

ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI LINE BANK DI GOOGLE PLAY STORE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE BERBASIS SMOTE DENGAN OPTIMASI GRIDSEARCHCV

Amelia Dian Melinda, Virgania Sari
Matematika, Universitas Negeri Semarang
Semarang, Indonesia
ameliadian3012@students.unnes.ac.id

ABSTRAK

Pesatnya perkembangan teknologi digital telah membawa perubahan signifikan terutama pada sektor perbankan. Perkembangan ini mendorong munculnya berbagai aplikasi perbankan digital, salah satunya aplikasi Line Bank. Ulasan pengguna yang tersedia pada platform digital menjadi sumber data penting untuk menganalisis persepsi, pengalaman, dan sentimen pengguna terhadap kualitas layanan aplikasi, sehingga diperlukan analisis sentimen untuk memahami persepsi pengguna secara sistematis. Penelitian ini bertujuan untuk mengoptimasi kinerja *Support Vector Machine* berbasis SMOTE menggunakan *GridSearchCV* dalam analisis sentimen ulasan aplikasi pengguna aplikasi Line Bank. Data yang digunakan adalah data ulasan pengguna aplikasi Line Bank dari platform Google Play Store yang diambil menggunakan teknik *web scraping*, yaitu sebanyak 4000 data. Data ulasan diklasifikasikan ke dalam tiga kelas sentimen, yaitu positif, negatif, dan netral. Distribusi data pada ketiga kelas sentimen menunjukkan kondisi yang tidak seimbang, sehingga diperlukan penerapan SMOTE untuk menyeimbangkan data sebelum proses klasifikasi dilakukan. Pada model SVM dengan kernel linear sebagai kernel terbaik, sebelum dilakukan optimasi *GridSearchCV* diperoleh akurasi dan *recall* yang sama-sama sebesar 85,57%, *precision* 85,91%, serta *F1-score* 85,66%. Sedangkan, setelah diterapkan optimasi menggunakan *GridSearchCV* dengan kombinasi parameter terbaik (kernel = linear, C = 10, dan gamma = scale) kinerja model mengalami peningkatan yang signifikan dengan akurasi dan *recall* yang sama-sama menjadi sebesar 88,61%, *precision* 88,50%, serta *F1-score* 88,55%. Hal ini menunjukkan bahwa optimasi *hyperparameter tuning* menggunakan *GridSearchCV* terbukti efektif mampu meningkatkan kinerja dari model SVM secara konsisten pada semua metrik evaluasi.

Kata kunci : Analisis Sentimen, Line Bank, Support Vector Machine, SMOTE, GridSearchCV

1. PENDAHULUAN

Di era digital saat ini, perkembangan teknologi yang semakin pesat telah membawa perubahan signifikan dalam berbagai sektor, terutama sektor perbankan. Perkembangan ini mendorong munculnya berbagai aplikasi perbankan digital yang menawarkan kemudahan layanan kepada masyarakat salah satunya aplikasi Line Bank. Line Bank by Hana Bank merupakan aplikasi perbankan digital yang memberikan kemudahan dalam aktivitas finansial melalui sejumlah fitur-fitur yang ditawarkan [1].

Salah satu aspek penting dalam keberhasilan sebuah aplikasi perbankan digital adalah kepuasan pengguna yang sering kali tercermin melalui ulasan dan penilaian yang diberikan di platform digital seperti Google Play Store. Ulasan pengguna tidak hanya sebagai sarana umpan balik bagi penyedia layanan, tetapi juga menjadi sumber informasi berharga untuk memahami persepsi, pengalaman, serta sentimen pengguna terhadap kualitas layanan yang diberikan, sehingga analisis sentimen menjadi pendekatan yang relevan untuk memahami persepsi pengguna secara sistematis. Analisis sentimen merupakan bidang kajian dalam *text mining* yang berfokus pada proses pengolahan teks untuk mengidentifikasi opini, emosi, serta kecenderungan pandangan atau penilaian

individu pada suatu produk, layanan digital, atau peristiwa tertentu [2].

Dalam penerapan analisis sentimen, data ulasan pengguna sering kali mengalami masalah ketidakseimbangan, di mana jumlah sentimen negatif jauh lebih dominan dibandingkan dengan sentimen positif dan netral, ataupun sebaliknya. Masalah ketidakseimbangan ini bisa menyebabkan model klasifikasi *machine learning* berpotensi menjadi bias dan kurang akurat dalam mengenali kelas minoritas. Oleh karena itu, diperlukan teknik untuk mengatasi permasalahan ketidakseimbangan kelas dalam data tersebut. Pendekatan yang dapat diterapkan untuk mengatasi masalah ketidakseimbangan kelas dalam data adalah SMOTE. *Synthetic Minority Over-Sampling Technique* (SMOTE) merupakan salah satu pendekatan *oversampling* yang dipakai pada *machine learning* untuk mengatasi *class imbalance* dengan menghasilkan sampel buatan pada *minority class*, sehingga jumlah data pada *minority class* bertambah dan distribusinya menjadi lebih seimbang [3].

Dalam proses klasifikasi, salah satu metode yang populer digunakan yaitu Support Vector Machine. *Support Vector Machine* (SVM) merupakan salah satu metode *supervised learning* untuk klasifikasi yang terkenal dengan kemampuannya dapat menghasilkan akurasi yang tinggi dan efektif dalam memisahkan

data dalam kelas berbeda melalui pencarian *hyperplane* optimal atau bidang pembatas terbaik [4].

Pada penelitian yang dilaksanakan oleh yang membahas mengenai komparasi antara *Support Vector Machine* dengan Naive Bayes, diperoleh hasil bahwa akurasi pada SVM lebih tinggi dibandingkan dengan Naive Bayes.

Penelitian lain juga dilakukan oleh [5] dengan mengkomparasikan KNN, SVM, dan Random Forest. Diperoleh akurasi SVM lebih tinggi dibandingkan dengan KNN serta Random Forest. Hal ini menunjukkan bahwa penggunaan metode SVM terbukti memiliki performa paling unggul dalam mengklasifikasikan sentimen ulasan pengguna.

