

## ARTICLE

# Optimasi Metode Klasifikasi Menggunakan FastText dan Grid Search pada Analisis Sentimen Ulasan Aplikasi SeaBank

## *Optimization of Classification Method Using FastText and Grid Search for Sentiment Analysis of SeaBank App Reviews*

Bagas Restya Ermawan\* dan Nuri Cahyono

Informatika, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia

\*Penulis Korespondensi: restyaaa@students.amikom.ac.id

(Disubmit 02-12-24; Diterima 02-01-25; Dipublikasikan online pada 05-02-25)

### Abstrak

Perkembangan pesat teknologi digital telah memberikan dampak signifikan pada sektor perbankan, termasuk aplikasi perbankan digital seperti SeaBank. Namun, pengguna melaporkan masalah seperti hilangnya saldo dan keterlambatan transfer, yang memerlukan analisis sentimen berbasis ulasan di Google Play Store. Penelitian ini menganalisis ulasan pengguna SeaBank menggunakan metode klasifikasi Support Vector Machine (SVM) dan K-Nearest Neighbors (KNN), dioptimalkan dengan Grid Search dan ekstraksi fitur FastText. Sebanyak 15.000 ulasan (Juli–September 2024) dilabeli secara manual (positif, negatif, netral) dan diseimbangkan menggunakan SMOTE. Optimasi Grid Search meningkatkan akurasi model, dengan SVM mencapai 92% dan KNN 91%. SVM yang dioptimalkan (C: 100, kernel: 'rbf') menunjukkan performa terbaik (presisi 0,93, recall 0,92, F1-score 0,92). Peningkatan akurasi ini bermanfaat bagi pengelola SeaBank untuk meningkatkan pengalaman pengguna dan kepercayaan terhadap layanan aplikasi.

**Kata kunci:** Analisis sentimen; Seabank; Support Vector Machine; K-Nearest Neighbors; Grid Search

### Abstract

The rapid development of digital technology has significantly impacted the banking sector, including digital banking applications like SeaBank. However, users have reported issues such as balance discrepancies and delayed transfers, necessitating sentiment analysis based on reviews from the Google Play Store. This study analyzes SeaBank user reviews using Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classification methods, optimized with Grid Search and FastText feature extraction. A total of 15,000 reviews (July–September 2024) were manually labeled (positive, negative, neutral) and balanced using SMOTE. Grid Search optimization enhanced model accuracy, with SVM achieving 92% and KNN 91%. The optimized SVM (C: 100, kernel: 'rbf') demonstrated the best performance (precision 0.93, recall 0.92, F1-score 0.92). These accuracy improvements benefit SeaBank's management by enhancing user experience and building trust in the application.

**KeyWords:** Sentiment Analysis; Seabank; Support Vector Machine; K-Nearest Neighbors; Grid Search

## 1. Pendahuluan

Kemajuan teknologi digital telah membawa perubahan besar dalam sektor perbankan. Bank digital menjadi solusi praktis sebagai layanan perbankan bagi masyarakat Indonesia saat ini [1]. Salah satu aplikasi dalam bidang ini adalah SeaBank, yang dapat diunduh melalui *Google Play Store*. SeaBank merupakan aplikasi perbankan digital yang membantu berbagai aktivitas finansial [2]. Namun, beberapa pengguna

This is an Open Access article - copyright on authors, distributed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY SA) (<http://creativecommons.org/licenses/by-sa/4.0/>)

**How to Cite:** B. R. Ermawan *et al.*, "Optimasi Metode Klasifikasi Menggunakan FastText dan Grid Search pada Analisis Sentimen Ulasan Aplikasi SeaBank", *JIKO (JURNAL INFORMATIKA DAN KOMPUTER)*, Volume: 9, No.1, Pages 226–238, Februari 2025, doi: 10.26798/jiko.v9i1.1523.

SeaBank melaporkan masalah serius seperti kehilangan saldo atau keterlambatan transfer. Masalah ini, jika tidak ditangani, dapat memengaruhi reputasi SeaBank secara signifikan, mengurangi tingkat kepercayaan pengguna dan potensi pertumbuhan aplikasi.

*Google Play Store* sebagai *platform* distribusi aplikasi memungkinkan pengguna memberikan ulasan dan penilaian secara langsung. Analisis sentimen ulasan pengguna menjadi penting untuk memahami persepsi pelanggan, menangani keluhan, dan meningkatkan kualitas layanan. Teknologi *Natural Language Processing* (NLP) dengan metode klasifikasi seperti *Support Vector Machine* (SVM) dan *K-Nearest Neighbors* (KNN) telah terbukti efektif untuk analisis sentimen [3, 4, 5]. Namun, kualitas hasil analisis sangat bergantung pada optimasi model dan teknik ekstraksi fitur yang digunakan.

Untuk meningkatkan performa klasifikasi, optimasi model melalui *Hyperparameter Tuning* seperti *Grid Search* sangat diperlukan. *Grid Search* menawarkan pendekatan sistematis untuk menemukan kombinasi parameter terbaik, dibandingkan metode seperti *Random Search* yang bersifat acak dan kurang terstruktur [6, 7]. Selain itu, penggunaan *FastText* sebagai teknik ekstraksi fitur memberikan keunggulan dalam memahami konteks kata dan makna secara mendalam, menghasilkan representasi teks yang lebih efektif untuk klasifikasi dibandingkan metode konvensional seperti TF-IDF [8].

