil tersebut, metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* dapat dihitung untuk menilai kinerja model secara keseluruhan.

```
===== Kinerja Naive Bayes =====
Parameter Naive Bayes:  {'alpha': 0.1}
Akurasi: 0.8578680203045685
Classification Report:
              precision    recall  f1-score   support

     negatif       0.71      0.97      0.82       662
     positif       0.98      0.80      0.88      1308

    accuracy                           0.86      1970
   macro avg       0.85      0.89      0.85      1970
weighted avg       0.89      0.86      0.86      1970
```

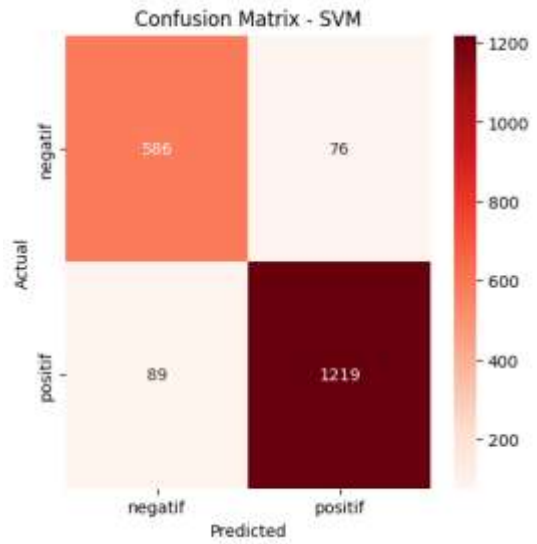
Gambar 9. Evaluasi model *Multinomial Naïve Bayes*

Evaluasi algoritma *Multinomial Naïve Bayes* yang dilatih menggunakan fitur TF-IDF menunjukkan kinerja yang cukup baik dengan akurasi 86%. Pada kelas negatif, model mencapai *precision* 71%, *recall* 97%, dan *F1-score* 82%. Sementara itu, untuk kelas positif, diperoleh *precision* 98%, *recall* 80%, dan *F1-score* 88%.

c. *Support Vector Machine* (SVM)

Support Vector Machine (SVM) dikembangkan menggunakan *library scikit-learn* dengan optimasi *hyperparameter* melalui GridSearchCV dan validasi silang *5-fold*. Parameter yang diuji meliputi C=[0.1, 1, 10, 100], *gamma*=[1, 0.1, 0.01, 0.001], dan kernel=['rbf', 'linear'], dengan hasil terbaik pada kombinasi C=100, gamma=0.1, dan kernel='rbf'. Model dilatih menggunakan data hasil SMOTE dan fitur TF-IDF. Evaluasi dilakukan pada data uji yang telah diproses serupa, dengan hasil prediksi dibandingkan terhadap label sebenarnya untuk menghasilkan *confusion matrix* sebagai dasar pengukuran performa.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 10. *Confusion matrix* model *Support Vector Machine* (SVM)

Berdasarkan analisis dari matriks kebingungan, model SVM mampu dengan tepat mengklasifikasikan 586 data negatif, meskipun ada 76 data negatif yang salah terdeteksi sebagai positif. Dalam hal data positif, model berhasil mengidentifikasi 1.219 data dengan akurat, sedangkan 89 data keliru dikategorikan sebagai negatif. Dari hasil tersebut, metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* dapat dihitung guna menilai kinerja model secara keseluruhan.

```
===== Kinerja SVM =====
Parameter SVM:  {'C': 100, 'gamma': 0.1, 'kernel': 'rbf'}
Akurasi: 0.916243654822335
Classification Report:
              precision    recall  f1-score   support

     negatif       0.87      0.89      0.88       662
     positif       0.94      0.93      0.94      1308

    accuracy                           0.92      1970
   macro avg       0.90      0.91      0.91      1970
weighted avg       0.92      0.92      0.92      1970
```

Gambar 11. Evaluasi model *Support Vector Machine* (SVM)

Evaluasi menyeluruh terhadap SVM yang dilatih dengan fitur TF-IDF menunjukkan hasil yang sangat baik, dengan akurasi sebesar 92%. Pada sentimen negatif, model mencapai *precision* 87%, *recall* 89%, dan *F1-score* 88%. Sementara itu, untuk sentimen positif, diperoleh *precision* 94%, *recall* 93%, dan *F1-score* 94%.