Meskipun demikian, kinerja pada sebuah metode juga sangat dipengaruhi oleh pemilihan *hyperparameter* yang tepat. Pemilihan *hyperparameter* yang tidak tepat maka akan menurunkan kinerja, baik dari akurasi ataupun konsistensi model secara keseluruhan. Oleh karena itu, dibutuhkan optimasi *hyperparameter tuning* untuk mencari kombinasi parameter terbaik (*best parameters*) sehingga kinerja model dapat menjadi lebih optimal. Beberapa metode *hyperparameter tuning* yang umum digunakan antara lain *GridSearchCV*, *RandomSearchCV*, dan Bayesian Optimization [6].

GridSearchCV merupakan metode *hyperparameter tuning* yang bekerja dengan cara mencoba berbagai kombinasi nilai *hyperparameter* secara sistematis yang telah ditentukan sebelumnya, kemudian menerapkan kombinasi tersebut pada model untuk dievaluasi dengan *cross-validation*, hingga akhirnya dipilih kombinasi parameter dengan kinerja terbaik sebagai *hyperparameter* optimal pada model [7].

Beberapa penelitian berikut menerapkan *hyperparameter tuning* untuk mencari kombinasi parameter terbaik. Seperti penelitian yang dilaksanakan oleh [8] yang membahas tentang komparasi antara optimasi pada Random Forest dengan *GridSearchCV* dan Bayesian Hyperparameter Optimization, diperoleh bahwa optimasi dengan *GridSearchCV* didapatkan akurasi sebesar 74%, tetapi memerlukan waktu lebih lama. Sedangkan, optimasi dengan Bayesian diperoleh akurasi sebesar 73%, tetapi memerlukan waktu yang lebih singkat. Hasil ini menunjukkan bahwa, *GridSearchCV* unggul dalam akurasi meskipun memerlukan waktu yang lebih lama.

Penelitian lain juga dilakukan oleh [9] mengenai metode *Support Vector Machine* yang dioptimasi menggunakan *hyperparameter tuning GridSearchCV*. Dari penelitian ini, diperoleh hasil akurasi sebelum dilakukan optimasi dengan *GridSearchCV* sebesar 73,31%. Sedangkan setelah dioptimasi dengan *GridSearchCV* akurasi mengalami peningkatan cukup tinggi sebesar 21,5%, sehingga menjadi 94,8%. Hal ini menunjukkan bahwa, penerapan *hyperparameter tuning* menggunakan *GridSearchCV* terbukti efektif dapat meningkatkan performa model klasifikasi.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengoptimasi kinerja *Support Vector Machine* berbasis SMOTE dengan menggunakan *GridSearchCV* dalam analisis sentimen ulasan aplikasi Line Bank. Diharapkan pendekatan ini mampu meningkatkan kinerja model klasifikasi sentimen secara lebih optimal dan memberikan gambaran yang lebih akurat mengenai persepsi pengguna terhadap layanan aplikasi perbankan digital. Hasil penelitian ini diharapkan dapat menjadi referensi bagi pengembangan sistem analisis sentimen serta pengambilan keputusan strategis di bidang layanan perbankan digital.

2. TINJAUAN PUSTAKA

2.1. Analisis Sentimen

Analisis sentimen atau yang biasa disebut *opinion mining* merupakan pendekatan dalam bidang komputasi untuk mendeteksi, menggali, serta menganalisis perasaan, penilaian, opini, evaluasi, sikap, dan emosi terhadap suatu entitas seperti produk, layanan, peristiwa, isu dari data teks tidak terstruktur secara otomatis [2].

Teks tersebut diklasifikasikan menjadi sentimen positif, negatif, ataupun netral. Dalam analisis sentimen terdapat tiga tingkatan klasifikasi yaitu klasifikasi tingkat dokumen, klasifikasi tingkat kalimat, dan klasifikasi tingkat fitur atau aspek [10].

2.2. GridSearchCV

GridSearchCV merupakan metode *hyperparameter tuning* yang bekerja dengan melakukan pencarian terhadap kombinasi nilai *hyperparameter* yang telah ditentukan sebelumnya, kemudian mengevaluasi setiap kombinasi tersebut menggunakan *cross-validation* (validasi silang) untuk menilai kinerja dari model, selanjutnya akan dipilih kombinasi yang memberikan kinerja paling baik sebagai parameter terbaik untuk model [7].

Keunggulan utama dari *GridSearchCV* adalah pada kemampuannya dalam mengotomatisasi proses optimasi model, terutama ketika jumlah *hyperparameter* dan *search space* nya sangat besar atau luas. Melalui *stratified k-fold cross validation*, metode ini dapat mengevaluasi performa setiap kombinasi parameter pada berbagai subset (*fold*) data, sehingga mengurangi risiko *overfitting* serta meningkatkan kemampuan generalisasi model. Selain itu juga, penerapannya yang mudah melalui pustaka *scikit-learn*. Walaupun, memiliki waktu komputasi cukup lama ketika ruang pencariannya yang terlalu luas. *Hyperparameter tuning* menggunakan *GridSearchCV* berperan penting dalam mengoptimalkan performa model *machine learning*.

2.3. Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan salah satu metode *supervised learning* yang digunakan untuk klasifikasi dan regresi. Algoritma SVM pertama kali diperkenalkan oleh Vladimir Naumovich Vapnik

dan rekan-rekannya pada tahun 1992. *Support Vector Machine* (SVM) terkenal dengan kemampuannya dapat menghasilkan akurasi yang tinggi dengan memisahkan data dalam kelas berbeda melalui pencarian *hyperplane* optimal atau bidang pembatas terbaik dengan memaksimalkan *margin* (jarak) antar kelasnya [12]. Pada SVM linear, *hyperplane* ini didefinisikan melalui persamaan 2.6 berikut.

$$(w \cdot x_i) + b = 0 \quad (1)$$

Sehingga didapatkan dua buah kelas yaitu kelas positif (+1) dan kelas negatif (-1) yang dirumuskan pada pertidaksamaan 2.7 dan 2.8 berikut.

$$(w \cdot x_i + b) \geq +1, \quad \text{untuk } y_i = +1 \quad (2)$$

$$(w \cdot x_i + b) \leq -1, \quad \text{untuk } y_i = -1 \quad (3)$$

dengan w adalah vektor bobot, x_i adalah input vektor fitur, b adalah konstanta bias (*intercept*), dan y_i adalah label kelas.

