JURNAL FASILKOM
Volume 15 No. 3 | Desember 2025: 610-620

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Analisis Sentimen Ulasan *Google Play Store*: Studi Komparatif Algoritma SVM, *Naïve Bayes*, dan *Logistic Regression*

Charles Dometian[1], Singgih Jatmiko[2]
[1,2]Program Studi Informatika, Fakultas Teknik Industri, Universitas Gunadama
[1]charlesdometian@student.gunadarma.ac.id, [2]singgih@staff.gunadarma.ac.id*

Abstract

This research aims to compare Support Vector Machine (SVM), Naïve Bayes, and Logistic Regression methods in sentiment analysis of app reviews on Google Play Store to identify the best method based on accuracy, precision, recall, and F1-Score using 2000 GoPay and LinkAja reviews from Google Play Store respectively. The methodology consists of six stages, namely, data collection, labeling method evaluation, preprocessing evaluation, SMOTE testing to overcome imbalanced data, hyperparameter tuning optimization, and consistency validation with a combination of TF-IDF and three classification methods. The data were split using an 80:20 ratio, with 80% of the data used for training and 20% for testing. Experimental results show SVM gives the best performance with 93% accuracy, 92% precision, 93% recall, and 92% F1-Score on the GoPay dataset due to its ability to find the optimal hyperplane, followed by Logistic Regression with 92% accuracy and the third Naïve Bayes despite identical accuracy but showing greater bias towards the majority class. Validation using the LinkAja dataset proves SVM still maintains the best performance with 95% accuracy, so the research concludes SVM is the best method for sentiment analysis of app reviews on the Google Play Store which is proven to provide optimal and consistent performance.

Keywords: SVM, naïve bayes, logistic regression, sentiment analysis, google play store

Abstrak

Penelitian ini bertujuan membandingkan metode *Support Vector Machine* (SVM), *Naïve Bayes*, dan *Logistic Regression* dalam analisis sentimen ulasan aplikasi di *Google Play Store* untuk mengidentifikasi metode terbaik berdasarkan akurasi, presisi, *recall*, dan *F1-Score* menggunakan masing-masing 2000 ulasan *GoPay* dan LinkAja dari *Google Play Store*. Metodologi terdiri dari enam tahapan yakni, pengumpulan data, evaluasi metode pelabelan, evaluasi *preprocessing*, pengujian SMOTE untuk mengatasi ketidakseimbangan data, optimasi hyperparameter tuning, dan validasi konsistensi dengan kombinasi TF-IDF dan tiga metode klasifikasi. Data dibagi menggunakan rasio 80:20, dengan 80% data sebagai data pelatihan (*training*) dan 20% sebagai data pengujian (*testing*). Hasil eksperimen menunjukkan SVM memberikan kinerja terbaik dengan akurasi 93%, *precision* 92%, *recall* 93%, dan *F1-Score* 92% pada dataset GoPay karena kemampuan menemukan hyperplane optimal, diikuti *Logistic Regression* dengan akurasi 92% dan *Naïve Bayes* ketiga meskipun akurasi identik namun menunjukkan bias lebih besar terhadap kelas mayoritas. Validasi menggunakan dataset LinkAja membuktikan SVM tetap mempertahankan kinerja terbaik dengan akurasi 95%, sehingga penelitian menyimpulkan SVM merupakan metode terbaik untuk analisis sentimen ulasan aplikasi di *Google Play Store* yang terbukti memberikan kinerja optimal dan konsisten.

Kata kunci: analisis sentimen, *logistic regression*, SVM, *Naïve Bayes*

1. Pendahuluan

Dengan pesatnya perkembangan berbagai metode pembayaran digital di Indonesia, *e-wallet* atau dompet digital telah menjadi metode pembayaran yang banyak digunakan oleh masyarakat. Penelitian ini menggunakan GoPay dan LinkAja sebagai studi kasus untuk analisis sentimen, karena keduanya merupakan dompet digital yang banyak digunakan dengan karakteristik pengguna dan layanan yang berbeda. GoPay merupakan uang elektronik atau dompet digital dalam bentuk saldo yang bisa digunakan sebagai metode pembayaran untuk berbagai layanan yang disediakan oleh GoJek [1] Sementara itu, LinkAja hadir dengan karakteristik berbeda sebagai e-wallet yang didukung konsorsium 10 BUMN dengan jangkauan ke 34 provinsi di Indonesia, berorientasi pada inklusivitas keuangan nasional [2].

Untuk menganalisis opini pengguna terhadap aplikasi *e-wallet* tersebut, diperlukan analisis sentimen. Analisis sentimen merupakan salah satu bagian dari *Natural Language Processing* yang berhubungan erat dengan teknik *machine learning* [3]. Metode ini berfungsi untuk mengekstrak, mengolah, dan menganalisis data tekstual secara otomatis untuk mendeteksi dan mengklasifikasikan sentimen atau muatan emosional yang terkandung dalam beragam bentuk pendapat [4]. Namun, penelitian terdahulu menunjukkan hasil inkonsisten dalam perbandingan metode klasifikasi, dimana *Naïve Bayes* unggul pada ulasan *Grab* dan aplikasi *mobile* (88,5% dan 91,6%) [5][6], SVM unggul pada *LinkedIn* dan *TikTokShop* (92% dan 81%) [7][8], dan Logistic Regression terbaik pada TikTok (84%) [9].

Inkonsistensi ini diduga disebabkan perbedaan konfigurasi penelitian seperti metode pelabelan, *preprocessing*, tidak adanya penanganan data tidak seimbang dan tuning *hyperparameter*. Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi secara komparatif performa SVM, *Naïve Bayes*, dan *Logistic*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Regression dengan menerapkan optimasi SMOTE, guna menentukan algoritma yang paling handal dalam menangani karakteristik ulasan aplikasi keuangan yang cenderung tidak seimbang

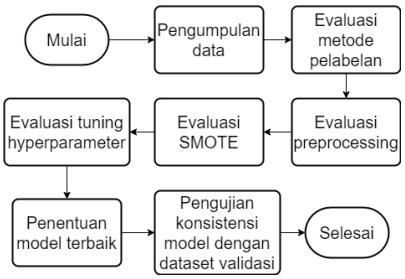
Meskipun sejumlah penelitian sebelumnya telah membandingkan metode *Support Vector Machine, Naïve Bayes,* dan *Logistic Regression* dalam analisis sentimen, sebagian besar masih menggunakan konfigurasi yang terbatas. Penelitian ini berbeda dengan menekankan evaluasi menyeluruh pada setiap tahapan analisis, mencakup metode pelabelan, variasi dan urutan *preprocessing*, penanganan *imbalanced data*, optimasi *hyperparameter*, serta pengujian konsistensi model pada dataset yang berbeda. Pendekatan ini memberikan gambaran yang lebih komprehensif mengenai kinerja dan stabilitas metode klasifikasi yang digunakan.

2. Metode Penelitian

2.1. Alur tahapan penelitian

Alur metodologi penelitian terdiri dari enam tahapan berurutan yang dirancang untuk mengembangkan model klasifikasi sentimen dengan kinerja optimal, sebagaimana ditampilkan pada Gambar 1. Setiap tahap evaluasi dirancang untuk menentukan konfigurasi terbaik melalui perbandingan sistematis menggunakan kombinasi TF-IDF dengan tiga metode klasifikasi (*SVM, Naive Bayes, dan Logistic Regression*) berdasarkan metrik akurasi, presisi, *recall*, dan *f1-score.*

Tahap akhir dilakukan validasi konsistensi menggunakan dataset independen ulasan aplikasi *GoPay* dan LinkAja dengan menerapkan pembagian data *training* dan testing, ekstraksi fitur menggunakan TF-IDF, serta evaluasi kinerja model berdasarkan metrik akurasi, *precision, recall,* dan *F1-Score* untuk memverifikasi kemampuan generalisasi model.