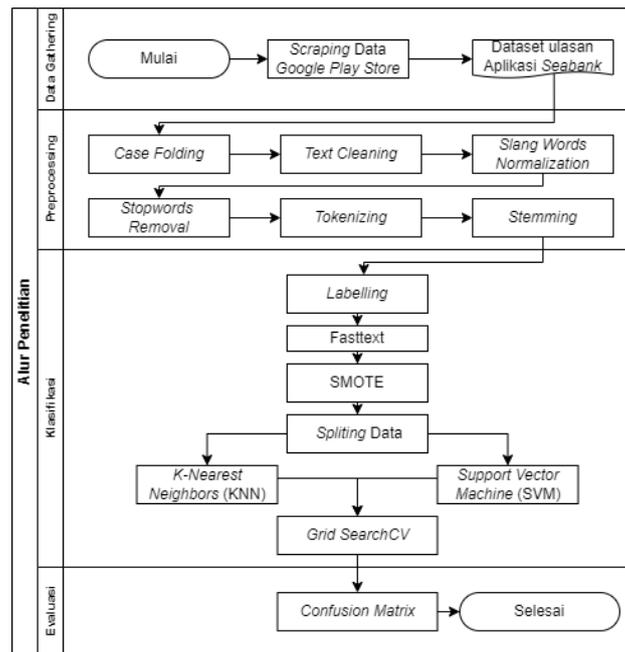
Pada penelitian yang dilakukan oleh Sri Rahayu et al., penerapan KNN untuk analisis sentimen pada aplikasi Flip mendapatkan akurasi 77,67% [1]. Sedangkan, pada penelitian yang dilakukan Dea Oktavia et al., menggunakan SVM untuk analisis sentimen terhadap sistem E-Tilang mendapatkan akurasi 74,20% [2]. Sementara itu, Fiki et al. melakukan perbandingan SVM dengan KNN dan mendapatkan akurasi SVM sebesar 76,5% dan KNN sebesar 61,3% dalam konteks analisis sentimen pemilihan presiden 2019 [3]. Sedangkan pada penelitian yang dilakukan oleh Anna Baita et al., peneliti menggunakan algoritma SVM dan KNN untuk menganalisis sentimen mengenai vaksin SINOVAQ dan mendapatkan hasil 70% untuk SVM, sedangkan KNN mendapatkan hasil akurasi 56% [4]. Di sisi lain, Muhammad Alfathan dan Harlis Setiyowati menggunakan algoritma *Decision Tree* dan KNN untuk mengklasifikasi daerah berdasarkan produksi listrik, dengan hasil akurasi *Decision Tree* dan KNN sebesar 85,78% dan 80,34% [5]. Sedangkan pada penelitian yang dilakukan oleh Ferry Putrawansyah dan Tri Susanti, menggunakan metode SVM untuk mengklasifikasi jenis jambu biji dan mendapatkan akurasi sebesar 80% [6]. Dari berbagai studi tersebut, terlihat bahwa metode SVM dan KNN memiliki potensi yang signifikan, tetapi hasil akurasinya masih dapat ditingkatkan melalui optimasi lebih lanjut seperti *Grid Search* dan penggunaan teknik ekstraksi fitur yang lebih kuat seperti *FastText* [9].

Berdasarkan paparan di atas, penelitian ini bertujuan untuk melakukan optimasi terhadap metode klasifikasi SVM dan KNN menggunakan *Particle Swarm Optimization* dan *FastText* untuk meningkatkan akurasi pada analisis sentimen ulasan aplikasi SeaBank. Pada penelitian ini, kami akan melakukan analisis sentimen pada ulasan aplikasi *mobile* SeaBank dengan menggunakan metode klasifikasi SVM dan KNN yang dioptimasi menggunakan *Grid Search* serta ekstraksi fitur *FastText*.

## 2. Metode Penelitian

### 2.1 Alur Penelitian

Alur penelitian dalam optimasi metode klasifikasi menggunakan *Fasttext* dan *Grid Search* pada analisis sentimen ulasan aplikasi Seabank terdiri dari beberapa tahapan yang ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

Penelitian ini dilakukan dengan beberapa langkah untuk mengoptimasikan metode klasifikasi menggunakan *Grid Search*. Berikut adalah alur penelitian yang dilakukan:

a. *Data Gathering*

Pengumpulan data dalam penelitian ini dilakukan dengan teknik *scraping* dengan menggunakan *library* dari *python* yaitu *Google-Play-Scraper*. Pada tahap ini, data mentah yang digunakan merupakan data ulasan yang diambil dari *Google Play Store* pada aplikasi Seabank.

b. *Text Preprocessing*

Data yang diperoleh melalui proses *scraping* selanjutnya masuk ke tahapan preprocessing dengan tujuan menjadikan data ulasan mudah di proses dan klasifikasi, diantaranya adalah dengan *case folding* yang dimana akan mengubah semua huruf menjadi huruf kecil (*lowercase*), Menghapus tanda baca, angka, dan karakter khusus (*cleaning*), Mengubah *slang word* ke bahasa baku, menghapus semua kata hubung (*stopwords removal*), *tokenizing*, mengubah suatu kata ke kata dasarnya (*stemming*).

c. *Labelling* dan ekstraksi fitur

Pada penelitian ini, dilakukan *labelling* pada ulasan *Google Play Store* berdasarkan *rating* pengguna, untuk mengelompokkan sentimen menjadi tiga kategori: positif, netral, dan negatif. Ulasan dengan *rating* 4-5 dilabel sebagai positif karena mencerminkan kepuasan pengguna, *rating* 3 sebagai netral karena dianggap tidak terlalu puas atau kecewa, dan *rating* 1-2 sebagai negatif karena menunjukkan ketidakpuasan. *Labelling* ini akan digunakan sebagai target variabel (*y*) dalam proses analisis sentimen, sementara ulasan yang telah diproses menjadi fitur (*X*).

*Fasttext* adalah model *word embedding* yang mampu menangkap informasi semantik dari kata-kata dalam teks, dengan mempertimbangkan sub-kata atau *n-gram*, sehingga lebih efektif dalam menangani kata-kata baru atau kata-kata dengan kesalahan penulisan. Dengan *Fasttext*, setiap kata dalam teks ulasan akan diubah menjadi vektor yang merepresentasikan makna kata tersebut dalam ruang vektor [7]. Vektor-vektor ini kemudian digunakan sebagai *input* fitur (*X*) dalam proses pelatihan model *machine learning*.

d. *SMOTE* dan *splitting data*

Pada tahap ini, *SMOTE* digunakan untuk menyeimbangkan jumlah sampel pada kelas minoritas dengan memilih sampel sehingga setara dengan kelas mayoritas. Pada tahapan ini, data kelas minoritas akan ditambah jumlahnya melalui *oversampling*, sehingga dataset menjadi lebih seimbang antara kelas minoritas dengan kelas mayoritas. *SMOTE* dipilih karena menghasilkan data sintesis melalui interpolasi sampel yang ada, sehingga lebih realistis dibandingkan *Random Oversampling* yang hanya meng-

gandakan sampel kelas minoritas. Selain itu, SMOTE mengurangi risiko *overfitting* yang sering terjadi pada metode penggantian sampel sederhana. Pada tahap ini juga, dataset akan dibagi menjadi dua bagian, yaitu data latih (*training*) dan data uji (*testing*). Data latih akan digunakan untuk melatih model yang akan dibuat, sementara data uji digunakan untuk menguji kinerja model yang sudah dibuat dalam melakukan klasifikasi.