Margin maksimum dapat diperoleh dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu $\frac{1}{\|w\|}$. Margin maksimum ditentukan dengan persamaan 2.9 berikut.

$$\min w = \frac{1}{2} \|w\|^2 \quad (4)$$

dengan kendala pertidaksamaan 2.10 berikut.

$$y_i(w \cdot x_i + b) - 1 \geq 0, \quad (i = 1, 2, 3, \dots, n) \quad (2.10)$$

Beberapa jenis fungsi kernel yang sering digunakan pada SVM antara lain sebagai berikut [13]:

a. Kernel Linear

$$K(x_i, x_j) = x_i \cdot x_j \quad (5)$$

b. Kernel Polynomial

$$K(x_i, x_j) = (x_i \cdot x_j + c)^d \quad (6)$$

c. Kernel Radial Basis Function (RBF)

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (7)$$

d. Kernel Sigmoid

$$K(x_i, x_j) = \tanh(\alpha x_i \cdot x_j + c) \quad (8)$$

2.4. SMOTE

Synthetic Minority Oversampling Technique (SMOTE) merupakan salah satu teknik *oversampling* yang dipakai pada *machine learning* untuk mengatasi ketidakseimbangan kelas (*class imbalance*) dalam dataset antara kelas positif, negatif, dan netral. Teknik SMOTE menghasilkan sampel buatan pada *minority class*, sehingga jumlah data pada kelas minoritas bertambah dan distribusinya menjadi lebih seimbang [3].

Tahapan dalam pembuatan data sintesis pada SMOTE dimulai dengan memilih sebuah sampel dari *minority class* secara acak, selanjutnya mencari beberapa k tetangga terdekatnya (*k-nearest neighbors*) dari *minority class* dalam ruang fitur, di mana jumlah k tetangga terdekat yang umum digunakan adalah lima, setelah tetangga terdekat teridentifikasi, selanjutnya dipilih salah satu tetangga terdekatnya secara acak, kemudian membuat data sintesis baru dengan menempatkannya di antara titik-titik tersebut yang menghubungkan sampel minoritas asli dengan tetangga yang terpilih [11].

Rumus pembuatan data sintesis pada teknik SMOTE dapat dilihat pada persamaan berikut.

$$x_{baru} = x + (x^* - x) \cdot s \quad (9)$$

Keterangan:

x_{baru} : Data sintesis baru

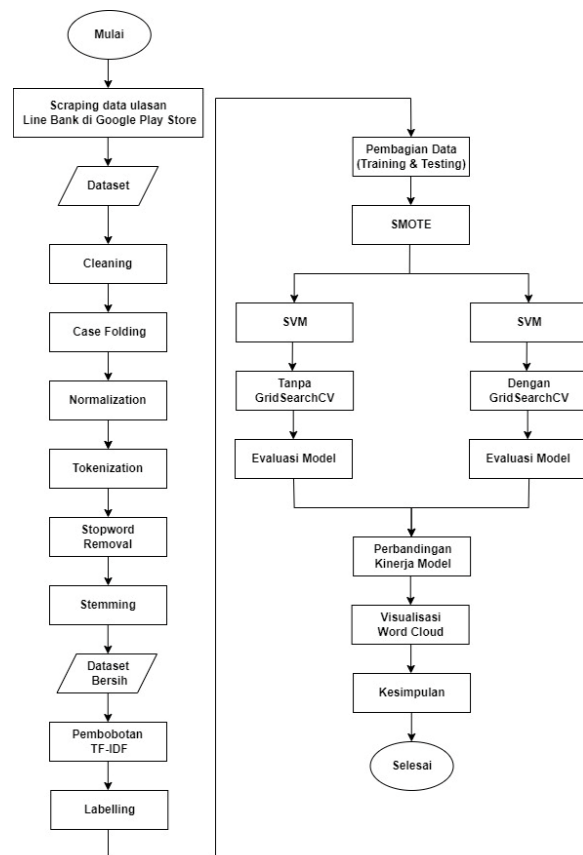
x : Data yang akan direplikasi

x^* : Data yang mempunyai jarak terdekat dari data x

s : Bilangan acak dengan rentang $0 \leq s \leq 1$

3. METODE PENELITIAN

Penelitian ini dilakukan melalui tahapan-tahapan yang tersusun secara sistematis dan terstruktur guna mencapai tujuan penelitian. Adapun diagram alur penelitian ini disajikan pada Gambar 1 sebagai berikut.



Gambar 1. Diagram Alur Penelitian

Berdasarkan Gambar 1 di atas, penelitian diawali dengan tahap *scraping* data ulasan aplikasi Line Bank dari Google Play Store yang kemudian dilanjutkan dengan tahapan *preprocessing*. Data yang telah bersih kemudian dilakukan pembobotan menggunakan TF-IDF serta pelabelan sentimen. Selanjutnya, data dibagi menjadi data training dan testing, kemudian diterapkan SMOTE untuk mengatasi ketidakseimbangan kelas dalam data. Proses klasifikasi dilakukan menggunakan metode SVM dengan dua pendekatan, yaitu SVM tanpa optimasi dan SVM dengan optimasi *hyperparameter tuning* menggunakan *GridSearchCV*. Hasil klasifikasi kemudian dievaluasi dan dibandingkan berdasarkan nilai akurasi, presisi, *recall*,

dan *F1-score*. Selanjutnya, dilakukan visualisasi menggunakan *word cloud* untuk menggambarkan kata-kata yang dominan pada masing-masing kelas sentimen.

