STORAGE – Jurnal Ilmiah Teknik dan Ilmu Komputer, Vol. 4 No. 2, Mei, 2025, 69 - 81
e-ISSN: 2828-5344 | DOI: 10.55123

ANALISIS SENTIMEN ULASAN APLIKASI WONDR BY BNI MENGGUNAKAN ALGORITMA SVM DENGAN OPTIMASI KERNEL TRICK

[1)]Ikbal Bazar, [2)]Farid Wajidi, [3)]A. Amirul Asnan Cirua

[1,2,3)](Teknik Informatika, Fakultas Teknik, Universitas Sulawesi Barat)
[1)]ikbalbazarpc@gmail.com, [2)]faridwajidi@unsulbar.ac.id, [3)]amirulasnancirua@unsulbar.ac.id

INFO ARTIKEL

Riwayat Artikel :
Diterima : 7 Mei 2025
Disetujui : 27 Mei 2025

Kata Kunci :
Analisis Sentimen, SVM, Kernel Trick, TF-IDF, Wondr by BNI

ABSTRAK

Transformasi digital dalam sektor perbankan Indonesia mendorong peluncuran aplikasi *Wondr by* BNI pada Juli 2024 sebagai pengganti BNI *Mobile Banking*. Meskipun jumlah pengguna aktifnya telah melampaui 5,3 juta hingga akhir 2024, ulasan di Google Play Store menunjukkan berbagai keluhan teknis, seperti kesulitan login, kegagalan transaksi, dan proses verifikasi yang rumit. Penelitian ini bertujuan untuk menganalisis sentimen pengguna terhadap aplikasi *Wondr* dengan menerapkan algoritma *Support Vector Machin*e (SVM) dan mengoptimasi performanya menggunakan empat jenis kernel: *Linear*, RBF, *Polynomial*, dan *Sigmoid*. Sebanyak 2.000 ulasan dikumpulkan melalui teknik scraping dari kategori *most relevant* di Google Play Store. Setelah melalui proses *preprocessing* dan pelabelan otomatis menggunakan model Multilingual BERT, sebanyak 1.719 data yang berkategori positif dan negatif digunakan untuk proses klasifikasi. Ekstraksi fitur dilakukan dengan metode Term *Frequency–Inverse Document Frequency* (TF-IDF). Evaluasi performa menggunakan metrik *F1-score* menunjukkan bahwa kernel *Sigmoid* dengan parameter $C = 3$ dan $\gamma = 0{,}5$ menghasilkan performa terbaik dengan nilai *Accuracy* dan *F1-score* sebesar 0,922. Analisis frekuensi kata menunjukkan bahwa kata "gagal" dan "verifikasi" mendominasi sentimen negatif, sementara "mudah" dan "cepat" banyak muncul dalam sentimen positif. Temuan ini menunjukkan bahwa kombinasi pelabelan otomatis berbasis BERT dan klasifikasi SVM dengan optimasi kernel mampu menghasilkan model analisis sentimen yang akurat dan stabil untuk mengevaluasi layanan aplikasi perbankan digital.

ARTICLE INFO

Article History :
Received : Mei 7, 2025
Accepted : Mei 27, 2025

Keywords*:*
Sentiment Analysis, SVM, Kernel Trick, TF-IDF, Wondr by BNI

ABSTRACT

The digital transformation of Indonesia's banking sector prompted the launch of Wondr by BNI in July 2024 as a replacement for BNI Mobile Banking. Although the application had surpassed 5.3 million active users by the end of 2024, user reviews on the Google Play Store revealed. recurring technical issues, including login failures, transaction errors, and complex verification processes. This study aims to analyze user sentiment toward the Wondr application using the Support Vector Machine (SVM) algorithm, optimized through the evaluation of four kernel types: Linear, RBF, Polynomial, and Sigmoid. A total of 2,000 user reviews were collected through web scraping from the "most relevant" category on the Google Play Store. After undergoing

preprocessing and automatic sentiment labeling using the Multilingual BERT model, 1,719 reviews labeled as either positive or negative were retained for classification. Feature extraction was conducted using the Term Frequency–Inverse Document Frequency (TF-IDF) method. Performance evaluation based on the F1-score indicated that the Sigmoid kernel, with parameters C = 3 and γ = 0.5, achieved the best results, with an accuracy and F1-score of 0.922. Keyword frequency analysis revealed that the terms "failed" and "verification" were dominant in negative reviews, while "easy" and "fast" appeared frequently in positive reviews. These findings demonstrate that the combination of BERT-based automatic labeling and kernel-optimized SVM classification provides an effective and stable approach for assessing user sentiment toward digital banking applications.