e. *Modelling*

Pada tahap ini, peneliti menggunakan dataset yang sudah dibagi menjadi data latih dan data uji, untuk melatih model dengan kumpulan dataset yang tersedia, lalu dilakukan pengujian untuk menguji kemampuan model dengan menggunakan data uji.

f. Optimasi *Grid Search* Dalam *machine learning*, *Grid Search* digunakan untuk menemukan parameter terbaik bagi suatu model. Prosesnya melibatkan pencarian menyeluruh melalui sejumlah nilai parameter yang ditentukan sebelumnya, kemudian memilih kombinasi parameter yang memberikan kinerja terbaik berdasarkan metrik evaluasi yang digunakan [10].

g. Evaluasi Setelah tahap klasifikasi dan optimasi model, selanjutnya dilakukan evaluasi dengan menggunakan *confusion matrix*. Dengan menghitung nilai akurasi, presisi, *recall*, dan *F1-score* dapat memberikan gambaran menyeluruh terkait performa model tersebut.

## 2.2 Support Vector Machine (SVM)

*Support Vector Machine* (SVM) adalah salah satu teknik *supervised learning* yang digunakan untuk klasifikasi dan regresi [11]. Prinsip dasar yang mendasari metode SVM adalah klasifikasi linear, yang selanjutnya disempurnakan melalui integrasi mekanisme trik *kernel*. Kelebihan metode SVM terletak pada kapasitasnya yang luar biasa untuk menggambarkan *hyperplane* yang berbeda, sehingga mengoptimalkan margin yang memisahkan berbagai kelas dengan presisi [8]. Namun, metode SVM juga memiliki kelemahan, yaitu apabila terdapat fitur yang mirip atau sangat serupa, hal tersebut dapat mempengaruhi tingkat akurasi secara signifikan [8]. Persamaan *hyperplane* dalam bentuk umum adalah:

$$b + wx = 0 \tag{1}$$

Di mana:

- $w$  = parameter *hyperlane* yang dicari (garis tegak lurus antara garis *hyperlane* dan titik *support vector*)
- $x$  = data *input* SVM ( $x_1$  = index kata,  $x_2$  = bobot kata)
- $b$  = parameter yang dicari (nilai bias)

## 2.3 K-Nearest Neighbors (KNN)

Metode *K-Nearest Neighbor* (KNN) adalah pendekatan non-parametrik yang digunakan untuk tugas klasifikasi dengan mempertimbangkan k tetangga terdekat, serta untuk analisis regresi. Algoritma *K-Nearest Neighbor* merupakan teknik yang dimanfaatkan untuk mengklasifikasi objek dengan mengidentifikasi objek yang paling dekat dengan suatu objek tertentu, berdasarkan pada himpunan data pelatihan [12]. Jarak antara dua tetangga terdekat berdasarkan nilai similaritas dapat dihitung menggunakan jarak *Euclidean*, yang didefinisikan sebagai berikut:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \tag{2}$$

Di mana:

- $d(x, y)$  = Jarak *Euclidean* antara titik  $x$  dan  $y$ .
- $x_i$  = Koordinat ke- $i$  dari titik  $x$
- $y_i$  = Koordinat ke- $i$  dari titik  $y$
- $n$  = Dimensi atau jumlah fitur dari titik data
- $\sum_{i=1}^n$  = Notasi *sigma* yang berarti penjumlahan dari  $i=1$  hingga  $n$ .
- $(x_i - y_i)^2$  = Kuadrat dari selisih antara koordinat  $x_i$  dan  $y_i$ , yang mengukur perbedaan antara kedua titik pada dimensi ke- $i$ .

## 2.4 Fasttext Word Embedding

*FastText* adalah *library* yang dikembangkan oleh Facebook yang dapat digunakan untuk *word embedding*. *FastText* merupakan pengembangan dari *library Word2Vec* yang telah lebih lama dikenal sebagai metode untuk proses *word embedding*. *FastText* dipilih sebagai metode *word embedding* dalam penelitian ini karena keunggulannya dibandingkan metode lain seperti *Word2Vec* dan *GloVe*. *FastText* mampu menangkap informasi semantik kata dengan mempertimbangkan sub-kata atau n-gram, yang membuatnya lebih efektif dalam menangani kata-kata baru atau kata dengan kesalahan penulisan. Berbeda dengan *Word2Vec* yang hanya menghasilkan representasi kata berbasis konteks keseluruhan tanpa mempertimbangkan struktur internal kata, *FastText* memanfaatkan elemen-elemen sub-kata untuk menghasilkan representasi yang lebih mendalam. Sementara itu, *GloVe* yang mengandalkan hubungan statistik antar kata tidak sefleksibel *FastText* dalam menangani variasi kata atau kesalahan penulisan. Dengan *FastText*, ulasan pengguna yang sering mengandung kata-kata tidak baku dapat dikonversi menjadi vektor yang lebih representatif, sehingga meningkatkan kualitas analisis sentimen. Jika kita memiliki kata  $w$  dan mengonversinya menjadi n-gram  $g_1, g_2, g_3, \dots, g_k$ , maka representasi vektor kata dapat dinyatakan sebagai:

$$v_w = \frac{1}{|g|} \sum_{i=1}^{|g|} v_{g_i} \quad (3)$$

Di mana:

- $W - w$  = vektor representasi untuk kata  $W$
- $|g|$  = Jumlah n-gram dari kata  $W$
- $v_{g_i}$  = vektor representasi untuk n-gram ke- $i$ .