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Pengumpulan data ulasan pengguna aplikasi Line Bank menggunakan teknik *web scraping* yang diambil dari platform Google Play Store dengan bantuan *library google-play-scraper* yang dijalankan pada Google Colaboratory dengan bahasa pemrograman *Python*. Pengambilan data ulasan dilakukan pada rentang waktu 1 Agustus 2023 hingga 21 November 2025 dengan total data yang diambil sebanyak 4.000 data ulasan Line Bank yang terdiri dari 11 kolom, yaitu *reviewid*, *username*, *userimage*, *reviewtext*, *rating*, *thumbsupcount*, *reviewcreatedversion*, *date*,

replycontent, *replieddate*, dan *appversion* tetapi pada penelitian ini hanya menggunakan 4 kolom yaitu *date*, *username*, *rating*, dan *review text* saja. Selanjutnya, seluruh data hasil *scraping* disimpan dalam format CSV.

4.2. Text Preprocessing

Setelah pengumpulan data selesai, kemudian dilakukan tahap *text preprocessing* untuk mempersiapkan hasil *scraping* data ulasan pengguna aplikasi Line Bank yang masih berupa teks mentah (*raw text*) agar menjadi lebih terstruktur, bersih, serta siap digunakan untuk proses analisis sentimen dan pemodelan klasifikasi. *Text preprocessing* meliputi beberapa tahapan, yaitu *cleaning*, *case folding*, *normalisasi*, *tokenisasi*, *stopword removal*, serta *stemming*. Contoh hasil *text preprocessing* disajikan pada Tabel 1 sebagai berikut.

Tabel 1. Tahapan *Text Preprocessing*

Tahapan	Teks
Ulasan	Aplikasi suka ngehang, tf / bayar gak diizinkan buat screenshot dari aplikasi. Giliran klik "Bagikan" malah keluar error, harus ngirim bukti pake foto hp lain bikin ribet aja
<i>Cleaning</i>	Aplikasi suka ngehang tf bayar gak diizinkan buat screenshot dari aplikasi giliran klik bagikan malah keluar error harus ngirim bukti pake foto hp lain bikin ribet aja
<i>Case Folding</i>	aplikasi suka ngehang tf bayar gak diizinkan buat screenshot dari aplikasi giliran klik bagikan malah keluar error harus ngirim bukti pake foto hp lain bikin ribet aja
<i>Normalization</i>	aplikasi suka ngehang transfer bayar tidak diizinkan buat screenshot dari aplikasi giliran klik bagikan malah keluar error harus ngirim bukti pakai foto hp lain bikin ribet saja
<i>Tokenization</i>	['aplikasi', 'suka', 'ngehang', 'transfer', 'bayar', 'tidak', 'diizinkan', 'buat', 'screenshot', 'dari', 'aplikasi', 'giliran', 'klik', 'bagikan', 'malah', 'keluar', 'error', 'harus', 'ngirim', 'bukti', 'pakai', 'foto', 'hp', 'lain', 'bikin', 'ribet', 'saja']
<i>Stopword Removal</i>	['aplikasi', 'suka', 'ngehang', 'transfer', 'bayar', 'diizinkan', 'screenshot', 'aplikasi', 'giliran', 'klik', 'bagikan', 'error', 'ngirim', 'bukti', 'pakai', 'foto', 'hp', 'bikin', 'ribet']
<i>Stemming</i>	aplikasi suka ngehang transfer bayar izin screenshot aplikasi giliran klik bagi error kirim bukti pakai foto hp bikin ribet

4.3. Pembobotan TF-IDF

Setelah tahapan *text preprocessing*, kemudian dilakukan pembobotan kata menggunakan metode pembobotan *Term Frequency Inverse Document Frequency* (TF-IDF). Pembobotan TF-IDF digunakan untuk mengonversi data ulasan pengguna aplikasi Line Bank yang berbentuk teks menjadi bentuk numerik dengan menggunakan fungsi *TfidfVectorizer* dari *library* (pustaka) *scikit-learn*. TF-IDF ini menghitung tingkat kepentingan sebuah kata di dalam suatu dokumen tertentu dengan mempertimbangkan kemunculan kata tersebut terhadap seluruh dokumen yang ada. Semakin sering suatu kata muncul pada satu dokumen, namun jarang muncul pada dokumen lain, maka semakin tinggi bobot TF-IDFnya. Berikut daftar sepuluh kata yang memiliki bobot TF-IDF paling tinggi ditampilkan pada Tabel 2.

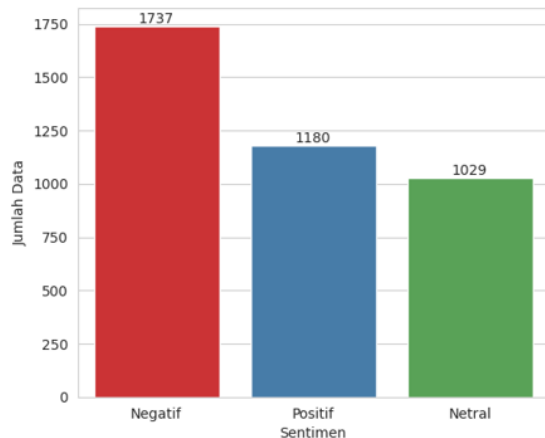
Tabel 2. 10 Kata dengan Bobot TF-IDF Tertinggi

Kata	Bobot TF-IDF
Aplikasi	0,05868
Bank	0.04524

Kata	Bobot TF-IDF
Bagus	0.04113
Line	0.03032
Update	0.02654
Dana	0.02522
OK	0.02228
Mantap	0.02215
Tarik	0.01963
Masuk	0.01962

4.4. Labelling

Tahap selanjutnya adalah proses *labelling* data, yang bertujuan untuk menentukan kategori sentimen pada setiap ulasan pengguna aplikasi Line Bank. Tahapan *labelling* pada penelitian ini menggunakan kamus *Indonesian Sentiment Lexicon (InSet Lexicon)* yang diunduh dari GitHub sebagai acuan penentuan polaritas sentimen pada setiap ulasan. Label sentimen dikategorikan menjadi tiga kelas, yaitu positif, negatif, dan netral. Distribusi data hasil *labelling* disajikan pada Gambar 2 sebagai berikut.