1. PENDAHULUAN

Transformasi digital dalam industri perbankan berkembang pesat secara global, didorong oleh teknologi seperti *artificial intelligence (AI)*, *machine learning*, dan *blockchain*, serta perubahan perilaku konsumen pasca pandemi COVID-19 (Judijanto *et al.*, 2024). Di Indonesia, transaksi perbankan digital juga mengalami peningkatan signifikan. Data Bank Indonesia (2024) mencatat bahwa pada September 2024, nilai transaksi *digital banking* mencapai Rp7.492,93 triliun, tumbuh 54,89% dibandingkan periode yang sama tahun sebelumnya (*year-on-year*) (Fharyana Otang, 2024). Beberapa aplikasi *mobile banking* mengalami lonjakan jumlah pengguna dalam beberapa tahun terakhir. Salah satunya adalah *Wondr* by BNI, layanan perbankan digital pengganti BNI Mobile Banking yang baru diluncurkan pada Juli 2024, Hingga akhir Desember 2024, aplikasi ini telah digunakan oleh lebih dari 5,3 juta pengguna aktif dan menyediakan fitur-fitur seperti transfer, pembayaran, serta investasi. Meningkatnya popularitas *mobile banking* juga diiringi dengan berbagai tantangan dalam pengalaman pengguna, sebagaimana terlihat dari ulasan di Google Play Store. Keluhan yang sering muncul meliputi kesulitan login, proses verifikasi yang kompleks, bug dalam aplikasi, serta kegagalan transaksi meskipun saldo terpotong (Munandar *et al.*, 2024). Ulasan pengguna ini dapat menjadi sumber informasi berharga bagi bank, tetapi analisis manual terhadap ribuan ulasan tidak efisien. Oleh karena itu, pendekatan berbasis *machine learning* diperlukan untuk mengotomatisasi evaluasi sentimen secara objektif dan sistematis

Analisis sentimen adalah proses mengidentifikasi pola dalam teks yang mencerminkan opini atau sentimen tertentu. Teknik ini menggunakan berbagai pendekatan, termasuk metode statistik, linguistik, dan pembelajaran mesin, untuk mengklasifikasikan teks ke dalam kategori seperti positif, negatif, atau netral (Nuraeni Herlinawati *et al.*, 2020). Dalam konteks pengembangan bisnis, *analisis sentimen* dapat digunakan untuk menilai perspektif konsumen terhadap suatu produk atau layanan. Melalui analisis ini, perusahaan dapat menarik kesimpulan mengenai strategi pengembangan produk di masa depan (Wankhade, Rao and Kulkarni, 2022). Proses *analisis sentimen* dapat dilakukan dengan pendekatan *machine learning*, yang merupakan cabang dari *artificial intelligence (AI)* dan berfokus pada pengembangan sistem agar mampu belajar secara mandiri tanpa perlu pemrograman ulang. Dalam praktiknya, *machine learning* memerlukan data pelatihan (*training data*) sebagai referensi utama untuk mempelajari pola sebelum digunakan dalam proses prediksi atau klasifikasi (Chazar, Erawan and Widhiaputra, 2020).

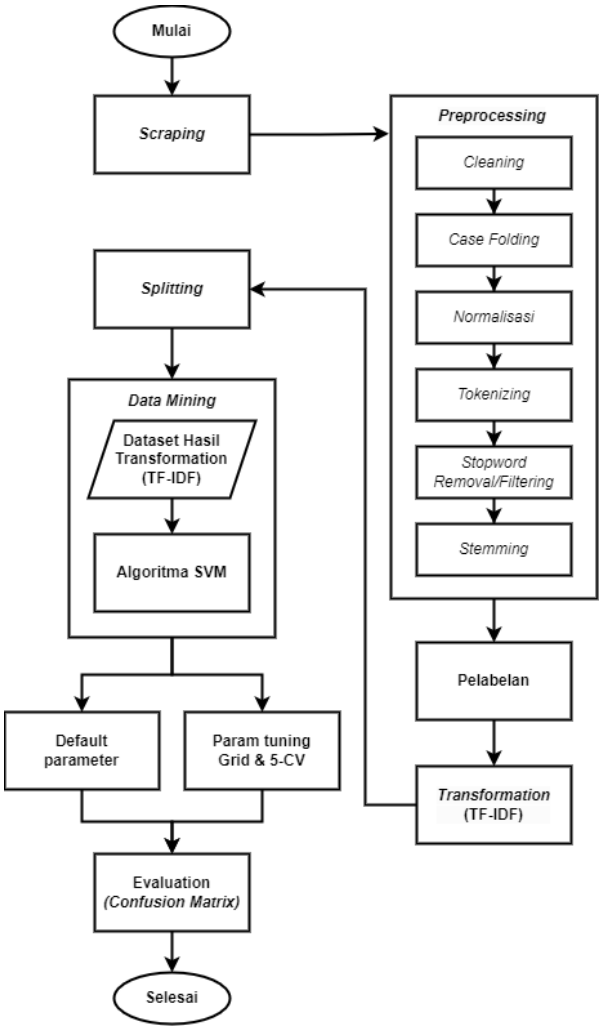
Beberapa algoritma yang sering digunakan dalam *machine learning* untuk keperluan klasifikasi dan regresi antara lain *Naïve Bayes* (NB), *Support Vector Machine* (SVM), serta *Maximum Entropy* (ME) (Isnain *et al.*, 2021). Menurut penelitian yang dilakukan oleh Pooja dan Bhalla (Pooja and Bhalla, 2022), terdapat beberapa algoritma lain yang sering digunakan