## 2.5 Grid Search CV

*Grid Search CV* mencoba berbagai kombinasi *hyperparameter* yang telah ditetapkan dan menghitung rata-rata nilai *cross-validation* untuk setiap kombinasinya, *Hyperparameter* sendiri merupakan parameter dalam model pembelajaran mesin yang tidak dipelajari dari data, tetapi ditentukan sebelum proses pelatihan dimulai. Kombinasi dengan rata-rata nilai *cross-validation* tertinggi kemudian digunakan dalam model. Proses ini mencakup perhitungan tingkat akurasi untuk setiap kombinasi parameter dalam *grid*, dengan tujuan untuk menemukan nilai parameter yang paling optimal, yaitu yang menghasilkan akurasi tertinggi [13].

Rumus *Grid Search* :

$$\text{GridSearch}(P_1, P_2, \dots, P_n) = \arg \max_{p \in P} \text{Score}(p) \quad (4)$$

Di mana:

- $P_1, P_2, P_3, \dots, P_n$  = Himpunan nilai dari *hyperparameter* yang dicoba. Setiap  $P_i$  merepresentasikan daftar nilai yang mungkin untuk *hyperparameter* ke- $i$ .
- $p$  = salah satu kombinasi nilai *hyperparameter* dari ruang parameter  $p$
- $\text{Score}(p)$  = fungsi evaluasi yang menghitung performa model untuk suatu kombinasi parameter  $p$  yang biasanya didasarkan pada metrik seperti akurasi, presisi, recall, atau metrik lainnya.
- $\arg \max_{p \in P}$  = pemilihan parameter  $p$  yang memaksimalkan fungsi evaluasi, yaitu memilih kombinasi *hyperparameter* yang memberikan nilai performa tertinggi.

## 2.6 Confusion Matrix

*Confusion Matrix* adalah salah satu alat analisis prediktif yang menampilkan serta membandingkan nilai aktual dengan hasil prediksi model. Alat ini dapat digunakan untuk menghasilkan metrik evaluasi seperti Akurasi, Presisi, *Recall*, dan *F-Measure*. *Confusion Matrix* menggambarkan jumlah prediksi yang benar dan salah yang dibuat oleh model dalam bentuk tabel matriks. Matriks ini memiliki dua dimensi yang merepresentasikan kelas positif dan negatif, baik untuk hasil prediksi maupun nilai aktualnya. Ada empat nilai yang dihasilkan di dalam tabel *confusion matrix*, di antaranya *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN) [14].

Tabel 1. Confusion Matrix

	Prediksi Positif	Prediksi Negatif
Aktual Positif	True Positive (TP)	False Negative (FN)
Aktual Negatif	False Positive (FP)	True Negative (TN)

Di mana :

- True Positive (TP) = Jumlah kasus di mana model memprediksi positif dan nilai aktualnya juga positif.
- False Positive (FP) = Jumlah kasus di mana model memprediksi positif, tetapi nilai aktualnya negatif.
- True Negative (TN) = Jumlah kasus di mana model memprediksi negatif dan nilai aktualnya juga negatif.
- False Negative (FN) = Jumlah kasus di mana model memprediksi negatif, tetapi nilai aktualnya positif.

Terdapat beberapa metrik evaluasi yang dapat dihitung berdasarkan *Confusion Matrix*. Metrik-metrik ini membantu dalam menilai seberapa baik model klasifikasi bekerja dalam memprediksi hasil. Berikut adalah rumus-rumus penting yang sering digunakan untuk mengevaluasi kinerja model.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$\text{Presisi} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F1-Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \tag{8}$$

### 3. Hasil

#### 3.1 Pemrosesan Data

Pada tahap data gathering, peneliti melakukan pengumpulan data ulasan aplikasi Seabank dari Platform Google Play Store dengan alamat <https://play.google.com/store/apps/details?id=ph.seabank.seabank>. Pengumpulan data dilakukan dengan menggunakan *platform Google Colaboratory* dan memanfaatkan *library python* yaitu *Google-Play-Scraper* untuk melakukan proses *scraping* data. Pengambilan data ulasan dilakukan pada 27 September 2024 dengan data ulasan yang diambil pada penelitian kali ini sebanyak 15.000 yang diambil mulai dari 27 Juli 2024 sampai 26 September 2024.

Data yang diperoleh melalui proses *scraping* selanjutnya masuk ke tahapan *preprocessing* dengan tujuan menjadikan data ulasan mudah di proses dan klasifikasi, diantaranya adalah dengan *Casefolding* yang mana akan mengubah semua huruf menjadi huruf kecil (*lowercase*), Menghapus tanda baca, angka, dan karakter khusus (*cleaning*), Mengubah *slang word* ke bahasa baku (*slang words normalizer*), menghapus semua kata hubung (*stopwords removal*), memecah satu kalimat menjadi per satuan kata (*tokenizing*), mengubah suatu kata ke kata dasarnya (*stemming*).

Tabel 2. Hasil Text Preprocessing

Sebelum	Sesudah
Aplikasi nya mantap jelas simpel mudah dipahami semoga aman nyaman dan menguntungkan sesama apalagi nilai buat deposito nya keuntungan yang stabil	["aplikasi", "mantap", "jelas", "simpler", "mudah", "paham", "aman", "nyaman", "untung", "apalagi", "nilai", "deposito", "untung", "stabil"]
Bagus simple n sangat mudah di fahami sistem ya..tp blm dapat mode pinjaman saja sih..selebih ya baik semua dan sukses selalu seabank	["bagus", "simple", "mudah", "faham", "sistem", "tapi", "mode", "pinjam", "ya", "baik", "semua", "sukses", "selalu", "seabank"]
Gak rekomendasi untuk di jadikan handalan menyimpan uang dan transaksi.. masak iya status pengiriman berhasil. Saldo udah terpotong. 30 menit lebih gak ada masuk saldo nya.. payah 🙄	["rekomendasi", "jadi", "andal", "simpan", "uang", "transaksi", "masak", "status", "kirim", "hasil", "saldo", "potong", "menit", "lebih", "masuk", "saldo", "buruk"]

Pada penelitian ini, karena data ulasan yang diambil dari *Google Play Store* sudah terdapat fitur *score* sehingga peneliti menggunakan fitur *score* untuk melakukan proses *labelling*. Pada kelas positif terdapat 13.880 data, negatif terdapat 824 data, dan netral terdapat 296 data.