Gambar 1. Alur Tahapan Penelitian

2.2. Pengumpulan Data

Pengumpulan data ulasan dilakukan menggunakan API *Google-Play-Scraper* melalui library *google-play-scraper* untuk mengekstrak ulasan publik aplikasi GoPay dan LinkAja. Teknik pengambilan sampel dilakukan secara *purposive* dengan mengambil ulasan yang tersedia di *Google Play Store* berdasarkan tingkat relevansi dan menggunakan bahasa Indonesia. Jumlah ulasan yang dikumpulkan dibuat sama untuk kedua aplikasi, yaitu masing-masing sebanyak 2.000 ulasan, yang diambil melalui 10 iterasi dengan 200 ulasan pada setiap iterasi [10]. Data yang diekstrak mencakup isi ulasan dan rating (skala 1–5), kemudian disaring, diorganisasi, dan disimpan dalam format CSV untuk analisis selanjutnya.

2.3. Evaluasi Metode Pelabelan

Penelitian ini membandingkan dua metode pelabelan, yaitu *rating-based* dan *lexicon-based*. Pada metode *rating-based*, setiap ulasan diberi label sentimen berdasarkan nilai rating yang diberikan pengguna, dimana rating 4–5 diklasifikasikan sebagai sentimen positif dan rating 1–3 sebagai sentimen negatif. Metode ini mudah diterapkan karena memanfaatkan metadata ulasan, namun memiliki keterbatasan karena rating tidak selalu mencerminkan isi teks ulasan secara akurat. Sementara itu, metode *lexicon-based* dilakukan dengan menghitung skor sentimen setiap ulasan berdasarkan kamus leksikon Bahasa Indonesia, dimana setiap kata memiliki bobot *polarity score* pada rentang −5 hingga +5. Skor total ulasan diperoleh dari akumulasi bobot kata, kemudian digunakan untuk menentukan label sentimen positif atau negatif. Pendekatan ini lebih mempertimbangkan konteks teks, tetapi sangat bergantung pada kelengkapan dan kualitas kamus leksikon yang digunakan [11] .

2.4. Pembagian Data (Data *Split*)

Pembagian data dilakukan sebelum proses pemodelan untuk memastikan evaluasi kinerja model dilakukan secara objektif. Dataset yang telah melalui tahap pelabelan dibagi menggunakan rasio 80:20, dengan 80% data digunakan sebagai data pelatihan (*training*) dan 20% sebagai data validasi (*validation*), serta pengaturan *random state* sebesar 100 untuk menjaga konsistensi pembagian data [12],. Pada penelitian ini, data validasi berfungsi sebagai data pengujian (*testing*), sehingga tidak digunakan dataset testing terpisah.

Evaluasi kinerja model dilakukan menggunakan ekstraksi fitur TF-IDF [13] dan pemodelan dengan S*upport Vector Machine*, Multinomial *Naive Bayes*, dan *Logistic Regression* tanpa *preprocessing* tambahan maupun optimasi parameter. Konfigurasi ini diterapkan secara konsisten untuk memastikan bahwa perbedaan kinerja yang diperoleh terutama mencerminkan pengaruh metode pelabelan yang digunakan.

2.5. Evaluasi *Preprocessing*

Text preprocessing merupakan tahap awal untuk menyiapkan data teks tidak terstruktur menjadi terstruktur agar optimal untuk dianalisis [14], dimana kelengkapan dan urutan tahap *preprocessing* dapat memengaruhi representasi fitur dan berdampak pada akurasi model [15]. Oleh karena itu, penelitian ini mengevaluasi 10 kombinasi *preprocessing* berbeda dengan variasi kelengkapan (3–6 tahapan) dan urutan penerapan untuk mengidentifikasi konfigurasi *preprocessing* yang paling optimal dan konsisten pada berbagai metode klasifikasi. Seluruh kombinasi diuji menggunakan pembagian data *training–validation*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

80:20, ekstraksi fitur TF-IDF, serta pemodelan SVM, *Multinomial Naive Bayes,* dan *Logistic Regression* tanpa *tuning* parameter, sehingga perbedaan kinerja yang dihasilkan dapat dikaitkan langsung dengan pengaruh *preprocessing*.

Sepuluh kombinasi *preprocessing* yang dievaluasi mencakup variasi kelengkapan dan urutan penerapan yang berbeda. Kombinasi pertama menggunakan *cleaning, casefolding, dan tokenizing*. Kombinasi kedua menambahkan *normalization* setelah casefolding dengan urutan *cleaning*, *casefolding*, *normalization*, dan *tokenizing*. Kombinasi ketiga mengubah urutan menjadi *tokenizing, cleaning, casefolding,* dan *normalization*. Kombinasi keempat mengintegrasikan *stopword removal* dan *stemming* dengan urutan *casefolding*, *cleaning*, *stopword*, *stemming*, dan *normalization*. Kombinasi kelima menggunakan *tokenizing, casefolding, cleaning, normalization,* dan *stopword.* Kombinasi keenam merupakan konfigurasi lengkap dengan *cleaning, casefolding, tokenizing, normalization, stopword,* dan *stemming.* Kombinasi ketujuh menggunakan urutan *cleaning, casefolding, normalization, stopword, stemming,* dan *tokenizing*. Kombinasi kedelapan menerapkan *tokenizing, casefolding, cleaning, normalization, stopword,* dan *stemming*. Kombinasi kesembilan menggunakan pendekatan terbalik dengan *stopword, normalization, stemming, cleaning, tokenizing,* dan *casefolding*. Kombinasi kesepuluh menerapkan *stemming, stopword, normalization, tokenizing, casefolding,* dan *cleaning* untuk mengevaluasi dampak urutan *preprocessing* terhadap kinerja model.

Evaluasi pada tahap *preprocessing* dilakukan dengan menghitung rata-rata akurasi dan standar deviasi dari kinerja model untuk setiap kombinasi *preprocessing* yang diuji. Perhitungan rata-rata akurasi digunakan untuk memperoleh gambaran umum kinerja model pada masing-masing kombinasi *preprocessing*, sedangkan standar deviasi digunakan untuk mengukur tingkat konsistensi kinerja model antar metode klasifikasi. Rata-rata akurasi dihitung menggunakan rumus [16]:

$$\bar{X} = \frac{\sum_{i=1}^{n} x_i}{n} \quad (1)$$

dengan $\bar{X}$ adalah nilai rata-rata akurasi dari seluruh model, x_i adalah nilai akurasi ke-i, dan n adalah jumlah data yang dihitung.

Selanjutnya, standar deviasi dihitung untuk mengevaluasi kestabilan hasil kinerja model pada setiap kombinasi preprocessing, dimana nilai standar deviasi yang rendah menunjukkan kinerja yang lebih konsisten. Standar deviasi dihitung menggunakan rumus [16]:

$$\sigma = \sqrt{\sum \frac{(X_i - \mu)^2}{N}} \quad (2)$$

dengan σ adalah standar deviasi, X_i adalah nilai akurasi ke-i, μ adalah nilai rata-rata akurasi, dan N adalah jumlah total data.

Pemilihan kombinasi *preprocessing* terbaik dilakukan pada tahap ini dengan menjadikan standar deviasi sebagai parameter utama dan rata-rata akurasi sebagai parameter pendukung, karena kombinasi *preprocessing* yang optimal diharapkan tidak hanya menghasilkan akurasi tinggi tetapi juga stabil pada berbagai model klasifikasi.