dalam tugas klasifikasi, di antaranya *Decision Tree* (DT), *K-Nearest Neighbors* (KNN), *Random Forest* (RF), *Artificial Neural Network* (ANN), serta *Linear Regression* (LR). Studi yang dilakukan oleh Rabbani (Rabbani *et al.*, 2023) berfokus pada perbandingan penggunaan kernel dalam algoritma *SVM* untuk *analisis sentimen*. Dalam penelitian tersebut, tiga jenis kernel yang diuji adalah *linear* kernel, *polynomial kernel*, dan RBF kernel. Hasilnya menunjukkan bahwa RBF kernel memberikan performa terbaik dibandingkan kernel lainnya. Sementara itu, penelitian yang dilakukan oleh Syahputra H (Syahputra, 2021) melakukan penelitian tentang analisis sentimen komunitas pada platform *online store* dengan menerapkan *Support Vector Machine* (SVM) menggunakan *linear kernel*, RBF *kernel*, dan *Sigmoid kernel*. Hasil eksperimen menunjukkan bahwa *Sigmoid* kernel memberikan *Accuracy* tertinggi, yakni 82%.

Beberapa studi sebelumnya telah menerapkan algoritma Support Vector Machine (SVM) dalam analisis sentimen ulasan aplikasi digital di Google Play Store. Idris dan Mussalimun (2024) menganalisis ulasan aplikasi *fintech Igrow* menggunakan kernel linear dan RBF, dengan *Accuracy* serupa sebesar 77%. Namun, penelitian ini belum mengeksplorasi tuning parameter (*C*, *gamma*) maupun tahap praproses secara rinci. Ranjan dan Mishra (2020) menggunakan SVM dan *TF-IDF* untuk mengklasifikasikan ulasan aplikasi umum, mencapai *Accuracy* 93,37% dan F1-score 0,88, tetapi tidak membahas variasi kernel maupun strategi optimasi model. Sementara itu, studi oleh Che Mohd Safawi dan Shafie (2023) menekankan efektivitas *TF-IDF* dalam klasifikasi sentimen ulasan aplikasi Shopee, namun tidak mengelaborasi pemilihan kernel SVM atau tuning parameter.

Melihat keterbatasan pada studi-studi tersebut, penelitian ini mengusulkan pendekatan yang lebih menyeluruh dengan membandingkan performa empat kernel utama SVM *(Linear*, RBF, *Polynomial*, dan *Sigmoid)*, disertai optimasi parameter dan pemrosesan fitur menggunakan TF-IDF. Pendekatan ini diharapkan dapat memberikan kontribusi terhadap *Accuracy* dan efektivitas analisis sentimen ulasan aplikasi digital.

2. METODE



Gambar 1. Alur penelitian

2.1 *Scraping*

Scraping merupakan teknik untuk mengumpulkan atau mengekstrak informasi dari berbagai sumber data. Data yang diperoleh dapat berupa teks, gambar, maupun video, dan umumnya dikonversi ke dalam format lain sesuai kebutuhan. Proses *scraping* dapat dilakukan dengan menulis rangkaian kode program atau menggunakan alat khusus yang dirancang untuk mengekstrak data dari internet (Sholihah, Diyase and Puspanigrum, 2024).

2.2 *Preprocessing*

Preprocessing adalah tahap persiapan data sebelum digunakan dalam *text mining*, sehingga dataset dibersihkan agar siap dianalisis (Kurniawan and Apriliani, 2020). Tahapan

preprocessing yang dilakukan dalam penelitian ini meliputi:

a. ***Cleaning:*** Menghapus atribut yang tidak diperlukan, seperti URL, *HTML tags*, emoji, simbol, dan angka (Darwis *et al.*, 2020).

b. ***Case Folding*****:** Mengubah seluruh huruf dalam teks menjadi huruf kecil agar konsistensi data tetap terjaga (Işik and Dağ, 2020).

c. ***Normalisasi:*** Mengubah kata tidak baku atau slang menjadi bentuk baku agar dapat diproses lebih lanjut secara akurat (Fransiska and Irham Gufroni, 2020).

d. ***Tokenizing:*** Memecah teks menjadi kata-kata yang lebih kecil agar dapat dianalisis lebih lanjut (Işik and Dağ, 2020).

e. ***Stopword Removal:*** Menghapus kata-kata yang dianggap tidak memiliki pengaruh signifikan terhadap makna dalam sebuah kalimat (Kulkarni and Mhaske, 2020).

f. ***Stemming:*** Mengubah kata berimbuhan menjadi bentuk dasar untuk menyederhanakan teks sebelum diproses lebih lanjut (Işik and Dağ, 2020).

2.3 Pelabelan

Pelabelan data dilakukan secara otomatis menggunakan model *nlptown/bert-base-multilingual-uncased-sentiment*, yang menghasilkan klasifikasi sentimen dalam skala bintang (1–5). Dalam penelitian ini, label '1 star' dan '2 stars' dikategorikan sebagai sentimen negatif, sedangkan '4 stars' dan '5 stars' dikategorikan sebagai sentimen positif.