*FastText Word Embedding* menggunakan representasi vektorial dengan dimensi 300 untuk setiap kata, proses ini bertujuan untuk mengubah kata-kata dalam teks menjadi representasi vektorial yang dapat digunakan untuk analisis lebih lanjut.

```
Proses Pembobotan untuk Kalimat: 'gratis transfer kemanapun keren'
Kata: 'gratis', Embedding: [ 0.0631 0.0379 -0.001 ] ... [ 0.0103 -0.0151 0.0372]
Kata: 'transfer', Embedding: [0.004 0.0247 0.0452] ... [-0.0458 -0.0011 0.0585]
Kata: 'kemanapun', Embedding: [ 0.0471 -0.065 -0.0136] ... [-0.0128 0.0373 0.0332]
Kata: 'keren', Embedding: [ 0.0173 -0.0641 -0.0006] ... [ 0.0024 -0.009 0.0474]

Rata-rata Embedding Kalimat: [ 0.032875 -0.016625 0.0075 ] ... [-0.011475 0.003025 0.044075]
```

**Gambar 2.** Hasil *Fasttext Word Embedding*

Pada Gambar 2, setiap kata dalam kalimat tersebut akan diubah menjadi vektor menggunakan model *Fast-Text*. Setelah mendapatkan *embedding* untuk masing-masing kata, langkah selanjutnya adalah menghitung rata-rata *embedding* untuk seluruh kalimat. Rata-rata ini memberikan representasi vektorial yang lebih komprehensif dari keseluruhan kalimat, dan dapat digunakan sebagai fitur dalam analisis lebih lanjut.

Karena data ulasan yang dikumpulkan tidak seimbang yang dapat dilihat pada Gambar 2, dari 15.000 data terdapat 13.880 positif, 824 negatif, dan 296 netral. Sehingga perlu dilakukan *oversampling*, peneliti menggunakan *Synthetic Minority Oversampling Technique* (SMOTE) pada penelitian ini. Pada Gambar 2, persebaran data setelah SMOTE sudah seimbang yang dimana positif, negatif, dan netral mempunyai 13.880 data, sehingga total semua dataset setelah dilakukan proses SMOTE berjumlah 41.640.

Hasil dataset setelah dilakukan SMOTE selanjutnya akan dibagi menjadi data *train* dan *data test*, dengan perbandingan 80:20. Dari total dataset yang berjumlah 41.640, dibagi menjadi 33.312 data *training*, dan 8.328 data *testing*.

## 3.2 Evaluasi Model

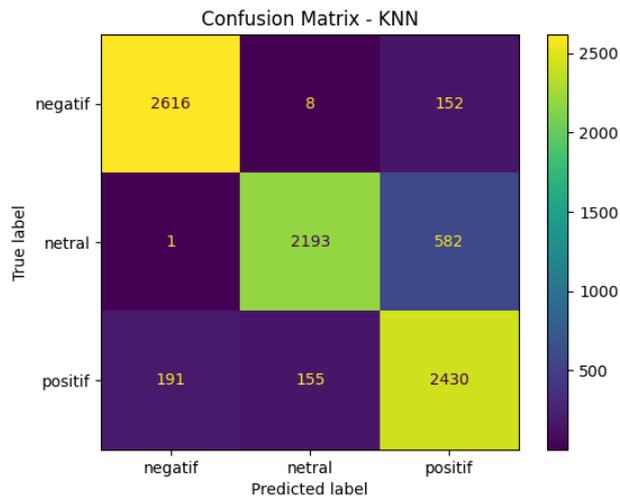
### 3.2.1 K-Nearest Neighbors

Dari hasil klasifikasi menggunakan algoritma KNN yang dilakukan pada dataset ulasan aplikasi Seabank algoritma KNN mencapai akurasi yang cukup baik yaitu sebesar 87%. Rincian hasil klasifikasi KNN dapat dilihat pada Tabel 3.

**Tabel 3.** Hasil klasifikasi KNN

	Precision	Recall	F1-Score	Support
<i>Negatif</i>	0.93	0.94	0.94	2776
<i>Netral</i>	0.93	0.79	0.85	2776
<i>Positif</i>	0.77	0.88	0.82	2776
<i>Accuracy</i>			0.87	8328
<i>Macro Avg</i>	0.88	0.87	0.87	8328
<i>Weighted Avg</i>	0.88	0.87	0.87	8328

Berikut visualisasi *confusion matrix* dari model KNN :



Gambar 3. Confusion Matrix – KNN

Gambar 3 menunjukkan *confusion matrix model* KNN dalam klasifikasi sentimen ulasan SeaBank. Model berhasil memprediksi dengan benar 2.616 data negatif, 2.193 data netral, dan 2.430 data positif. Namun, terdapat beberapa kesalahan prediksi, seperti 582 data netral diprediksi sebagai positif dan 191 data positif diprediksi sebagai negatif. Meskipun performa KNN cukup baik, model cenderung mengalami kesalahan pada kelas netral yang sering diprediksi sebagai positif, yang kemungkinan disebabkan oleh kemiripan fitur antar kategori.

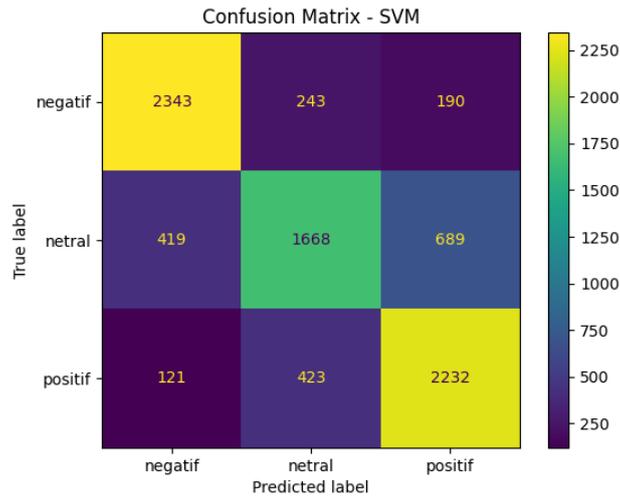
### 3.2.2 Support Vector Machine (SVM)

Dari hasil klasifikasi menggunakan algoritma SVM yang dilakukan pada dataset ulasan aplikasi Seabank algoritma SVM mencapai akurasi yang cukup baik yaitu sebesar 75%. Rincian hasil klasifikasi SVM dapat dilihat pada Tabel 4.