2.6. Evaluasi *SMOTE*

Ketidakseimbangan kelas (*class imbalance*) merupakan kondisi dimana salah satu kategori dalam dataset memiliki jumlah sampel yang sangat berbeda jauh dengan kategori lainnya, dimana kategori dengan jumlah sampel terbanyak disebut kelas mayoritas dan kategori dengan jumlah sampel paling sedikit dinamakan kelas minoritas. Kondisi ini berdampak signifikan terhadap akurasi hasil prediksi metode pembelajaran mesin [17]. Untuk mengukur tingkat ketidakseimbangan *dataset*, digunakan perhitungan *Imbalance Ratio* (IR) [18]:

$$Imbalance\ Ratio\ (\%) = \frac{N_{minority}}{N_{majority}+N_{minority}} \quad (3)$$

Klasifikasi tingkat ketidakseimbangan dapat dikategorikan berdasarkan proporsi kelas minoritas, seperti tunjukkan pada Tabel 1 [19].

Tabel 1.Tabel Kategori Tingkat Ketidakseimbangan Dataset

Tingkat Ketidakseimbangan	Proporsi Kelas Minoritas
Sedikit	>50% dari dataset
Ringan	20-40% dari dataset
Sedang	1-20% dari dataset
Ekstrim	<1% dari dataset

SMOTE (*Synthetic Minority Over-sampling Technique*) adalah metode *oversampling* yang menyeimbangkan jumlah data kelas minoritas dengan menghasilkan data sintetis baru berdasarkan *K-Nearest Neighbors* dari data minoritas asli, mengurangi risiko overfitting dengan menciptakan data baru yang mirip namun tidak identik [19], namun penerapannya harus hati-hati untuk menghindari data *leakage* yang menyebabkan bias signifikan pada metrik kinerja [20]. Evaluasi SMOTE dilakukan dengan membandingkan kinerja model dengan dan tanpa SMOTE menggunakan *sampling strategy* 0.8 dan *k-neighbors* sebesar 5. Penerapan SMOTE dilakukan hanya pada data training setelah proses pembagian data dan transformasi fitur TF-IDF, sementara data validasi/testing dibiarkan dalam kondisi asli tanpa *oversampling*. Pendekatan ini diterapkan untuk mencegah terjadinya *data leakage* dan memastikan evaluasi kinerja model tetap objektif.

2.7. Evaluasi *Tuning Hyperparameter*

Penelitian ini menggunakan tiga metode pembelajaran mesin, yaitu *Support Vector Machine (SVM), Naïve Bayes,* dan *Logistic Regression* untuk klasifikasi teks.

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Proses *hyperparameter tuning* dilakukan menggunakan metode Random Search melalui *RandomizedSearchCV* karena lebih efisien dibandingkan *Grid Search* dalam mengeksplorasi ruang parameter. Evaluasi membandingkan kinerja model hasil tuning dan non-tuning, dengan penerapan SMOTE hanya pada data training, sementara data validasi dibiarkan dalam kondisi original untuk mengukur kemampuan generalisasi model.

Support Vector Machine merupakan metode pembelajaran terawasi yang bekerja dengan menemukan *hyperplane* optimal yang memaksimalkan margin antara dua kelas dalam ruang fitur [21], hanya bergantung pada *support vectors* dan mampu menangani data yang tidak terpisahkan secara linier melalui teknik kernel [22], dengan tuning hyperparameter meliputi parameter regularisasi C (0.1-0.3), kernel linear dan *sigmoid*, parameter gamma ('scale' dan 'auto'), serta *class weight* (*None* dan *'balanced'*) [23] menggunakan S*tratified KFold 5-fold cross-validation.*

Naive Bayes adalah metode klasifikasi probabilistik yang menghitung probabilitas berdasarkan frekuensi dan kombinasi nilai [24], dengan asumsi independensi kondisional antar atribut [25], mengevaluasi tiga varian yaitu MultinomialNB untuk klasifikasi teks, *GaussianNB* untuk data kontinu, dan BernoulliNB untuk keberadaan fitur [26], dengan parameter *tuning* meliputi alpha, fit_prior, var_smoothing pada *GaussianNB*, dan *binarize* pada *BernoulliNB*. *Logistic Regression* menggunakan fungsi sigmoid untuk menghasilkan probabilitas 0-1 [27] dengan *tuning hyperparameter* melalui 500 kombinasi parameter menggunakan 7-*fold StratifiedKFold cross-validation*, mencakup *regularization strength* C (1.75-2.3), *penalty* L1 dan L2, serta solver 'saga', *'liblinear'*, dan *'lbfgs'* [28].

Hasil *tuning hyperparameter* terbaik dari ketiga model akan dibandingkan dengan versi *non-tuning* untuk mengevaluasi efektivitas proses optimasi parameter. Perbandingan dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score* pada dataset yang sama untuk menentukan apakah kompleksitas tambahan dari proses tuning memberikan peningkatan kinerjayang signifikan.

2.8. Penentuan dan Pengujian Konsistensi Model

Evaluasi kinerja model klasifikasi dilakukan menggunakan *classification report* untuk menentukan metode terbaik di antara SVM, Naïve Bayes, dan Logistic Regression. *Classification report* memberikan laporan komprehensif menggunakan *scikit-learn* [12] yang mencakup metrik evaluasi berupa akurasi, precision, recall, dan F1-Score untuk menganalisis detail kinerja dan mengidentifikasi metode terbaik.

Setelah memperoleh konfigurasi optimal berdasarkan tahapan evaluasi menggunakan dataset GoPay, dilakukan pengujian konsistensi untuk memverifikasi stabilitas kinerja ketiga metode klasifikasi. Dataset LinkAja digunakan sebagai dataset validasi independen dengan menerapkan pipeline analisis yang identik, meliputi metode pelabelan yang sama, preprocessing terbaik yang sama, skema penanganan *imbalanced data* dengan SMOTE yang sama (hanya pada data training), serta konfigurasi *hyperparameter* hasil optimasi sebelumnya. Dengan menerapkan proses yang sama secara konsisten, evaluasi dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *F1-Score* untuk memastikan bahwa perbandingan kinerja antar metode tetap adil dan dapat diandalkan pada dataset yang berbeda.

3. Hasil dan Pembahasan

3.1. Hasil Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil ulasan dari Google Play Store pada aplikasi GoPay dan LinkAja dengan bantuan library *google-play-scraper*. Data diambil melalui proses scraping dengan jumlah data yang dikumpulkan masing-masing sebanyak 2.000 ulasan, di mana data ulasan GoPay diambil pada 8 Oktober 2025 dan data ulasan LinkAja diambil pada 30 Oktober 2025. Setiap data mencakup isi ulasan dan rating. Hasil pengumpulan data disimpan dalam format CSV yang dapat dilihat pada Tabel 2.

Tabel 1. Hasil Pengumpulan Data Ulasan

Ulasan GoPay	*Rating*	Ulasan LinkAja	*Rating*
aplikasi nya bagus dan transfer nya pun cepat	5	sangat membantu dan mudah di gunakan	5
tolong diperbaiki lagi, ini ada bug dimana gak bisa ganti foto profil	3	masuk login nya susah lama udah masuk keluar lagi	1
gopay terlalu banyak gangguanya, buat bayar parkir susah amat	1	baru upgrade, mau byr halo. eh... koq bisa ngga ada menunya bayar halo . koq bisa ya	2
scan qris, saldo terpotong tapi tidak masuk dana nya.	2	APK mudah digunakan, pengisian data nya jg mudah dan sangat membantu, thnxs..	5
bagus	5	Buka aplikasinya stuck terus di logo	1

3.2. Hasil Evaluasi Metode Pelabelan

Setelah mendapatkan data ulasan, data tersebut dilakukan pelabelan untuk mengidentifikasi serta mengelompokkan apakah ulasan yang ada terkategori sentimen positif atau negatif. Terdapat 2 metode pelabelan yang digunakan pada data ulasan ini yakni *Rating-based* dan *Lexicon-based*. Hasil distribusi pelabelan *rating* dan *lexicon* dapat dilihat pada Tabel 3 dan 4.