Ulasan dengan '3 stars' tidak digunakan karena dianggap netral. Berbeda dengan pendekatan end-to-end BERT seperti pada Singgalen (2025) penelitian ini hanya memanfaatkan model BERT untuk pelabelan otomatis, sementara ekstraksi fitur untuk klasifikasi menggunakan metode TF-IDF. Hal ini dipilih untuk menjaga efisiensi dan interpretabilitas dalam proses klasifikasi menggunakan algoritma SVM.

2.4 TF-IDF

Pembobotan Term Frequency-Inverse Document Frequency (TF-IDF) diperoleh dari dua komponen utama, yaitu Term Frequency (TF), yang menghitung jumlah kemunculan kata dalam suatu dokumen, dan Inverse Document Frequency (IDF), yang menunjukkan seberapa jarang kata tersebut muncul dalam keseluruhan dokumen. Sebuah kata akan memiliki bobot tinggi apabila sering muncul dalam satu dokumen, tetapi jarang ditemukan pada dokumen lain. Dalam penelitian ini, perhitungan dilakukan menggunakan TfidfVectorizer dari pustaka Scikit-learn, yang secara default menggunakan frekuensi mentah untuk TF dan IDF. Rumus perhitungan TF-IDF adalah sebagai berikut:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \quad (1)$$

a. *Term Frequency (TF)*

Menghitung TF dihitung sebagai jumlah kemunculan kata dalam dokumen tanpa normalisasi atau pembobotan tambahan:

$$TF(t,d) = ft,d \quad (2)$$

Keterangan:
ft,d: Jumlah kemunculan term t dalam dokumen d

b. *Inverse Document Frequency (IDF)*

IDF dihitung dengan formula logaritmik yang telah disesuaikan untuk menghindari pembagian nol dan memastikan hasil tetap positif:

$$IDF(t) = \log\left(\frac{1+N}{1+df\ (t)\_}\right) + 1 \quad (3)$$

Keterangan:
N: Total jumlah dokumen
$df(t)$: Jumlah kemunculan suatu term/kata dalam dokumen
log : Logaritma natural (In)

2.5 *Splitting* Data

Dataset dibagi menjadi dua bagian, yaitu data pelatihan (*training data*) dan data pengujian (*testing data*). Data pelatihan digunakan untuk melatih algoritma agar dapat mengenali pola dalam dataset dan membangun model. Model yang telah dibentuk kemudian diuji menggunakan data pengujian guna mengevaluasi performa serta tingkat *Accuracy* prediksi.

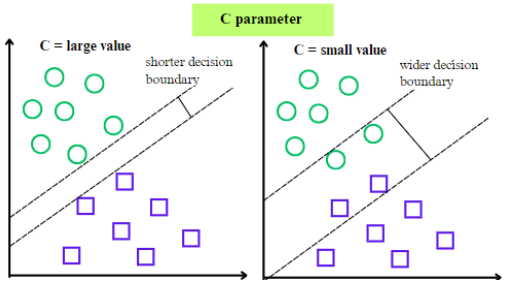
2.6 Pembuatan Model

Model dibangun menggunakan *training data* yang telah melewati tahap *preprocessing* dan perhitungan TF-IDF. Parameter yang digunakan disesuaikan dengan jenis kernel yang dipilih untuk mengoptimalkan kinerja model dalam klasifikasi sentimen (Rahardi, 2022). Dalam proses pengembangan model, pemilihan parameter dilakukan secara sistematis agar hasil klasifikasi lebih optimal (Fremmuzar and Baita, 2023).

2.6.1 *Hyperparameter Tuning* SVM

Dalam penelitian ini, terdapat beberapa *hyperparameter* yang akan diuji, di antaranya *Cost* (C), *gamma* (γ), dan *degree* (d) (Rusman, 2023)

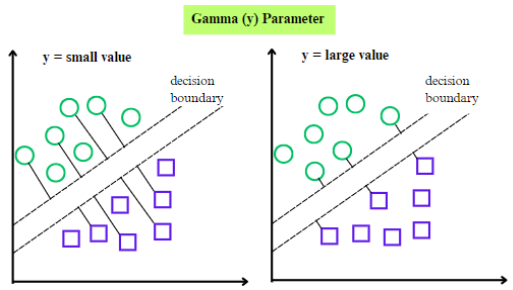
a. **Parameter *Cost*** terdapat pada semua jenis kernel dalam SVM. Nilai *Cost* yang tinggi mempersempit *decision boundary*, sehingga mengurangi kemampuan model dalam melakukan generalisasi dan berisiko menyebabkan *overfitting*. Pengaruh nilai *Cost* terhadap klasifikasi dapat dilihat pada Gambar 2



(a) (b)

Gambar 2. (a) Hasil klasifikasi dengan nilai *cost* yang rendah (b) Hasil klasifikasi dengan nilai *cost* yang tinggi

b. **Parameter *gamma* (γ)** Menentukan seberapa besar pengaruh suatu titik data terhadap proses pelatihan model. Dengan nilai *gamma* yang tinggi, hanya titik-titik yang berada dekat dengan *hyperplane* yang memiliki pengaruh signifikan. Sebaliknya, *gamma* yang rendah memungkinkan titik-titik yang lebih jauh dari hyperplane tetap mempengaruhi proses klasifikasi Visualisasi dampak nilai *gamma* terhadap klasifikasi dapat dilihat pada Gambar 3.