Gambar 2. Distribusi Data Hasil Labelling

Pada Gambar 2 di atas, dapat dilihat bahwa distribusi sentimen masih belum seimbang, di mana didominasi oleh kelas sentimen negatif dibandingkan dengan kelas sentimen positif maupun netral. Kondisi ini menunjukkan bahwa terdapat permasalahan ketidakseimbangan kelas (*class imbalance*), sehingga diperlukan teknik untuk mengatasi masalah ketidakseimbangan kelas tersebut.

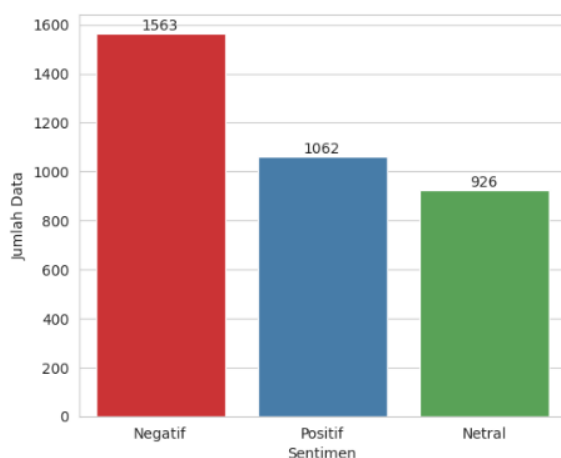
4.5. Data Splitting

Tahap berikutnya adalah *data splitting* (pembagian data) dengan membagi data ulasan pengguna aplikasi Line Bank menjadi *data training* dan *data testing*. Rasio pembagian data, yaitu 90:10 dengan 90% untuk *data training*, yaitu sebanyak 3551 serta 10% untuk *data testing*, yaitu sebanyak 395. Pembagian data disajikan pada Tabel 3 sebagai berikut.

Tabel 3. Pembagian Data

Data	Presentase	Jumlah Data
Data Training	90%	3551
Data Testing	10%	395
Total	100%	3946

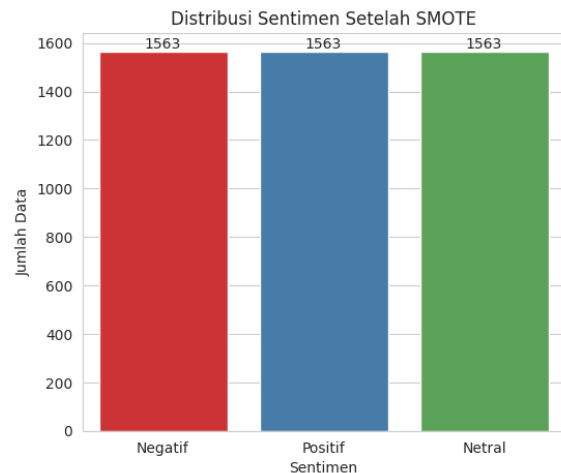
4.6. SMOTE



Gambar 3. Distribusi Sentimen Sebelum SMOTE

Setelah dilakukan pembagian data, selanjutnya diterapkan *Synthetic Minority Oversampling Technique* (SMOTE) untuk mengatasi masalah ketidakseimbangan kelas (*class imbalance*) antara kelas positif, negatif, dan netral dalam dataset ulasan pengguna aplikasi Line Bank. SMOTE ini hanya diterapkan pada *data training* saja.

Berdasarkan Gambar 3, menunjukkan bahwa distribusi sentimen sebelum dilakukan teknik SMOTE didominasi oleh kelas negatif dengan jumlah data 1563, positif 1062, serta netral 926.



Gambar 4. Distribusi Sentimen Setelah SMOTE

Setelah diterapkan SMOTE, terlihat bahwa pada Gambar 4 distribusi sentimen menjadi lebih seimbang antara kelas negatif, positif, dan netral yang sama-sama memiliki jumlah 1563 data. Hal ini menunjukkan bahwa SMOTE telah berhasil mengatasi masalah ketidakseimbangan kelas (*class imbalance*) dalam dataset dengan membuat data sintesis pada kelas minoritas, sehingga setiap kelas memiliki proporsi yang sama.

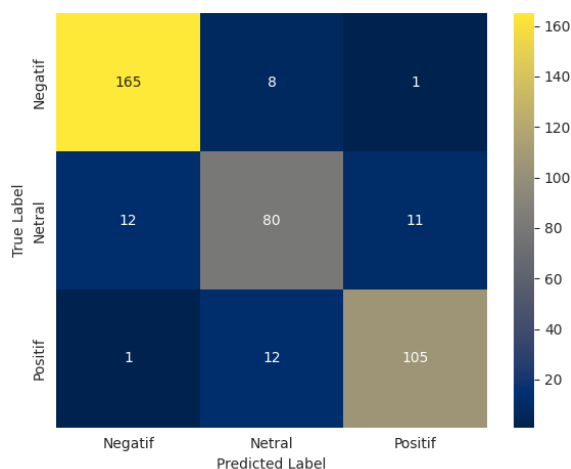
4.7. Support Vector Machine (SVM)

Tahap selanjutnya adalah klasifikasi menggunakan *Support Vector Machine* (SVM). Pada penelitian ini, pengujian SVM dilakukan pada berbagai jenis kernel, meliputi kernel linear, Radial Basis Function (RBF), polynomial, dan sigmoid, untuk menentukan kernel dengan kinerja terbaik. Evaluasi model SVM dilakukan menggunakan beberapa metrik, yaitu akurasi, *precision*, *recall*, serta *F1-score*. Hasil evaluasi model SVM dengan beberapa kernel disajikan pada Tabel 4 sebagai berikut.