Tabel 2. Distribusi Pelabelan dengan *Rating*

Label	Jumlah
Negatif	1362
Positif	638

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Tabel 3. Distribusi Pelabelan dengan Lexicon

Label	Jumlah
Negatif	1394
Positif	606

Terdapat perbedaan jumlah pelabelan sentimen positif dan negatif pada kedua metode pelabelan. Pada metode *rating-based*, diperoleh 1362 ulasan dengan sentimen negatif dan 638 ulasan dengan sentimen positif, sedangkan pada metode *lexicon-based* terdapat 1394 ulasan negatif dan 606 ulasan positif. Dominasi sentimen negatif ini dipengaruhi oleh karakteristik ulasan aplikasi di *Google Play Store*, dimana pengguna cenderung lebih aktif memberikan ulasan ketika mengalami kendala atau ketidakpuasan terhadap layanan. Selain itu, pendekatan pelabelan berbasis rating dan leksikon sama-sama sensitif terhadap ulasan bernada keluhan, sehingga memperkuat proporsi sentimen negatif yang teridentifikasi. Meski demikian, kedua hasil tersebut sama-sama menunjukkan hasil ulasan dengan sentimen negatif lebih banyak dari positif. Contoh hasil pelabelan rating dan lexicon dapat dilihat pada Tabel 5.

Tabel 4.Hasil Pelabelan dengan *Rating* dan *Lexicon*

Ulasan	Label dengan *Rating*	Label dengan *Lexicon*
kecewa verifikasi wajah gagal terus, padahal udah terang ngikutin petunjuk tetep gagal , tolong dong gmn ini	Negatif	Negatif
lupa pin, ribet banget padahal no hp masih aktif, tapi tetep gabisa login, saldo gw masih nyangkut tuh udah 4 bulan	Negatif	Negatif
Setidaknya kita punya Dompet digital yang sewaktu waktu dibutuhkan bisa dimanfaatkan ok Terimakasih	Positif	Negatif
dengan gopay ,transfer,beli token beli paket data menjadi sangat mudah	Positif	Positif
Aplikasinya bagus transfer uangnya cepat dan gampang digunakan	Positif	Negatif

Setelah proses pelabelan, kedua hasil pelabelan dievaluasi secara tidak langsung melalui kinerja model klasifikasi. Evaluasi dilakukan dengan mengukur akurasi model klasifikasi, bukan akurasi pelabelan otomatis, menggunakan pembagian data latih dan validasi dengan rasio 80:20 (*random state* 100), dimana dari total 2.000 data diperoleh 1.600 data latih dan 400 data validasi. Pada tahap ini, pemodelan dilakukan tanpa *preprocessing* tambahan, SMOTE, maupun tuning *hyperparameter* untuk memastikan bahwa perbedaan kinerja yang dihasilkan mencerminkan pengaruh metode pelabelan. Hasil perbandingan kinerja model terhadap kedua metode pelabelan disajikan pada Tabel 6 dan Tabel 7.

Tabel 5. Kinerja Model dengan Pelabelan *Rating*

Model	*Accuracy*	*F1-Score*	*Recall*	*Precision*
SVM	90%	90%	90%	91%
Naïve Bayes	90%	89%	90%	90%
Logistic Regression	89%	89%	89%	90%

Tabel 6. Kinerja Model dengan Pelabelan *Lexicon*

Model	*Accuracy*	*F1-Score*	*Recall*	*Precision*
SVM	81%	80%	81%	81%
Naïve Bayes	74%	67%	74%	77%
Logistic Regression	78%	74%	78%	79%

Berdasarkan hasil evaluasi kinerja model klasifikasi, metode pelabelan data berbasis rating menunjukkan kinerja yang lebih baik dibandingkan metode pelabelan data berbasis leksikon pada seluruh metrik evaluasi. Pada pelabelan rating, ketiga model menghasilkan akurasi berkisar antara 89%–90%, sedangkan pada pelabelan leksikon akurasi yang diperoleh berada pada rentang 74%–81%. Pola serupa juga terlihat pada nilai F1-*Score, recall,* dan *precision*, dimana pelabelan rating secara konsisten menghasilkan nilai yang lebih tinggi pada ketiga model. Hasil ini menunjukkan bahwa metode pelabelan data memiliki pengaruh signifikan terhadap kinerja model secara keseluruhan, sehingga pada tahap selanjutnya penelitian ini menggunakan metode pelabelan data berbasis rating karena terbukti memberikan kinerja paling optimal.

3.3. Hasil Evaluasi *Preprocessing*

Setelah menentukan metode pelabelan, data yang ada selanjutnya akan diproses dengan *preprocessing* agar data lebih siap untuk digunakan dalam pelatihan model. Terdapat 10 kombinasi *preprocessing* yang akan diuji pada data ulasan, hasilnya dapat dilihat pada Tabel 8.

Tabel 7. Kinerja Model Terhadap Variasi *Preprocessing* Teks

Percobaan	Tahapan *Preprocessing*	Akurasi		
		SVM	NB	LR
1	*Cleaning - Casefolding - Tokenizing*	90%	89%	88%
2	*Cleaning - Casefolding - Normalization - Tokenizing*	91%	90%	89%
3	*Tokenizing - Cleaning - Casefolding - Normalization*	91%	89%	89%
4	*Casefolding - Cleaning - Stopword - Stemming - Normalization*	90%	91%	90%
5	*Tokenizing - Casefolding - Cleaning - Normalization - Stopword*	89%	90%	88%
6	*Cleaning - Casefolding - Tokenizing - Normalization - Stopword - Stemming*	91%	92%	91%
7	*Cleaning - Casefolding - Normalization - Stopword - Stemming - Tokenizing*	91%	90%	90%

P-ISSN : 2089-3353
E-ISSN : 2808-9162

8	*Tokenizing - Casefolding - Cleaning - Normalization - Stopword - Stemming*	91%	90%	90%
9	*Stopword - Normalization - Stemming - Cleaning - Tokenizing - Casefolding*	91%	90%	89%
10	*Stemming - Stopword - Normalization - Tokenizing - Casefolding - Cleaning*	91%	91%	90%

Sebagai ilustrasi, nilai akurasi pada setiap algoritma dihitung dari proporsi data validasi yang berhasil diklasifikasikan dengan benar, dengan prosedur perhitungan yang sama diterapkan pada SVM, *Naïve Bayes,* dan *Logistic Regression.*

Pada proses ini untuk menentukan tahap *preprocessing* yang paling optimal untuk ketiga model maka perlu dilakukan perhitungan rata-rata akurasi untuk memperoleh gambaran menyeluruh tentang kinerja kombinasi *preprocessing* terhadap ketiga model, serta standar deviasi untuk mengukur tingkat konsistensi hasil evaluasi model dalam satu kombinasi *preprocessing* yang sama. Hasil dari urutan kombinasi *preprocessing* optimal dapat dilihat pada Tabel 9.

Tabel 8. Urutan Kombinasi *Preprocessing* Optimal terhadap model berdasarkan Standar Deviasi dan Rata-Rata Akurasi

Urutan	Tahap *Preprocessing*	Standar Deviasi	Rata-Rata Akurasi
1	*Cleaning - case folding - tokenizing - normalization - stopword - stemming*	0,58%	91,33%
2	*Stemming - stopword - normalization - tokenizing - case folding - cleaning*	0,58%	90,67%
3	*Casefolding - cleaning - stopword - stemming - normalization*	0,58%	90,33%
4	*Cleaning - case folding - normalization - stopword - stemming - tokenizing*	0,58%	90,33%
5	*Tokenizing - case folding - cleaning - normalization - stopword - stemming*	0,58%	90,33%
6	*Cleaning - case folding - normalization - tokenizing*	1,00%	90,00%
7	*Stopword - normalization - stemming - cleaning - tokenizing - case folding*	1,00%	90,00%
8	*Tokenizing - case folding - cleaning - normalization - stopword*	1,00%	89,00%
9	*Cleaning - case folding - tokenizing*	1,00%	89,00%
10	*Tokenizing - cleaning - case folding - normalization*	1,15%	89,67%

Berdasarkan evaluasi kombinasi *preprocessing* menggunakan nilai rata-rata akurasi dan standar deviasi sesuai Persamaan (1) dan (2), kombinasi *preprocessing cleaning–casefolding–tokenizing–normalization–stopword–stemming* dipilih sebagai konfigurasi paling optimal. Kombinasi ini menghasilkan nilai rata-rata akurasi tertinggi pada ketiga model, yaitu berada pada kisaran 91%–92%, serta menunjukkan nilai standar deviasi yang relatif rendah dibandingkan kombinasi lain, yang mengindikasikan kinerja yang lebih konsisten antar model. Meskipun perbedaan akurasi antar kombinasi preprocessing tidak terlalu besar, pemilihan kombinasi ini didasarkan pada keseimbangan antara tingkat akurasi dan stabilitas kinerja. Oleh karena itu, tahapan preprocessing tersebut digunakan pada proses selanjutnya karena terbukti memberikan kinerja yang paling optimal dan stabil pada evaluasi *preprocessing*.