(a) (b)

Gambar 3. (a) Hasil klasifikasi dengan nilai *gamma* yang rendah (b) Hasil klasifikasi dengan nilai *gamma* yang tinggi

2.7 *Grid Search*

Grid Search merupakan salah satu algoritma yang digunakan untuk menentukan parameter optimal dalam suatu model. Algoritma ini beroperasi dengan menguji berbagai kombinasi parameter yang telah ditetapkan oleh pengguna. Metode ini menyusun hasil kombinasi parameter dalam sebuah *grid* dan memilih kombinasi terbaik berdasarkan nilai galat terkecil (Fajri and Primajaya, 2023).

2.8 Evaluasi Model

Langkah berikutnya adalah menilai performa model pada Setiap kernel dievaluasi menggunakan confusion matrix. Melalui *confusion matrix*, kinerja model dapat dianalisis secara menyeluruh berdasarkan jumlah prediksi yang benar dan salah untuk setiap kelas, berbagai matrik evaluasi seperti *accuracy*, *precision*, *recall*, dan F1-score dapat dihitung guna mengukur efektivitas model (Kavabilla, Widiharih and Warsito, 2023).

3. HASIL DAN PEMBAHASAN

Data penelitian dikumpulkan melalui scraping pada awal Maret 2025 menggunakan library *google-play-scraper* di Python. Ulasan berjumlah 2.000 dengan kategori *most relevant* dan berbahasa Indonesia, diambil dari Google Play Store. Rentang waktu ulasan dari 18 September 2024 hingga Maret 2025. Versi aplikasi yang dominan adalah 1.3.0

3.1 *Preprocessing*

Setelah tahap *scraping*, langkah berikutnya adalah p*reprocessing* data. Analisis sentimen

dilakukan terhadap ulasan pengguna aplikasi *Wondr by* BNI. Berikut adalah tahapan *preprocessing* data yang dilakukan.

a. Hasil *Cleaning*

Tahap yang menghilangkan atribut yang tidak digunakan seperti emoji, tanda baca dan karakter kosong ditunjukkan oleh Tabel 1.

Tabel 1. Hasil *cleaning*

Sebelum	Sesudah
Makin hari makin parah aplikasi iniAsli super Nyesel gua pakai BNI. Mau pindah aja. Ada yang transfer katanya sudah berhasil akan tetapi gak keluar kertas print di ATM	Makin hari makin parah aplikasi iniAsli super Nyesel gua pakai BNI Mau pindah aja Ada yang transfer katanya sudah berhasil akan tetapi gak keluar kertas print di ATM

a. Hasil *Case Folding*

Tahap mengubah seluruh huruf dalam teks menjadi huruf kecil. Hasil dari proses *case folding* ditampilkan pada Tabel 2.

Tabel 2. Hasil *case folding*

Sebelum	Sesudah
Makin hari makin parah aplikasi iniAsli super Nyesel gua pakai BNI Mau pindah aja Ada yang transfer katanya sudah berhasil akan tetapi gak keluar kertas print di ATM	makin hari makin parah aplikasi iniasli super nyesel gua pakai bni mau pindah aja ada yang transfer katanya sudah berhasil akan tetapi gak keluar kertas print di atm

a. Hasil *Normalisasi*

Kata-kata tidak baku dikonversi ke dalam bentuk baku, seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Hasil *normalisasi*

Sebelum	Sesudah
makin hari makin parah apk ini asli super nyesel gua pakai bni mau pindah aja ada yg transfer katanya sudah berhasil akan tetapi gak keluar kertas print di atm	makin hari makin parah aplikasi ini asli super nyesel gua pakai bni mau pindah aja ada yang transfer katanya sudah berhasil akan tetapi tidak keluar kertas print di atm

d. Hasil *Tokenizing*

Pada proses *tokenizing*, spasi digunakan sebagai pemisah antar token. Hasil *transformasi* kalimat menjadi token-token ditampilkan pada Tabel 4.

Tabel 4. Hasil *tokenizing*

Sebelum	Sesudah
makin hari makin parah aplikasi iniasli super nyesel gua pakai bni mau pindah aja ada yang transfer katanya sudah berhasil akan tetapi tidak keluar kertas print di atm	['makin', 'hari', 'makin', 'parah', 'aplikasi', 'iniasli', 'super', 'menyesal', 'gua', 'pakai', 'bni', 'mau', 'pindah', 'saja', 'ada', 'yang', 'transfer', 'katanya', 'sudah', 'berhasil', 'akan', 'tetapi', 'tidak', 'keluar', 'kertas', 'print', 'di', 'atm']

e. Hasil *Stopword Removal*

Tahap ini menghapus kata-kata yang dianggap tidak memiliki pengaruh signifikan. sebagaimana ditampilkan pada Tabel 5.