Tabel 4. Hasil Evaluasi SVM Berbagai Kernel

Kernel	Accuracy	Precision	Recall	F1-Score
Linear	85,57%	85,91%	85,57%	85,66%
RBF	85,32%	85,28%	85,32%	85,04%
Polynomial	77,72%	80,12%	80,12%	78,10%
Sigmoid	83,29%	84%	83,29%	83,49%

Pada Tabel 4 di atas, menunjukkan bahwa kernel linear memperoleh performa terbaik dibandingkan kernel lainnya. Nilai akurasi yang diperoleh kernel linear sebesar 85,57%, presisi 85,91%, *recall* 85,57%, serta *F1-score* 85,66%. Untuk kernel RBF nilai akurasi yang diperoleh sebesar 85,32%, presisi 85,28%, *recall* 85,32%, serta *F1-score* 85,04%. Untuk kernel polynomial nilai akurasi yang diperoleh sebesar 77,72%, presisi 80,12%, *recall* 80,12%, dan *F1-score* 78,10%. Sedangkan, kernel sigmoid nilai akurasi yang diperoleh sebesar 83,29%, presisi 84%, *recall* 83,29%, dan *F1-score* 83,49%. Kemudian, evaluasi model dilakukan untuk kernel dengan hasil performa terbaik dengan menggunakan *confusion matrix*. Hasil *confusion matrix* model SVM dengan kernel linear dapat dilihat pada Gambar 5 berikut.



Gambar 5. Confusion Matrix SVM Kernel Linear

Berdasarkan Gambar 5 di atas, menunjukkan bahwa model mampu mengklasifikasikan sebanyak 160 dari total 174 data pada kelas negatif secara benar sebagai negatif. Kemudian, untuk kelas netral sebanyak 81 data dari total 103 data pada kelas netral berhasil diklasifikasi secara benar sebagai netral. Selanjutnya, dari total 118 data kelas positif, sebanyak 97 data berhasil diklasifikasikan dengan benar sebagai positif.

4.8. GridSearchCV

Optimasi *hyperparameter* pada model SVM dilakukan dengan menggunakan *GridSearchCV* untuk mencari kombinasi parameter terbaik pada model SVM. Parameter dan nilai yang diuji dalam proses *hyperparameter tuning* menggunakan *GridSearchCV* disajikan pada Tabel 5 berikut.

Tabel 5. Search Space Hyperparameter SVM

Parameter	Nilai
Kernel	[linear, rbf, poly, sigmoid]
C	[0.1, 1, 10]
Gamma	[scale, auto, 0.01, 0.1, 1]

Setelah menentukan parameter dan nilai yang diuji, selanjutnya dilakukan teknik *Stratified K-Fold Cross Validation* dengan 10 *fold*. *Stratified K-Fold Cross Validation* digunakan untuk menjaga proporsi kelas sentimen agar tetap seimbang pada setiap *fold*. Dengan menerapkan 10-*fold Stratified Cross Validation*, data latih dibagi menjadi sepuluh bagian. Setiap kombinasi parameter SVM dievaluasi secara bergantian dengan menggunakan satu *fold* sebagai data uji dan sembilan *fold* lainnya sebagai data latih. Proses ini dilakukan secara berulang hingga sepuluh kali. Seluruh kombinasi parameter yang sudah ditetapkan sebelumnya kemudian dievaluasi dengan *GridSearchCV* untuk menentukan kombinasi parameter terbaik. Hasil parameter terbaik disajikan pada Gambar 6 berikut.

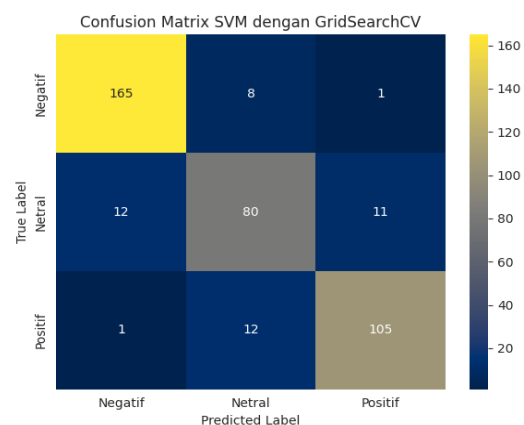
Best parameters: {'svm_C': 10, 'svm_gamma': 'scale', 'svm_kernel': 'linear'}

Gambar 6. Hasil Parameter Terbaik

Pada Gambar 6 di atas, menunjukkan bahwa kombinasi parameter terbaik (*best parameters*) yang dihasilkan melalui proses optimasi dengan *GridSearchCV* adalah $C = 10$, $\gamma = \text{scale}$, $\text{kernel} = \text{linear}$.

4.9. SVM dengan GridSearchCV

Setelah optimasi *hyperparameter tuning* menggunakan *GridSearchCV*, selanjutnya penerapan model *Support Vector Machine* (SVM) dengan parameter terbaik (*best parameters*) hasil *GridSearchCV* yang telah diperoleh. Model SVM kemudian dilatih menggunakan parameter terbaik tersebut, yaitu $C = 10$, $\gamma = \text{scale}$, dan $\text{kernel} = \text{linear}$, untuk memperoleh model dengan performa paling optimal. Setelah model SVM dilatih, selanjutnya dilakukan pengujian menggunakan data uji untuk mengevaluasi performa model klasifikasi sentimen terhadap ulasan pengguna aplikasi Line Bank yang lebih akurat. Evaluasi model dilakukan menggunakan *confusion matrix*. Hasil *confusion matrix* model SVM dengan *GridSearchCV* disajikan pada Gambar 7 berikut.