Tabel 10. Hasil Pembagian Data Latih dan Data Validasi

Jenis Data	Jumlah Sampel
Data Latih	1.600
Data Validasi	400
Total	2.000

3.4. Hasil Pembagian Data (Data *Split*)

Pembagian *dataset* dilakukan dengan rasio 80:20 antara data pelatihan (*training*) dan data validasi (*validation*) untuk memastikan model memperoleh data yang cukup dalam proses pembelajaran sekaligus dapat dievaluasi secara objektif. Pembagian data dilakukan secara acak menggunakan *random state* tertentu untuk menjaga distribusi kata serta proporsionalitas kelas positif dan negatif pada kedua subset data. Pada penelitian ini, teknik SMOTE diterapkan hanya pada data pelatihan setelah proses pembagian data, sementara data validasi dibiarkan dalam kondisi asli untuk mencegah terjadinya data *leakage*. Dengan pendekatan ini, hasil evaluasi model menjadi lebih representatif dan dapat diandalkan. Hasil lengkap pembagian data ditunjukkan pada Tabel 10.

3.5. Hasil TF-IDF

Proses transformasi TF-IDF berhasil dilakukan dengan baik pada seluruh data teks ulasan yang telah melalui tahap *preprocessing*. Fungsi utama TF-IDF adalah mengubah data teks menjadi bentuk numerik yang dapat diproses oleh model *machine learning*, karena metode klasifikasi memerlukan input berupa angka untuk dapat melakukan analisis sentimen. Transformasi ini menghasilkan 1500 fitur unik dengan *shape* data *training* (1600, 1500) dan data validasi (400, 1500). Hasil dari proses TF-IDF dapat dilihat pada Tabel 11.

Tabel 11. Sepuluh *Term* Teratas dengan *TF-IDF Score* Tertinggi

Term	*TF-IDF Score*
gopay	0.048686
aplikasi	0.039558
saldo	0.035504
masuk	0.031133
transaksi	0.027703
pakai	0.026512

P-ISSN : 2089-3353
E-ISSN : 2808-9162

bayar	0.026254
padahal	0.025593
sangat	0.024481
transfer	0.022207

3.6. Hasil Evaluasi SMOTE

Setelah menentukan tahapan *preprocessing*, berdasarkan jumlah distribusi pelabelan menggunakan *Rating-based* akan dilakukan perhitungan untuk mengetahui tingkat keparahan ketidakseimbangan kelas yang ada pada *dataset* (data ulasan) menggunakan rumus (3) *Imbalance Ratio* (%), berikut ini hasil perhitungannya :

$$Imbalance\ Ratio\ (\%) = \frac{N_{minority}}{N_{majority} + N_{minority}}$$

$$= \frac{638}{1362 + 638} = \frac{638}{2000} = 0.319 = 31{,}9\%$$

Berdasarkan Tabel 2, ketidakseimbangan dataset termasuk kategori 'Ringan', sehingga diterapkan teknik SMOTE (*Synthetic Minority Oversampling Technique*) dengan *sampling strategy* 0.8 untuk menyeimbangkan distribusi kelas. Distribusi sebelum SMOTE menunjukkan 1090 sampel negatif dan 510 sampel positif pada data *training*. Setelah SMOTE, distribusi menjadi 1090 sampel negatif dan 872 sampel positif, seperti terlihat pada Gambar 2. SMOTE membangkitkan 362 sampel sintetis baru untuk kelas positif dengan menginterpolasi fitur-fitur dari sampel minoritas *existing*, menghasilkan karakteristik serupa namun tidak identik.

```
================================================
SMOTE ON TF-IDF - OPTIMIZED
================================================

=== Input Data Check ===
Shape X_train (TF-IDF): (1600, 1500)
Original class distribution: Counter({'Negatif': 1090, 'Positif': 510})

=== Applying SMOTE to TF-IDF Features ===

=== SMOTE Results ===
Before SMOTE - X_train shape: (1600, 1500)
After SMOTE - X_resampled shape: (1962, 1500)
Class distribution after SMOTE: Counter({'Negatif': 1090, 'Positif': 872})
```

Gambar 2. Hasil Distribusi Kelas Sebelum dan Sesudah SMOTE

Untuk mengevaluasi efektivitas teknik SMOTE, dilakukan perbandingan kinerjaantara model dengan dan tanpa SMOTE. Meskipun SMOTE dapat menyeimbangkan distribusi kelas, penerapannya tidak selalu meningkatkan kinerjadan dapat menurunkan kinerja akibat *overfitting* atau *noise* dari data sintetis. Ketiga metode klasifikasi dilatih pada kedua kondisi dataset dengan konfigurasi yang sama untuk memastikan perbandingan objektif. Evaluasi kinerjarinci dari kelas positif dan negatif untuk memberikan pemahaman mendalam tentang kemampuan klasifikasi setiap kelas. Hasil perbandingan ini menentukan konfigurasi dataset optimal untuk tahap hyperparameter tuning selanjutnya yang dapat dilihat pada Tabel 12 dan 13.

Tabel 12. Evaluasi Model Tanpa SMOTE per Kelas

Model	Class	Precision	Recall	F1-Score	Support
SVM	Negatif	90%	97%	93%	272
	Positif	92%	78%	84%	128
Naïve Bayes	Negatif	90%	99%	94%	272
	Positif	97%	77%	86%	128
Logistic Regression	Negatif	89%	99%	93%	272
	Positif	96%	73%	83%	128

Tabel 13. Evaluasi Model Tanpa SMOTE per Kelas

Model	Class	Precision	Recall	F1-Score	Support
SVM	Negatif	91%	96%	93%	272
	Positif	89%	80%	84%	128
Naïve Bayes	Negatif	92%	96%	94%	272
	Positif	91%	81%	86%	128
Logistic Regression	Negatif	91%	97%	94%	272
	Positif	92%	80%	85%	128

Meskipun *dataset* memiliki ketidakseimbangan 'ringan', evaluasi menyeluruh diperlukan untuk mencegah *overfitting* dan *bias*. Hasil menunjukkan SMOTE tidak berpengaruh signifikan pada akurasi keseluruhan, namun mengubah metrik lainnya. SMOTE menurunkan *F1-score* dan *precision* 1% pada SVM dan *Naïve Bayes*, tetapi meningkatkan *F1-score Logistic Regression* 1%. Model tanpa SMOTE menunjukkan bias kelas mayoritas dengan recall negatif sangat tinggi (99%) namun *recall* positif rendah (77-73%). Setelah SMOTE, *recall* menjadi seimbang: negatif turun menjadi 96-97% dan positif meningkat menjadi 80-81%, menghasilkan model lebih stabil.