4. Kesimpulan

Berdasarkan analisis sentimen yang dilakukan pada ulasan aplikasi *mobile* dengan memanfaatkan tiga algoritma pembelajaran mesin, *Logistic Regression* dan *Support Vector Machine* menunjukkan kinerja terbaik dengan tingkat akurasi mencapai 92%, sementara *Multinomial Naïve Bayes* mencatatkan akurasi sebesar 86%. Penggunaan SMOTE terbukti efektif dalam mengatasi ketidakseimbangan kelas dengan menyamakan jumlah data menjadi 5.233 untuk setiap kategori sentimen. Di samping itu, proses ekstraksi fitur melalui TF-IDF berhasil mengubah data teks menjadi representasi numerik yang sesuai untuk pemrosesan oleh model pembelajaran mesin. Penelitian ini menunjukkan bahwa *Logistic Regression* dan *Support Vector Machine* merupakan metode yang paling optimal untuk klasifikasi sentimen ulasan aplikasi *mobile* dengan tingkat *precision* dan *recall* yang konsisten, sehingga dapat dijadikan acuan bagi

pengembang aplikasi dalam memahami *feedback* pengguna secara otomatis untuk meningkatkan kualitas layanan berdasarkan analisis sentimen yang akurat. Penelitian selanjutnya dapat mengeksplorasi penggunaan metode *deep learning* atau *ensemble learning* untuk meningkatkan performa klasifikasi sentimen lebih lanjut.

Daftar Pustaka

[1] R. M. Fadhillah, “Analisis Sentimen Berbasis Aspek pada Review Aplikasi ChatGPT Menggunakan Multinomial Naive Bayes,” 2024.

[2] N. A. Salsabila, “ANALISIS SENTIMEN PADA MEDIA SOSIAL TWITTER TERHADAP TOKOH GUS DUR MENGGUNAKAN METODE NAÏVE BAYES DAN SUPPORT VECTOR MACHINE (SVM),” 2022.

[3] F. S. Lestari, Harliana, M. M. Huda, and T. Prabowo, “Sentiment Analysis of iPusnas Application Reviews on Google Play Using Support Vector Machine,” in *The 1st International Seminar August 2022:The Changing Role of Knowledge and Living Sustainability in ASEAN Community*, Universitas Nahdlatul Ulama Blitar, 2022, pp. 178–188. doi: 10.29407/int.v1i1.2656.

[4] D. Septiani and I. Isabela, “ANALISIS TERM FREQUENCY INVERSE DOCUMENT FREQUENCY (TF-IDF) DALAM TEMU KEMBALI INFORMASI PADA DOKUMEN TEKS,” *SINTESIA: Jurnal Sistem dan Teknologi Informasi Indonesia*, vol. 1, no. Vol. 1 No. 2 (2022): SINTESIA, pp. 81–88, Mar. 2022, Accessed: Jul. 01, 2025. [Online]. Available: https://journal.unj.ac.id/unj/index.php/SINTESIA/article/view/39364

[5] S. Sofyan and A. Prasetyo, “Penerapan Synthetic Minority Oversampling Technique (SMOTE) Terhadap Data Tidak Seimbang Pada Tingkat Pendapatan Pekerja Informal Di Provinsi D.I. Yogyakarta Tahun 2019,” 2021.

[6] F. D. Pramakrisna, F. D. Adhinata, and N. A. F. Tanjung, “Aplikasi Klasifikasi SMS Berbasis Web Menggunakan Algoritma Logistic Regression,” *Teknika*, vol. 11, no. 2, pp. 90–97, Jun. 2022, doi: 10.34148/teknika.v11i2.466.

[7] C. Dewi, R. C. Chen, H. J. Christanto, and F. Cauteruccio, “Multinomial Naïve Bayes Classifier for Sentiment Analysis of Internet Movie Database,” *Vietnam Journal of Computer Science*, vol. 10, no. 4, pp. 485–498, Nov. 2023, doi: 10.1142/S2196888823500100.

[8] Shahmirul Hafizullah Imanuddin, Kusworo Adi, and Rahmat Gernowo, “Sentiment Analysis on Satusehat Application Using Support Vector Machine Method,” *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 5, no. 3, pp. 143–149, Jul. 2023, doi: 10.35882/jeemi.v5i3.304.

[9] M. S. Anggreany, “Confusion Matrix.” Accessed: Jul. 02, 2025. [Online]. Available: https://socs.binus.ac.id/2020/11/01/confusion-matrix/.