Tabel 4. Hasil klasifikasi SVM

	Precision	Recall	F1-Score	Support
<i>Negatif</i>	0.81	0.84	0.83	2776
<i>Netral</i>	0.71	0.6	0.65	2776
<i>Positif</i>	0.72	0.8	0.76	2776
<i>Accuracy</i>			0.75	8328
<i>Macro Avg</i>	0.75	0.75	0.75	8328
<i>Weighted Avg</i>	0.75	0.75	0.75	8328

Berikut visualisasi *confusion matrix* dari model SVM :



Gambar 4. Confusion Matrix – SVM

Gambar 4 menunjukkan *confusion matrix model* SVM dalam klasifikasi sentimen ulasan SeaBank. Model berhasil memprediksi dengan benar 2.343 data negatif, 1.668 data netral, dan 2.232 data positif. Namun, terdapat beberapa kesalahan prediksi, seperti 423 data positif diprediksi sebagai netral dan 689 data netral diprediksi sebagai positif. Dibandingkan dengan KNN, SVM menunjukkan performa yang lebih baik dalam memprediksi kategori negatif dan positif, tetapi masih memiliki kelemahan dalam membedakan kategori netral, yang sering diprediksi sebagai positif atau netral. Hal ini menunjukkan kemampuan SVM yang unggul dalam margin optimal tetapi tetap memiliki tantangan dalam menangani kelas dengan fitur yang mirip.

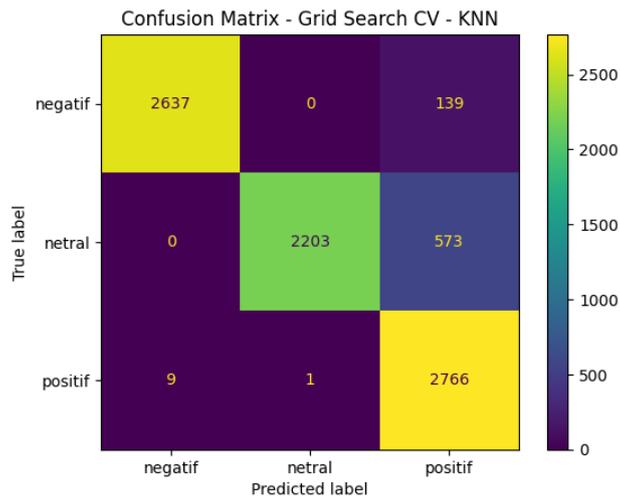
### 3.2.3 K-Nearest Neighbors (KNN) dengan Grid Search

Dari hasil klasifikasi menggunakan algoritma KNN yang di optimasi menggunakan *Grid Search* yang dilakukan pada dataset ulasan aplikasi Seabank, algoritma KNN yang di optimasi *Grid Search* mencapai akurasi yang baik yaitu sebesar 91%. Rincian hasil klasifikasi KNN dengan Grid Search dapat dilihat pada Tabel 5.

Tabel 5. Hasil klasifikasi KNN dengan *Grid Search*

<i>Best parameter: {metric: manhattan, n_neighbors: 3, weights: distance}</i>				
	Precision	Recall	F1-Score	Support
<i>Negatif</i>	1	0.95	0.97	2776
<i>Netral</i>	1	0.79	0.88	2776
<i>Positif</i>	0.8	1	0.88	2776
<i>Accuracy</i>			0.91	8328
<i>Macro Avg</i>	0.93	0.91	0.91	8328
<i>Weighted Avg</i>	0.93	0.91	0.91	8328

Berikut visualisasi *confusion matrix* dari model KNN dengan *Grid Search* :



Gambar 5. Confusion Matrix – KNN Grid Search

Gambar 5 menunjukkan *confusion matrix* KNN setelah optimasi *Grid Search CV*. Model memprediksi dengan benar 2.637 data negatif, 2.203 data netral, dan 2.766 data positif. Kesalahan prediksi berkurang signifikan, dengan 139 data negatif dan 573 data netral diprediksi sebagai positif. Optimasi ini meningkatkan akurasi KNN secara signifikan dibandingkan sebelumnya.

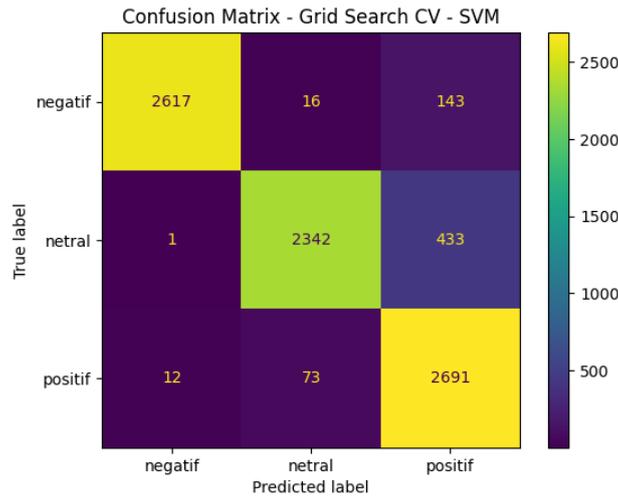
### 3.2.4 Support Vector Machine (SVM) dengan Grid Search

Dari hasil klasifikasi menggunakan algoritma SVM yang di optimasi menggunakan *Grid Search* yang dilakukan pada dataset ulasan aplikasi Seabank, algoritma SVM yang di optimasi *Grid Search* mencapai akurasi yang baik yaitu sebesar 92%. Rincian hasil klasifikasi SVM dengan *Grid Search* dapat dilihat pada Tabel 6.