3.7. Hasil *Evaluasi Tuning Hyperparameter*

Setelah menentukan penggunaan SMOTE pada data ulasan, dilakukan *hyperparameter* tuning untuk mencari kombinasi parameter terbaik yang dapat memaksimalkan kinerjamodel dalam melakukan klasifikasi. Proses tuning dilakukan menggunakan *RandomizedSearchCV* dengan berbagai parameter yang telah ditentukan untuk setiap metode. Tabel. 14, 15, 16, 17, dan 18 berikut menampilkan 10 kombinasi parameter terbaik untuk masing-masing model.

Tabel 14. Top 10 Hyperparameter SVM

Urut	C	Kernel	*Class Weight*	*Accuracy*	*F1-Score*	*Recall*	*Precision*
1	0.376	sigmoid	balanced	93%	92%	93%	92%
2	0.280	sigmoid	None	92%	92%	92%	92%
3	0.236	sigmoid	None	92%	92%	92%	92%
4	0.263	sigmoid	None	92%	92%	92%	92%
5	0.265	linear	None	92%	92%	92%	92%
6	0.258	linear	None	92%	92%	92%	92%
7	0.124	linear	balanced	92%	92%	92%	92%
8	0.136	linear	None	92%	92%	92%	92%
9	0.367	linear	None	92%	91%	92%	92%
10	0.367	linear	None	92%	91%	92%	92%

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Tabel 15. Top 10 *Hyperparameter* MultinomialNB

Urut	*Alpha*	*Fit_Prior*	*Force_Alpha*	*Accuracy*	*F1-Score*	*Recall*	*Prec.*
1	8.388	*True*	*True*	92%	92%	92%	92%
2	1.840	*True*	*True*	92%	91%	92%	91%
3	0.004	*True*	*False*	92%	91%	92%	91%
4	0.003	*True*	*False*	92%	91%	92%	91%
5	1.523	*True*	*False*	91%	91%	91%	91%
6	0.482	*True*	*True*	91%	91%	91%	91%
7	0.001	*True*	*False*	91%	91%	91%	91%
8	0.050	*True*	*True*	91%	91%	91%	91%
9	3.675	*True*	*True*	91%	91%	91%	91%
10	0.249	*True*	*True*	91%	91%	91%	91%

Tabel 16. *Top Hyperparameter GaussianNB*

Urutan	***Var_Smoothing***	***Priors***	***Accuracy***	***F1-Score***	***Recall***
1	1.17E-07	None	80%	80%	80%
2	7.41E-08	None	80%	80%	80%
3	2.23E-07	None	80%	80%	80%
4	9.02E-08	None	80%	80%	80%
5	7.89E-08	None	80%	80%	80%
6	7.44E-07	None	80%	80%	80%
7	1.06E-08	None	80%	80%	80%
8	1.82E-09	None	80%	80%	80%
9	3.52E-10	None	80%	80%	80%
10	4.68E-11	None	80%	80%	80%

Tabel 17. *Top* 10 *Hyperparameter BernoulliNB*

Urut	*Alpha*	*Fit_Prior*	*Binarize*	*Accuracy*	*F1-Score*	*Recall*	*Precision*
1	0.002634	*False*	0	91%	91%	91%	91%
2	0.003522	*False*	0	91%	91%	91%	91%
3	0.007565	*True*	None	91%	91%	91%	91%
4	0.049897	*True*	0	91%	90%	91%	90%
5	0.005031	*False*	None	90%	90%	90%	90%
6	8.388483	*True*	0	90%	90%	90%	90%
7	0.053311	*False*	0	90%	90%	90%	90%
8	3.675258	*True*	0	90%	90%	90%	90%
9	0.248527	*True*	0	90%	90%	90%	90%
10	1.840344	*True*	0	90%	90%	90%	90%

Tabel 18. *Top* 10 *Hyperparameter Logistic Regression*

U	C	*Penalty*	*Solver*	*Class_Weight*	*Max_Iter (rb)*	*Intercept*	*Accuracy (%)*	*F1-Score %*	*Recall %*	*Precision %*
1	1.75	l2	saga	None	7	False	92	92	92	92
2	1.82	l2	liblinear	None	3	False	92	92	92	92
3	1.78	l2	saga	None	7	False	92	92	92	92
4	1.86	l2	liblinear	None	3	False	92	92	92	92
5	1.75	l2	liblinear	None	5	False	92	92	92	92
6	1.77	l2	saga	None	7	False	92	92	92	92
7	1.76	l2	saga	None	10	False	92	92	92	92
8	1.83	l2	liblinear	None	1.5	False	92	92	92	92
9	1.82	l2	saga	None	2	False	92	92	92	92
10	1.84	l2	liblinear	None	2.5	False	92	92	92	92

Berdasarkan hasil *hyperparameter tuning* menunjukkan variasi kinerjayang berbeda untuk setiap metode. Pada model SVM didapatkan parameter yang memberikan kinerjaoptimal pada *settingan* C = 0.376007, *kernel sigmoid, class weight balanced* dengan *accuracy* 93%, *f1-score* 92%, *recall* 93% dan *precision* 92%. Pada model *Naïve Bayes* didapatkan parameter yang memberikan kinerja optimal pada varian *MultinomialNB* dengan *accuracy*, *f1-score, recall,* dan *precision* 92%. Dan terakhir pada model *Logistic Regression* didapatkan parameter yang memberikan kinerja optimal pada setinggan C = 1.75, *penalty* = l2, *solver* = saga, *class_weight = None*, *max_iter* = 7000, dan *fit_intercept = False* dengan *accuracy*, *f1-score*, *recall*, dan *precision* 92%. Hasil tuningan optimal ini akan dibandingkan dengan parameter default dengan menggunakan SMOTE untuk melihat perbedaannya yang dapat dilihat pada Tabel 19 dan 20.

Tabel 19. Hasil Kinerja Model Tanpa *Tuning Hyperparameter*

Model	*Accuracy*	*F1-Score*	*Recall*	*Precision*
SVM	91%	90%	91%	90%
Naïve Bayes	92%	91%	92%	91%
Logistic Regression	91%	91%	91%	91%

Tabel 20. Hasil KinerjaModel Dengan Tuning Hyperparameter

Model	*Accuracy*	*F1-Score*	*Recall*	*Precision*
SVM	93%	92%	93%	92%
Naïve Bayes	92%	92%	92%	92%
Logistic Regression	92%	92%	92%	92%

Tuning hyperparameter memberikan dampak positif konsisten terhadap kinerjasemua model klasifikasi. Model SVM menunjukkan peningkatan paling signifikan dengan akurasi naik dari 91% menjadi 93%, diikuti perbaikan seluruh metrik: *F1-Score* (90% ke 92%), *recall* (91% ke 93%), dan *precision* (90% ke 92%). Model *Naive Bayes* mempertahankan akurasi 92% namun mengalami peningkatan *F1-Score* dari 91% menjadi 92% serta perbaikan *recall* dan precision menjadi 92%. Model *Logistic Regression* menunjukkan konsistensi peningkatan dengan semua metrik naik dari 91% menjadi 92%.

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Hasil membuktikan proses *tuning* berhasil menemukan kombinasi parameter optimal tanpa mengorbankan kinerjaaspek tertentu. Tidak ada model yang mengalami penurunan, bahkan sebaliknya semua mengalami peningkatan atau mempertahankan kualitas baik. Konsistensi peningkatan menunjukkan model hasil tuning tidak hanya lebih akurat tetapi juga lebih seimbang dalam klasifikasi, sehingga lebih dapat diandalkan dalam kondisi nyata. Keputusan menggunakan tuning *hyperparameter* dianggap tepat karena dapat meningkatkan atau mempertahankan kinerjamodel yang sudah baik.