Gambar 7. Confusion Matrix SVM dengan GridSearchCV

Berdasarkan Gambar 7 di atas, menunjukkan hasil bahwa pada kelas negatif sebanyak 165 data berhasil diklasifikasikan dengan benar sebagai negatif dari total 174 data negatif. Kemudian, untuk kelas netral sebanyak 80 data berhasil diklasifikasikan dengan benar sebagai netral dari total 103 data netral. Selanjutnya, pada kelas positif sebanyak 105 data berhasil diklasifikasikan dengan benar sebagai positif dari total 118 data positif.

```

=== Classification Report SVM dengan GridSearchCV ===

```

	precision	recall	f1-score	support
Negatif	0.9270	0.9483	0.9375	174
Netral	0.8000	0.7767	0.7882	103
Positif	0.8974	0.8898	0.8936	118
accuracy			0.8861	395
macro avg	0.8748	0.8716	0.8731	395
weighted avg	0.8850	0.8861	0.8855	395

Gambar 8. Classification Report SVM dengan GridSearchCV

Pada Gambar 8 di atas, menunjukkan bahwa model SVM dengan *GridSearchCV* memperoleh nilai *accuracy* sebesar 88,61%, *precision* 88,50%, *recall* 88,61%, serta *F1-score* 88,55%.

4.10. Perbandingan SVM Sebelum Dan Sesudah Gridsearchcv

Selanjutnya, untuk mengetahui pengaruh dari penerapan *hyperparameter tuning* menggunakan *GridSearchCV* terhadap kinerja model SVM, dilakukan perbandingan hasil evaluasi SVM sebelum dan sesudah *hyperparameter tuning* *GridSearchCV*. Evaluasi performa model SVM dilakukan menggunakan beberapa metrik, yaitu akurasi, presisi, *recall*, serta *F1-score*. Hasil perbandingan disajikan pada Tabel 6 berikut.

Tabel 6. Perbandingan Kinerja SVM Sebelum dan Sesudah GridSearchCV

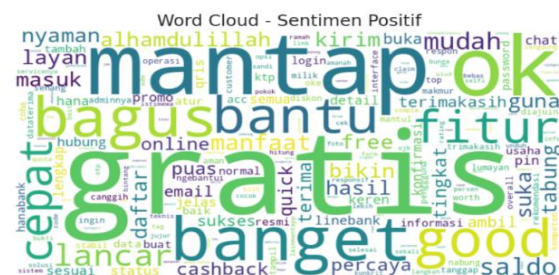
Model	Accuracy	Precision	Recall	F1-score
SVM	85,57%	85,91%	85,57%	85,66%
SVM + GridSearchCV	88,61%	88,50%	88,61%	88,55%

Berdasarkan Tabel 6 di atas, menunjukkan bahwa sebelum dilakukan optimasi *hyperparameter tuning* menggunakan *GridSearchCV*, diperoleh nilai akurasi pada model SVM sebesar 85,57%, presisi 85,91%, *recall* 85,57%, serta *F1-score* 85,66%. Setelah diterapkan optimasi *hyperparameter tuning* menggunakan *GridSearchCV*, terlihat bahwa adanya peningkatan kinerja model SVM dengan nilai akurasi menjadi 88,61%, presisi menjadi 88,50%, *recall* menjadi 88,61%, serta *F1-score* juga menjadi 88,55%. Sehingga, nilai akurasi dan *recall* sama-sama meningkat sebesar 3,04%, nilai presisi meningkat sebesar 2,59%, dan nilai *F1-score* meningkat sebesar 2,89%. Hasil ini menunjukkan bahwa penerapan *hyperparameter tuning* *GridSearchCV* mampu

meningkatkan kinerja dari model SVM secara konsisten pada semua metrik evaluasi.

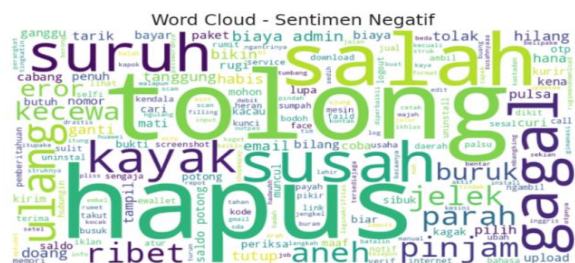
4.11. Visualisasi Word Cloud

Visualisasi *word cloud* digunakan untuk menampilkan kata-kata apa saja yang paling banyak atau paling sering muncul pada ulasan pengguna aplikasi line bank pada masing-masing kelas sentimen, yaitu positif, negatif, dan netral. Semakin sering kata tersebut muncul dalam dokumen maka ukuran kata tersebut akan semakin besar. Berikut *word cloud* untuk sentimen positif, negatif, dan netral pada ulasan pengguna aplikasi Line Bank.



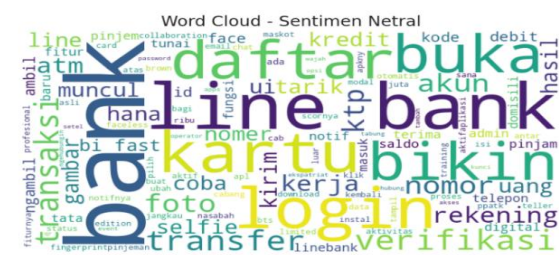
Gambar 9. Word Cloud Sentimen Positif

Pada Gambar 9 di atas, terlihat bahwa kata yang paling sering muncul antara lain “gratis”, “mantap”, “bagus”, “bantu”, “cepat”. Dominasi kata-kata tersebut mengindikasikan bahwa pengguna memberikan respons positif terhadap kemudahan dalam penggunaan aplikasi, kualitas fitur yang disediakan, serta kecepatan layanan.



Gambar 10. Gambar Word Cloud Sentimen Negatif

Pada Gambar 10 di atas, terlihat bahwa kata-kata paling sering muncul antara lain “tolong”, “hapus”, “salah”, “gagal”, “susah”. Dominasi penggunaan kata-kata tersebut mengindikasikan bahwa adanya keluhan pengguna terkait dengan kendala teknis dan kesulitan dalam penggunaan aplikasi.