Tabel 6. Hasil klasifikasi SVM dengan *Grid Search*

Best parameter: {C: 100, kernel: rbf }				
	Precision	Recall	F1-Score	Support
Negatif	1	0.94	0.97	2776
Netral	0.96	0.84	0.9	2776
Positif	0.82	0.97	0.89	2776
Accuracy			0.92	8328
Macro Avg	0.93	0.92	0.92	8328
Weighted Avg	0.93	0.92	0.92	8328

Berikut visualisasi *confusion matrix* dari model SVM dengan *Grid Search* :



Gambar 6. Confusion Matrix –SVMGrid Search

Gambar 6 menunjukkan *confusion matrix* SVM setelah optimasi *Grid Search* CV. Model memprediksi dengan benar 2.617 data negatif, 2.342 data netral, dan 2.691 data positif. Kesalahan prediksi menurun, dengan hanya 143 data negatif dan 433 data netral diprediksi sebagai positif. Optimasi *Grid Search* CV meningkatkan akurasi SVM secara signifikan, terutama dalam menangani kategori negatif dan positif.

### 3.2.5 Perbandingan Hasil Model

Pada hasil perbandingan metode klasifikasi akan dilakukan perbandingan hasil akurasi dari berbagai metode klasifikasi yang digunakan dalam penelitian ini, yaitu KNN dan SVM, baik sebelum maupun setelah dilakukan optimasi menggunakan *Grid Search*. Perbandingan ini bertujuan untuk melihat sejauh mana optimasi melalui *Grid Search* dapat meningkatkan performa model serta menentukan metode klasifikasi mana yang memberikan hasil terbaik dalam memprediksi data. Perbandingan akurasi untuk memperlihatkan perbedaan sebelum dilakukan optimasi, setelah dilakukan optimasi menggunakan *Grid Search*, dan perbandingan dengan penelitian sebelumnya ditampilkan pada Tabel 7.

Tabel 7. Evaluasi Model

Model	Akurasi	Presisi	Recall	F-1 Score
K-Nearest Neighbors (KNN)	87.00%	88.00%	87.00%	87.00%
Support Vector Mechine (SVM)	75.00%	75.00%	75.00%	75.00%
Grid Search + KNN	91.00%	93.00%	91.00%	91.00%
Grid Search + SVM	92.00%	93.00%	92.00%	92.00%

Tabel 8. Perbandingan penelitian terdahulu

Model	Sebelum Optimasi	Setelah Optimasi	Penelitian Terdahulu
K-Nearest Neighbors (KNN)	87.00 %	91.00%	83.92%
Support Vector Machine (SVM)	75.00%	92.00%	78.21%

Pada tabel 8, terlihat bahwa akurasi awal model KNN sebesar 87% lebih tinggi dibandingkan dengan hasil penelitian terdahulu yang dilakukan oleh Dany Pratmanto *et al.* yang mencapai 83.92% [15]. Hal yang sama berlaku untuk SVM, di mana akurasi awal sebesar 75% sedikit lebih rendah dibandingkan dengan penelitian sebelumnya yang dilakukan oleh Fathurahman Bei dan Sudin Saepudin yang mencapai 78.21% [8]. Setelah dilakukan optimasi menggunakan *Grid Search*, kedua model mengalami peningkatan akurasi, dengan KNN mencapai 91% dan SVM mencapai 92%. Peningkatan ini menunjukkan bahwa optimasi parameter dengan *Grid Search* tidak hanya meningkatkan performa model secara signifikan, tetapi juga melampaui hasil dari penelitian terdahulu.

#### 4. Pembahasan

Penelitian ini berhasil mengoptimalkan metode klasifikasi KNN dan SVM dalam analisis sentimen ulasan aplikasi SeaBank dengan menggunakan ekstraksi fitur *FastText Word Embedding* dan optimasi menggunakan *Grid Search*. Peningkatan akurasi model KNN cukup baik, yaitu sebesar 4%, dari yang sebelumnya 87% menjadi 91% setelah dioptimasi. Sementara itu, model SVM menunjukkan peningkatan yang jauh lebih signifikan, yaitu sebesar 17%, dari akurasi awal 75% menjadi 92% setelah dioptimasi dengan *Grid Search*.

Perbedaan signifikan dalam peningkatan kinerja antara KNN dan SVM dapat dijelaskan melalui karakteristik dasar kedua algoritma. SVM memiliki kemampuan untuk menemukan *hyperplane* optimal yang memisahkan kelas dengan margin maksimum. Dalam penelitian ini, *Grid Search* membantu SVM menemukan kombinasi parameter yang lebih tepat, seperti parameter *kernel* ('rbf') dan nilai regularisasi (C=100), sehingga meningkatkan performanya secara signifikan. Sebaliknya, KNN merupakan algoritma berbasis instance yang sangat bergantung pada distribusi data di ruang fitur. Meskipun *Grid Search* berhasil meningkatkan akurasi KNN melalui pemilihan jumlah tetangga (k) yang optimal, efek optimasi ini lebih terbatas karena KNN tidak memiliki mekanisme seperti margin maksimal yang dimiliki SVM. Dengan kata lain, SVM lebih responsif terhadap perubahan parameter *hyperplane* yang dihasilkan oleh *Grid Search* dibandingkan KNN. Selain itu, ekstraksi fitur menggunakan *FastText* juga berkontribusi pada hasil yang lebih baik untuk SVM. Kemampuan *FastText* untuk menangkap informasi sub-kata meningkatkan representasi data teks, yang secara langsung mendukung SVM dalam mendefinisikan *hyperplane* dengan presisi yang lebih tinggi. Sementara itu, KNN lebih rentan terhadap distribusi data yang tidak merata, sehingga peningkatan performanya tidak sejelas pada SVM meskipun menggunakan representasi fitur yang sama.

Dalam perbandingan dengan penelitian sebelumnya, hasil penelitian ini mendapatkan hasil akurasi yang lebih tinggi dikarenakan pemilihan ekstraksi fitur dan optimasi yang dilakukan. Studi ini memiliki dampak signifikan dalam konteks pemilihan ekstraksi fitur, algoritma, dan *optimizer* yang tepat dalam melakukan analisis sentimen. Namun, penelitian ini memiliki keterbatasan dalam klasifikasi, walaupun sudah dilakukan SMOTE dalam menangani *imbancing* data, namun hasilnya akan lebih baik jika persebaran datanya tidak terlalu jauh perbedaannya.