Tabel 5. Hasil *stopword removal*

Sebelum	Sesudah
['makin', 'hari', 'makin', 'parah', 'aplikasi', 'iniasli', 'super', 'menyesal', 'gua', 'pakai', 'bni', 'mau', 'pindah', 'saja', 'ada', 'yang', 'transfer', 'katanya', 'sudah', 'berhasil', 'akan', 'tetapi', 'tidak', 'keluar', 'kertas', 'print', 'di', 'atm']	['parah', 'aplikasi', 'iniasli', 'super', 'menyesal', 'gua', 'pakai', 'bni', 'pindah', 'transfer', 'berhasil', 'kertas', 'print', 'atm']

f. Hasil *Stemming*

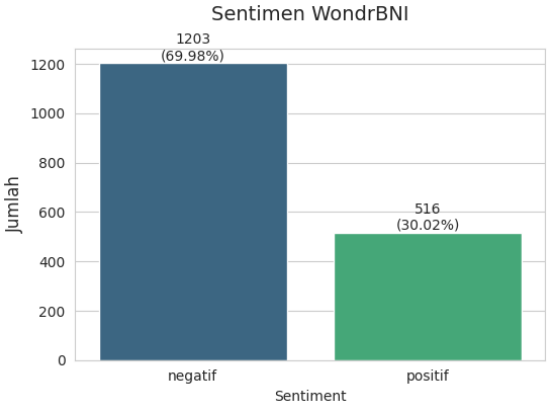
Tahap ini bertujuan untuk mengubah kata ke dalam bentuk dasarnya menggunakan *library* Sastrawi, seperti yang ditampilkan pada Tabel 6

Tabel 6. Hasil *stemming*

Sebelum	Sesudah
['parah', 'aplikasi', 'iniasli', 'super', 'menyesal', 'gua', 'pakai', 'bni', 'pindah', 'transfer', 'berhasil', 'kertas', 'print', 'atm']	parah aplikasi iniasli super sesal gua pakai bni pindah transfer hasil kertas print atm

3.2 Pelabelan

Dari total 2.000 ulasan yang dikumpulkan, proses pelabelan otomatis dilakukan menggunakan model *nlptown/bert-base-multilingual-uncased-sentiment*, yang menghasilkan label dalam skala bintang (1–5). Dalam penelitian ini, ulasan dengan label '1 star' dan '2 stars' dikategorikan sebagai sentimen negatif, sementara '4 stars' dan '5 stars' dikategorikan sebagai sentimen positif. Ulasan dengan label '3 stars' tidak digunakan karena dianggap bersifat netral dan ambiguitasnya dapat mempengaruhi *Accuracy* model klasifikasi biner. Gambar 4 menunjukkan distribusi jumlah data pada masing-masing kategori sentimen setelah proses pelabelan selesai dilakukan.

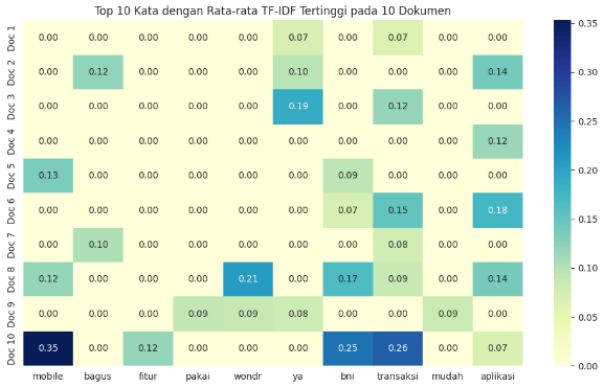


Gambar 4. Hasil Distribusi tiap kategori data

Setelah proses pelabelan dan penyaringan, sebanyak 1.719 data (positif dan negatif) digunakan untuk proses klasifikasi. Adapun 281 data berlabel netral (3 stars) dihilangkan guna menjaga kejelasan label dan meningkatkan konsistensi dalam pelatihan model klasifikasi biner.

3.3 TF-IDF

Tahapan ini bertujuan untuk mengubah data teks menjadi bentuk numerik yang dapat diproses oleh algoritma klasifikasi. Metode yang digunakan adalah TF-IDF (*Term Frequency – Inverse Document Frequency*), yang memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam sebuah ulasan dan kelangkaannya di seluruh *dataset*



Gambar 5. Hasil TF-IDF

Gambar 5 menampilkan *Heatmap* 10 Kata dengan Rata-rata Nilai TF-IDF Tertinggi pada 10 Dokumen Ulasan Pengguna Aplikasi *Wondr*. Setiap sel dalam matriks menunjukkan bobot TF-IDF untuk sebuah kata tertentu dalam satu dokumen. Warna yang lebih gelap menunjukkan bobot TF-IDF yang lebih tinggi, yang menunjukkan pentingnya kata tersebut dalam dokumen bersangkutan