Gambar 11. Word Cloud Sentimen Netral

Pada Gambar 11 di atas, terlihat bahwa kata-kata yang paling sering muncul antara lain “line bank”, “login”, “buka”, “daftar”, “kartu”. Kata-kata paling dominan tersebut menunjukkan bahwa ulasan pengguna yang bersifat informatif dan deskriptif mengenai proses penggunaan dan fitur dasar dari aplikasi.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang dilakukan dalam mengoptimasi kinerja *Support Vector Machine* berbasis SMOTE dengan menggunakan *hyperparameter tuning GridSearchCV* pada ulasan pengguna aplikasi Line Bank di Google Play Store diperoleh bahwa perbandingan kinerja SVM berbasis SMOTE sebelum dan sesudah dioptimasi dengan *hyperparameter tuning* menggunakan *GridSearchCV* menunjukkan adanya peningkatan berdasarkan hasil metrik evaluasi. Sebelum dilakukan optimasi *hyperparameter tuning GridSearchCV* pada model SVM dengan kernel linear sebagai kernel terbaik diperoleh akurasi dan *recall* sebesar 85,57%, *precision* sebesar 85,91%, serta F1-score sebesar 85,66%. Setelah diterapkan optimasi *hyperparameter tuning* menggunakan *GridSearchCV* dengan kombinasi parameter terbaik yaitu kernel = linear, C = 10, dan gamma = scale, kinerja model mengalami peningkatan yang signifikan dengan akurasi dan *recall* menjadi sebesar 88,61%, presisi sebesar 88,50%, serta F1-score sebesar 88,55%. Hal ini menunjukkan bahwa optimasi *hyperparameter tuning* menggunakan *GridSearchCV* terbukti efektif mampu meningkatkan kinerja dari model SVM secara konsisten pada semua metrik evaluasi pada ulasan pengguna aplikasi Line Bank di Google Play Store.

Hasil analisis *word cloud* pada sentimen positif didominasi oleh kepuasan pengguna pada kemudahan dalam penggunaan aplikasi, kualitas fitur yang disediakan, serta kecepatan layanan. Untuk *word cloud* sentimen negatif didominasi oleh keluhan pengguna mengenai kendala teknis dan kesulitan dalam penggunaan aplikasi. Sedangkan, *word cloud* sentimen netral didominasi oleh ulasan pengguna yang bersifat informatif dan deskriptif mengenai proses penggunaan serta fitur dasar dari aplikasi tanpa mengekspresikan penilaian positif maupun negatif secara jelas.

Saran untuk penelitian selanjutnya bisa membandingkan metode SVM dengan metode klasifikasi lain seperti *Random Forest*, KNN, atau metode berbasis *deep learning* seperti LSTM, serta dapat menggunakan teknik penanganan ketidakseimbangan lain seperti ADASYN yang lebih adaptif dalam menangani masalah ketidakseimbangan pada kelas minoritas.

DAFTAR PUSTAKA

- [1] T. Ayunda and E. Y. Metekohy, “Pengaruh System Availability dan Reliability Terhadap Minat

Penggunaan Ulang LINE Bank Pada Bank Hana,” 2023.

- [2] B. Liu, *Sentiment Analysis and Opinion Mining*. Springer Nature Switzerland AG, 2022. doi: 10.1007/978-3-031-02145-9.
- [3] M. F. Ilham, K. D. Annurahma, P. Wirayuda, and Rudiman, “Analisis Kepuasan Pengguna Aplikasi Donorku dengan Pendekatan Random Forest dengan SMOTE,” vol. 6, no. 3, pp. 508–513, Aug. 2024.
- [4] F. A. Indriyani, A. Fauzi, and S. Faisal, “Analisis Sentimen Aplikasi Tiktok Menggunakan Algoritma Naïve Bayes dan Support Vector Machine,” *TEKNOSAINS: Jurnal Sains, Teknologi dan Informatika*, vol. 10, no. 2, pp. 176–184, Jul. 2023, doi: 10.37373/tekno.v10i2.419.
- [5] S. Watmah, Suryanto, and Martias, “Komparasi Metode K-NN, Support Vector Machine, Dan Random Forest Pada E-Commerce Shopee,” May 2021. [Online]. Available: <http://jurnal.bsi.ac.id/index.php/insantek>
- [6] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, “Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis,” *Informatics*, vol. 8, no. 4, p. 79, Nov. 2021, doi: 10.3390/informatics8040079.
- [7] L. M. M. Matin, “Hyperparameter Tuning menggunakan *GridSearchCV* pada Random Forest untuk Deteksi Malware,” *JURNAL MULTINETICS*, vol. 9, no. 1, May 2023.
- [8] R. Muzayanah, D. A. A. Pertiwi, M. Ali, and M. A. Muslim, “Comparison of *GridSearchCV* and bayesian hyperparameter optimization in random forest algorithm for diabetes prediction,” *Journal of Soft Computing Exploration*, vol. 5, no. 1, pp. 86–91, Apr. 2024, doi: 10.52465/josce.v5i1.308.
- [9] A. Toha, P. Purwono, and W. Gata, “Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter *GridSearch CV*,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 1, pp. 12–21, May 2022, doi: 10.12928/biste.v4i1.6079.
- [10] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, “Sentiment analysis based on deep learning: A comparative study,” *Electronics (Switzerland)*, vol. 9, no. 3, Mar. 2020, doi: 10.3390/electronics9030483.
- [11] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” 2002.
- [12] J. Rusman, B. Z. Haryati, and A. Michael, “Optimisasi Hiperparameter Tuning pada Metode *Support Vector Machine* untuk Klasifikasi Tingkat Kematangan Buah Kopi,” *Jurnal Komputer dan Informatika*, vol. 11, no. 2, pp. 195–202, Oct. 2023, doi: 10.35508/jicon.v11i2.12571.
- [13] H. Byun and S.-W. Lee, “Applications of Support Vector Machines for Pattern Recognition: A Survey,” *LNCS*, vol. 2388, pp. 213–236, 2002, doi: 10.1007/3-540-45665-1_17.