3.8. Hasil Penentuan Model Terbaik

Setelah melakukan pengujian menyeluruh terhadap empat metrik evaluasi meliputi metode pelabelan, urutan dan kelengkapan *preprocessing*, penggunaan SMOTE, serta penerapan *hyperparameter* tuning, penelitian ini berhasil mengidentifikasi konfigurasi optimal yang memberikan kinerja terbaik untuk ketiga metode klasifikasi. Konfigurasi optimal tersebut terdiri dari penggunaan pelabelan *rating-based, preprocessing* 6 tahap, yaitu *cleaning-casefolding-tokenizing-normalization-stopword-stemming*, penerapan SMOTE untuk mengatasi ketidakseimbangan data, dan implementasi *hyperparameter* tuning untuk mengoptimalkan parameter model.

Hasil evaluasi menunjukkan kinerja sangat memuaskan Dimana, pada Tabel 21, SVM mencapai akurasi tertinggi 93%, diikuti *Naive Bayes* dan *Logistic Regression* masing-masing 92%. Pada Tabel 21, SVM menunjukkan kinerja seimbang dengan *precision* 0.93 (negatif) dan 0.92 (positif), recall 0.96 (negatif) dan 0.84 (positif). *Naive Bayes*, pada Tabel 22, memiliki precision tertinggi kelas positif (0.95) namun *recall* terendah (0.79), menunjukkan kecenderungan konservatif. *Logistic Regression*, pada Tabel 23, menampilkan kinerja paling konsisten dengan precision 0.92 untuk kedua kelas. SVM terpilih sebagai model terbaik karena kemampuan generalisasi superior melalui pendekatan margin maksimum dan kernel untuk menangani hubungan *non-linear*. Meskipun *Naïve Bayes* mencapai akurasi yang tinggi (92%), model ini menunjukkan bias terhadap kelas tertentu, yang tercermin dari perbedaan nilai *recall* antar kelas. Berdasarkan Tabel 22, *Naïve Bayes* memiliki *recall* yang sangat tinggi pada kelas negatif (0,98), namun *recall* pada kelas positif relatif lebih rendah (0,79). Kondisi ini menunjukkan bahwa model cenderung gagal mengenali sebagian ulasan positif, terutama dibandingkan dengan SVM yang memiliki distribusi *recall* yang lebih seimbang antar kelas.

Tabel 21. *Classification Report* Model SVM

	Precision	*Recall*	*F1-Score*	*Support*
Negatif	93%	96%	95%	272
Positif	92%	84%	88%	128
Accuracy			93%	400
Macro avg	92%	90%	91%	400
Weighted avg	92%	93%	92%	400

Tabel 22. *Classification Report* Model *Naïve Bayes*

	Precision	*Recall*	*F1-Score*	*Support*
Negatif	91%	98%	94%	272
Positif	95%	79%	86%	128
Accuracy			92%	400
Macro avg	93%	89%	90%	400
Weighted avg	92%	92%	92%	400

Tabel 23. *Classification Report* Model *Logistic Regression*

	Precision	*Recall*	*F1-Score*	*Support*
Negatif	92%	97%	94%	272
Positif	92%	83%	87%	128
Accuracy			92%	400
Macro avg	92%	90%	91%	400
Weighted avg	92%	92%	92%	400

3.9. Hasil Pengujian Konsistensi Model dengan *Dataset* Validasi

Setelah menentukan metode terbaik dari *dataset* GoPay, ketiga metode klasifikasi akan diuji kembali menggunakan *dataset* validasi (dataset LinkAja) dengan konfigurasi optimal yang sama untuk memverifikasi konsistensi performa. Pengujian ini bertujuan untuk mengevaluasi apakah urutan kinerjaketiga metode tetap stabil pada dataset yang berbeda. Berdasarkan hasil pengujian ketiga metode yakni SVM, *Naïve Bayes*, dan *Logistic Regression* terhadap *dataset* LinkAja didapat hasil, yang dapat dilihat pada Tabel 24.

Tabel 24. Kinerja Model Terhadap *Dataset* Validasi

Model	*Accuracy*	*F1-Score*	*Recall*	*Precision*
SVM	95%	95%	95%	96%
Naïve Bayes	95%	95%	95%	95%
Logistic Regression	94%	95%	94%	95%

Hasil menunjukkan SVM dan *Naïve Bayes* memiliki kinerjaidentik dengan akurasi, *F1-Score*, dan *recall* masing-masing 95%, namun SVM sedikit unggul pada *precision* (96% vs 95%). Kedua metode mengungguli *Logistic Regression* yang memperoleh akurasi dan *recall* 94% serta *precision* 95%, meskipun *F1-Score* konsisten 95%. Urutan kinerjatetap konsisten dengan dataset GoPay dimana SVM menjadi metode terbaik berdasarkan kinerjakomprehensif, menunjukkan SVM memiliki konsistensi baik pada *dataset* berbeda.

4. Kesimpulan

Berdasarkan analisis sentimen ulasan aplikasi GoPay di *Google Play Store* menggunakan tiga metode *machine learning* dengan menerapkan berbagai teknik *preprocessing*, SMOTE untuk mengatasi ketidakseimbangan kelas, dan optimalisasi *hyperparameter*, hasil eksperimen menunjukkan SVM memberikan kinerjaterbaik dengan akurasi 93%,

precision 92%, *recall* 93%, dan *F1-Score* 92% karena kemampuannya menemukan *hyperplane* optimal dengan margin maksimal dan generalisasi yang baik, terutama pada kelas negatif (*F1-Score* 95%) dan positif (*F1-Score* 88%). Meskipun *Logistic Regression* dan *Naïve Bayes* menunjukkan akurasi identik 92%, *Logistic Regression* menempati posisi kedua dengan konsistensi lebih baik (*recall* kelas positif 83% vs 79% *Naïve Bayes*), sementara *Naïve Bayes* berada di posisi ketiga karena bias lebih besar terhadap kelas mayoritas. Validasi menggunakan *dataset* LinkAja mengonfirmasi konsistensi kinerjaSVM dengan akurasi 95% dan *precision* 96%, membuktikan kemampuan generalisasi yang baik pada berbagai jenis aplikasi dan menjawab permasalahan inkonsistensi hasil penelitian sebelumnya yang disebabkan perbedaan konfigurasi metode pelabelan, *preprocessing*, dan tuning *hyperparameter*. Hasil komprehensif ini berhasil mengidentifikasi SVM sebagai metode superior yang konsisten dan dapat diandalkan untuk implementasi analisis sentimen ulasan aplikasi di *Google Play Store*. Selain itu, keunggulan SVM juga tercermin dari nilai *F1-Score* yang tinggi dan relatif seimbang pada kelas sentimen positif dan negatif, yang menunjukkan bahwa model ini mampu mengenali kedua kelas secara proporsional dan tidak bias terhadap kelas mayoritas.

Untuk pengembangan masa depan, penelitian menyarankan eksplorasi deep learning (LSTM, BERT, transformer), penggunaan *dataset* lebih besar dan beragam, sistem multi-kelas yang lebih detail, teknik feature engineering canggih seperti *word embeddings* dan *contextual embeddings*, serta evaluasi sistematis kombinasi preprocessing untuk menemukan konfigurasi optimal yang akurat dan konsisten pada berbagai dataset dan metode.

Daftar Rujukan

[1] A. Rakhmanita and D. T. Anggarini, "DAMPAK TRANSAKSI PEMBAYARAN GO-PAY BAGI PENINGKATAN PENJUALAN PEDAGANG KECIL MENENGAH DI PASAR MODERN BSD," *Widya Cipta: Jurnal Sekretari dan Manajemen*, vol. 4, no. 2, 2020, doi: 10.31294/widyacipta.v4i2.8416.

[2] R. M. Turjaman and I. Budi, "Analisis Sentimen Berbasis Aspek Marketing Mix Terhadap Ulasan Aplikasi Dompet Digital (Studi Kasus: Aplikasi Linkaja Pada Twitter)," *Jurnal Darma Agung*, vol. 30, no. 2, p. 266, 2022, doi: 10.46930/ojsuda.v30i2.1672.