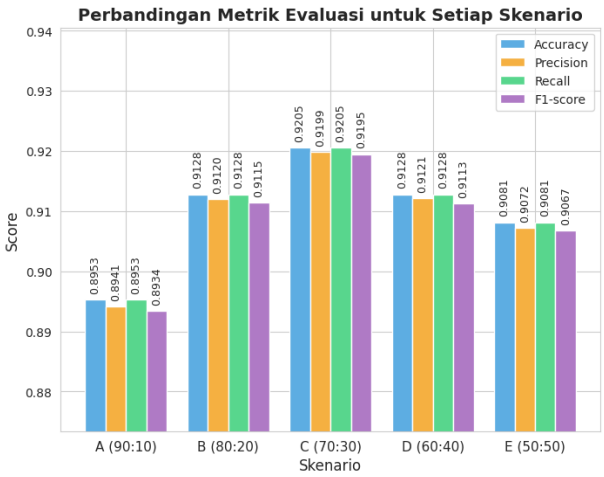
3.4 Implementasi SVM

Dalam penelitian ini, implementasi awal *Support Vector Machine* (SVM) dilakukan dengan berbagai skenario pembagian data *training* dan *testing* untuk menentukan konfigurasi dengan nilai *F1-score* optimal. *F1-score* dipilih sebagai matrik evaluasi karena distribusi sentimen dalam dataset tidak seimbang antara kategori positif dan negatif. Evaluasi proporsi pembagian data dilakukan berdasarkan perbandingan nilai *F1-score* yang diperoleh, seperti ditampilkan pada Tabel 7

Tabel 7. Pembagian Jumlah Dataset

Skenario	Jumlah Split
A	90% *Training* 10% *Testing*
B	80% *Training* 20% *Testing*
C	70% *Training* 30% *Testing*
D	60% *Training* 40% *Testing*
E	50% *Training* 50% *Testing*

Pada tahap awal, algoritma *Support Vector Machine* (SVM) diterapkan menggunakan kernel dan parameter *default* untuk mengevaluasi performa model berdasarkan *F1-score* dalam berbagai skenario pembagian dataset. Hasil pengujian ditampilkan pada Gambar 6.



Gambar 6. Perbandingan Performa Dataset

Dari hasil Gambar 6 tersebut, skenario C dengan pembagian data 70% untuk pelatihan dan 30% untuk pengujian menunjukkan nilai F1-score tertinggi, yaitu 0,9195. Oleh karena itu, skenario C dipilih sebagai konfigurasi terbaik. Untuk mengatasi ketidakseimbangan kelas dalam dataset, dilakukan *stratified oversampling* pada data pelatihan agar proporsi sentimen positif dan negatif tetap seimbang. Selanjutnya, optimasi *kernel trick* dilakukan dengan menyesuaikan parameter kernel untuk meningkatkan nilai *F1-score.*

3.5 Optimasi Kernel

a. Optimasi Kernel *Linear*

Pengujian algoritma *Support Vector Machine* (SVM) dengan *linear kernel* dilakukan dengan menguji berbagai nilai parameter *Cost (C)* untuk mendapatkan nilai *F1-score* terbaik. Parameter *C* yang diuji mencakup nilai *0.5*, *1*, *2*, *3*, dan *4*. Rentang nilai ini dipilih berdasarkan pertimbangan sebagai berikut:

1. Mengacu pada penelitian terdahulu yang menggunakan rentang serupa dalam pengujian performa *kernel* SVM terhadap data teks, seperti studi oleh (Fremmuzar and Baita, 2023) dan (Rusman, 2023).
2. Rentang *C* dari *0.5* hingga *4.0* memberikan variasi yang cukup untuk melihat tren performa model, tanpa menimbulkan risiko *underfitting* (jika *C* terlalu kecil) atau *overfitting* (jika *C* terlalu besar).
3. Rentang ini juga dipilih untuk efisiensi eksperimen, agar proses optimasi tidak memakan waktu dan sumber daya komputasi berlebih, mengingat kombinasi parameter dari *kernel* lain juga diuji dalam penelitian ini.

Pengujian parameter pada kernel *linear* dilakukan dengan mengevaluasi nilai *C* pada rentang [0.5, 1, 2, 3, 4]. Hasil validasi silang 5-*fold* serta evaluasi akhir pada data uji dilihat pada Tabel 8 dan Tabel 9.

Tabel 8. Hasil Validasi Kernel *Linear*

C	***Accuracy***	***Presisi***	***Recall***	***F1-score***
0.5	0.894	0.894	0.894	0.891
1	**0.902**	**0.902**	**0.902**	**0.901**
2	0.900	0.900	0.900	0.899
3	0.890	0.889	0.890	0.889
4	0.885	0.885	0.885	0.885

Berdasarkan Tabel 8, nilai *C* = 1 menghasilkan *F1-score* validasi silang tertinggi sebesar 0,901, yang menunjukkan kemampuan generalisasi model yang optimal. Peningkatan nilai *C* dari 0,5 ke 1 meningkatkan performa model (*F1-score* =+0,010). Namun, peningkatan *C* di atas 1 justru menurunkan *F1-score* secara bertahap, yang mengindikasikan risiko *overfitting* akibat kompleksitas model yang berlebihan.