Penelitian selanjutnya dapat mempertimbangkan penggunaan metode klasifikasi tambahan seperti *Random Forest* atau *Naive Bayes* untuk dibandingkan dengan KNN dan SVM. Di sisi ekstraksi fitur, penggunaan metode pre-trained yang lebih canggih, seperti *BERT* atau *Word2Vec*, dapat dieksplorasi untuk melihat apakah lebih unggul dibandingkan *Fasttext*. Selain itu, optimasi model dapat diperluas dengan menggunakan teknik lain seperti *Bayesian Optimization* atau *Genetic Algorithm*, yang mungkin akan memberikan hasil lebih optimal dibandingkan *Grid Search*.

#### 5. Simpulan

Berdasarkan penelitian yang dilakukan, metode klasifikasi KNN dan SVM dengan ekstraksi fitur *FastText* yang dioptimasi menggunakan *Grid Search* menunjukkan kinerja yang sangat baik dalam analisis sentimen ulasan aplikasi SeaBank. Dari 15.000 ulasan yang dianalisis (13.880 positif, 824 negatif, 296 netral), model KNN awal memiliki akurasi 87% yang meningkat menjadi 91% setelah optimasi, sementara SVM menunjukkan peningkatan signifikan dari 75% menjadi 92% dengan parameter terbaik (C: 100, kernel: 'rbf'). Hasil ini mengonfirmasi efektivitas *Grid Search* dalam meningkatkan kinerja model klasifikasi untuk berbagai kelas sentimen. Peningkatan akurasi ini memiliki implikasi praktis yang signifikan, seperti membantu SeaBank memahami persepsi pelanggan, meningkatkan pengalaman pengguna, serta mendukung pengambilan keputusan strategis berbasis data untuk meningkatkan layanan. Pengembangan lebih lanjut dapat mencakup penggunaan metode *deep learning*, seperti LSTM atau BERT, untuk meningkatkan kemampuan analisis sentimen, serta pengayaan dataset dengan mengintegrasikan ulasan dari platform lain, seperti App Store atau media sosial, guna mendapatkan wawasan yang lebih luas dan mendalam.

#### Pustaka

- [1] S. Rahayu, Y. MZ, J. E. Bororing, and R. Hadiyat, "Implementasi metode k-nearest neighbor (k-nn) untuk analisis sentimen kepuasan pengguna aplikasi teknologi finansial flip," *Edumatic: Jurnal Pen-*

- didikan Informatika*, vol. 6, no. 1, pp. 98–106, Jun. 2022.
- [2] D. Oktavia and Y. R. Ramadahan, “Analisis sentimen terhadap penerapan sistem e-tilang pada media sosial twitter menggunakan algoritma support vector machine (svm),” *Media Online*, vol. 4, no. 1, pp. 407–417, 2023.
  - [3] F. Firmansyah *et al.*, “Comparing sentiment analysis of indonesian presidential election 2019 with support vector machine and k-nearest neighbor algorithm,” in *6th International Conference on Computing, Engineering, and Design, ICCED 2020*. Institute of Electrical and Electronics Engineers Inc., Oct. 2020.
  - [4] A. Baita and N. Cahyono, “Analisis sentimen mengenai vaksin sinovac menggunakan algoritma support vector machine (svm) dan k-nearest neighbor (knn),” 2023.
  - [5] M. A. Harriz and H. Setiyowati, “Komparasi algoritma decision tree dan knn dalam mengklasifikasi daerah berdasarkan produksi listrik,” *JIKO (Jurnal Informatika dan Komputer)*, vol. 7, no. 2, p. 167, 2023.
  - [6] F. Putrawansyah, “Penerapan metode support vector machine terhadap klasifikasi jenis jambu biji,” *JIKO (Jurnal Informatika dan Komputer)*, vol. 8, no. 1, p. 193, 2024.
  - [7] A. Nurdin, B. Anggo, S. Aji, A. Bustamin, and Z. Abidin, “Perbandingan kinerja word embedding word2vec, glove, dan fasttext pada klasifikasi teks,” *Jurnal TEKNOKOMPAK*, vol. 14, no. 2, p. 74, 2020.
  - [8] F. Bei and S. Saepudin, “Analisis sentimen aplikasi tiket online di play store menggunakan metode support vector machine (svm),” 2021.
  - [9] F. H. Rachman and I. Imamah, “Pendekatan Data Science untuk Mengukur Empati Masyarakat terhadap Pandemi Menggunakan Analisis Sentimen dan Seleksi Fitur,” *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 8, no. 3, p. 492, Dec. 2022. [Online]. Available: <https://jurnal.untan.ac.id/index.php/jepin/article/view/56655>
  - [10] P. Mulinka, S. Sahoo, C. Kalalas, and P. H. J. Nardelli, “Optimizing a Digital Twin for Fault Diagnosis in Grid Connected Inverters - A Bayesian Approach,” in *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*. Detroit, MI, USA: IEEE, Oct. 2022, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/document/9947986/>
  - [11] L. B. Ilmawan and M. A. Mude, “Perbandingan Metode Klasifikasi Support Vector Machine dan Naïve Bayes untuk Analisis Sentimen pada Ulasan Tekstual di Google Play Store,” *ILKOM Jurnal Ilmiah*, vol. 12, no. 2, pp. 154–161, Aug. 2020. [Online]. Available: <http://jurnal.fikom.umi.ac.id/index.php/ILKOM/article/view/597>
  - [12] R. Harun, K. C. Pelangi, and Y. Lasena, “Penerapan data mining untuk menentukan potensi hujan harian dengan menggunakan algoritma k nearest neighbor (knn),” 2020. [Online]. Available: <http://e-journal.stmiklombok.ac.id/index.php/misi>
  - [13] Y. N. Fuadah, M. A. Pramudito, and K. M. Lim, “An optimal approach for heart sound classification using grid search in hyperparameter optimization of machine learning,” *Bioengineering*, vol. 10, no. 1, 2023.
  - [14] A. Sujjada, J. N. Novianti, and I. G. T. Isa, “Analisis sentimen terhadap review bank digital pada google play store menggunakan metode support vector machine (svm),” 2023. [Online]. Available: <https://rekayasa.nusaputra.ac.id/index>
  - [15] D. Pratmanto, F. Fandi, D. Imaniawan, and C. Author, “Analisis sentimen terhadap aplikasi canva menggunakan algoritma naive bayes dan k-nearest neighbors,” 2023. [Online]. Available: <http://jurnal.bsi.ac.id/index.php/co-science>