[3] Y. yuli Astari, A. Afiyati, and S. W. Rozaqi, "Analisis Sentimen Multi-Class Pada Sosial Media Menggunakan Metode Long Short-Term Memory (LSTM)," *Jurnal Linguistik Komputasional*, vol. 4, no. 1, pp. 8–12, 2021, [Online]. Available: http://inacl.id/journal/index.php/jlk/article/view/43

[4] A. Hendra and F. Fitriyani, "Analisis Sentimen Review Halodoc Menggunakan Naive Bayes Classifier," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 6, no. 2, pp. 78–89, 2021, doi: 10.14421/jiska.2021.6.2.78-89.

[5] K. Perdana, "Komparasi Metode Naive Bayes, Support Vector Machine, dan Logistic Regression pada Analisis Sentimen Pengguna Aplikasi Transportasi Online," *Kumpulan jurnal Ilmu Komputer (KLIK)*, vol. 10, no. 01, pp. 27–38, 2023, doi: 10.20527/klik.v10i1.616.

[6] K. A. Baihaqi, "A Comparison Support Vector Machine, Logistic Regression And Naive Bayes For Classification Sentimen Analisys user Mobile App," *International Journal of Artificial Intelligence Research*, vol. 7, no. 1, p. 64, 2023, doi: 10.29099/ijair.v7i1.962.

[7] H. Jauhary, "Perbandingan Metode Analisis Sentimen Support Vector Machine, Naive Bayes, dan Logistic Regression (Studi Kasus: Ulasan Google Playstore Aplikasi Linkedin)," *AT-TAWASSUTH: Jurnal Ekonomi Islam*, 2023, [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/76808

[8] O. S. D. Fadhillah, J. H. Jaman, and Carudin, "Perbandingan Naive Bayes, Support Vector Machine, Logistic Regression dan Random Forest dalam Menganalisis Sentimen Mengenai Tiktokshop," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 6, no. 1, pp. 840–847, 2021, doi: 10.23960/jitet.v13i1.5746.

[9] I. R. Ainunnisa and S. Sulastri, "Analisis Sentimen Aplikasi Tiktok dengan Metode Support Vector Machine (SVM), Logistic Regression dan Naive Bayes," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 6, no. 3, pp. 423–430, 2023, doi: 10.32493/jtsi.v6i3.31076.

[10] D. Surya Sayogo, B. Irawan, and A. Bahtiar, "Analisis Sentimen Ulasan Aplikasi DANA di Google Play Store Menggunakan Metode Naive Bayes," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 6, pp. 3314–3319, 2024, doi: 10.36040/jati.v7i6.8178.

[11] A. Fathin, "Analisis Sentimen Terhadap Ulasan Aplikasi Mobile Menggunakan Metode Support Vector Machine (Svm) Dan Pendekatan Lexicon Based," 2022. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/65009

[12] R. A. Afif, "Analisis Sentimen Aplikasi Adiraku di Google Play Store Menggunakan Metode Support Vectore Machine," *Jurnal Fasilkom*, vol. 15, no. 1, pp. 163–171, 2025, doi: 10.37859/jf.v15i1.8510.

[13] 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. 2, pp. 81–88, 2023.

[14] A. Alwasi'a, "Analisis Sentimen Pada Review Aplikasi Berita Online Menggunakan Metode Maximum Entropy (Studi Kasus: Review Detikcom pada Google Play 2019)," 2020. [Online]. Available: http://dspace.uii.ac.id/123456789/24006

[15] S. Shevira, I. M. A. D. Suarjaya, and P. W. Buana, "Pengaruh Kombinasi dan Urutan Pre-Processing pada Tweets Bahasa Indonesia," *JITTER: Jurnal Ilmiah Teknologi dan Komputer*, vol. 3, no. 2, p. 1074, 2022, doi: 10.24843/jtrti.2022.v03.i02.p06.

[16] T. A. Pamungkas and A. Salam, "Optimalisasi Model SciBERT dengan Attention-BiLSTM-CRF untuk Pengenalan Entitas Penyakit dalam Teks Biomedis," *Building of Informatics, Technology and Science (BITS)*, vol. 7, no. 1, pp. 147–156, 2025, doi: 10.47065/bits.v7i1.7263.

[17] K. Akbar and M. Hayaty, "Data Balancing untuk Mengatasi Imbalance Dataset pada Prediksi Produksi Padi," *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, vol. 2, no. 02, pp. 1–14, 2020, doi: 10.46772/intech.v2i02.283.

[18] K. Fujiwara, "Over- and Under-sampling Approach for Extremely Imbalanced and Small Minority Data Problem in Health Record Analysis," *Front Public Health*, vol. 8, pp. 1–15, 2020, doi: 10.3389/fpubh.2020.00178.

[19] M. Aminullah, "Perbandingan Kinerja Klasifikasi Machine Learning dengan Teknik Resampling pada Dataset Tidak Seimbang," 2021. [Online]. Available: https://repository.uinjkt.ac.id/dspace/bitstream/123456789/57648/1/MUHAMMAD AMINULLAH-FST.pdf

[20] C. B. Handoko and C. S. K. Aditya, "Penerapan Teknik SMOTE dalam Mengatasi Imbalance Data Penyakit Diabetes Menggunakan Metode ANN," *Smart Comp:*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Jurnalnya Orang Pintar Komputer, vol. 14, no. 105, pp. 13–20, 2025.

[21] A. Demircioglu, "Applying oversampling before cross-validation will lead to high bias in radiomics," *Sci Rep*, vol. 14, no. 1, pp. 1–11, 2024, doi: 10.1038/s41598-024-62585-z.

[22] O. H. Rahman, G. Abdillah, and A. Komarudin, "Klasifikasi Ujaran Kebencian pada Media Sosial Twitter Menggunakan Support Vector Machine," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 17–23, 2021, doi: 10.29207/resti.v5i1.2700.

[23] H. Hendiana, A. Purnamasari, and I. Ali, "Analisis Sentimen Komentar Berita Detik.com Menggunakan Metode Suport Vektor Machine (SVM)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, pp. 3175–3181, 2024, doi: 10.36040/jati.v8i3.8421.

[24] Y. Yuliyana and A. S. R. M. Sinaga, "Sistem Pakar Diagnosa Penyakit Gigi Menggunakan Metode Naive Bayes," *Fountain of Informatics Journal*, vol. 4, no. 1, p. 19, 2019, doi: 10.21111/fij.v4i1.3019.

[25] G. P. Kawani, "Implementasi Metode Klasifikasi Naive Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga," *Journal of Informatics, Information System, Software Engineering and Applications (INISTA)*, vol. 1, no. 2, pp. 73–81, 2019, doi: 10.20895/inista.v1i2.73.

[26] T. Taslim, S. Handayani, and F. Fajrizal, "Kinerja Komparatif Optimasi Metode Naive Bayes dalam Klasifikasi Teks untuk Uji Klinis Kanker," *Jurnal Eksplora Informatika*, vol. 13, no. 1, pp. 113–123, 2023, doi: 10.30864/eksplora.v13i1.994.

[27] K. Kelvin, "Analisis Perbandingan Sentimen Corona Virus Disease-2019 (Covid19) pada Twitter Menggunakan Metode Logistic Regression Dan Support Vector Machine (SVM)," *Jurnal Sistem Informasi dan Ilmu Komputer Prima (JUSIKOM PRIMA)*, vol. 5, no. 2, pp. 47–52, 2022, doi: 10.34012/jurnalsisteminformasidanilmukomputerprima.v5i2.2365.

[28] N. K. C. Pratiwi, N. Ibrahim, and S. Saidah, "Prediksi Kanker Paru menggunakan Grid search untuk Optimasi Hyperparameter pada Metode MLP dan Logistic Regression," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 12, no. 3, pp. 556–568, 2024, doi: 10.26760/elkomika.v12i3.556.