Tabel 9. Evaluasi model terbaik kernel *Linear*

C	***Accuracy***	***Presisi***	***Recall***	***F1-score***
1	0.921	0.920	0.921	0.920

Pada data uji Tabel 9, model dengan *C* = 1 mencapai *F1-score* sebesar 0,920, yang konsisten dengan hasil *validasi silang*. Tingginya *Accuracy* (0,921) dan presisi (0,920) menunjukkan bahwa model tidak hanya mampu mengklasifikasikan sampel dengan tepat, tetapi juga menjaga keseimbangan antara *false positive* dan *false negative*. Waktu pelatihan model selama 82,28 detik mencerminkan efisiensi komputasi yang memadai untuk dataset ini.

b. *Optimasi* Kernel RBF

Pengujian parameter pada kernel RBF dilakukan dengan mengevaluasi kombinasi nilai *C* [0.5, 1, 2, 3, 4] dan parameter γ [0.5, 1, 2]. Hasil *validasi*

silang 5-fold dan evaluasi akhir pada data uji dapat dilihat pada Tabel 10 dan Tabel 11.

Tabel 10. Hasil validasi kernel RBF

C	*γ*	*Accuracy*	*Presisi*	*Recall*	*F1-score*
0.5	0.5	0.865	0.871	0.865	0.855
0.5	1	0.858	0.867	0.858	0.846
0.5	2	0.780	0.818	0.780	0.734
1	0.5	0.893	0.894	0.893	0.889
1	1	0.889	0.890	0.889	0.884
1	2	0.842	0.856	0.842	0.826
2	**0.5**	**0.899**	**0.898**	**0.899**	**0.897**
2	1	0.894	0.894	0.894	0.890
2	2	0.853	0.863	0.853	0.840
3	0.5	0.896	0.896	0.896	0.894
3	1	0.893	0.893	0.893	0.890
3	2	0.853	0.863	0.853	0.840
4	0.5	0.893	0.893	0.893	0.891
4	1	0.893	0.893	0.893	0.890
4	2	0.853	0.863	0.853	0.840

Dari Tabel 10, kombinasi $C = 2$ dan $\gamma = 0.5$ menghasilkan F1-score validasi silang tertinggi sebesar 0.897. Peningkatan nilai *C* dari 0.5 ke 2 meningkatkan performa model (*F1-score* = +0.042), sementara peningkatan γ dari 0.5 ke 2 justru menurunkan *F1-score* hingga - 0.163 pada $C = 2$. Hal ini menunjukkan bahwa nilai γ yang lebih kecil ($\gamma = 0.5$) menghasilkan pola keputusan yang lebih general, sedangkan nilai γ yang lebih besar ($\gamma = 2$) cenderung menyebabkan *overfitting*.

Tabel 11. Evaluasi Model Terbaik Kernel RBF

C	*γ*	*Accuracy*	*Presisi*	*Recall*	*F1-score*
2	0.5	0.915	0.914	0.915	0.914

Pada Tabel 11, model dengan $C = 2$ dan $\gamma = 0.5$ mencapai *F1-score* sebesar 0.914, lebih tinggi dibandingkan hasil validasi silang. *Accuracy* dan recall yang tinggi (0.915) menunjukkan kemampuan model dalam menyeimbangkan false positive dan false negative. Namun, waktu eksekusi pelatihan yang signifikan (478.38 detik) mengindikasikan kompleksitas komputasi kernel *RBF*, terutama pada dataset berskala besar

c. *Optimasi* Kernel *Polynomial*

Pengujian *Polynomial* kernel dilakukan dengan menguji berbagai kombinasi parameter *Cost* (C) dan *Degree* (d). Parameter *C* yang diuji mencakup nilai 0.5, 1, 2, 3, dan 4, sedangkan parameter *d* terdiri dari 1 dan 2. Hasil pengujian dapat dilihat pada Tabel 12.

Tabel 12. Hasil validasi kernel *Polynomial*

C	*d*	*Accuracy*	*Presisi*	*Recall*	*F1-score*
0.5	1	0.894	0.894	0.894	0.891
0.5	2	0.815	0.842	0.815	0.786
1	**1**	**0.902**	**0.902**	**0.902**	**0.901**
1	2	0.852	0.859	0.852	0.840
2	1	0.900	0.900	0.900	0.899
2	2	0.860	0.865	0.860	0.850
3	1	0.890	0.889	0.890	0.889
3	2	0.860	0.865	0.860	0.850
4	1	0.885	0.885	0.885	0.885
4	2	0.860	0.865	0.860	0.850

Berdasarkan Tabel 12, kombinasi $C = 1$ dan $d = 1$ menghasilkan *F1-score* validasi silang tertinggi sebesar 0.901. Peningkatan derajat polinomial dari 1 ke 2 menurunkan *F1-score* rata-rata sebesar 0.051, mengindikasikan bahwa model dengan $d = 2$ cenderung *overfitting* karena kompleksitas berlebihan.

Tabel 13. Evaluasi Model Terbaik Kernel *Polynomial*

C	*d*	*Accuracy*	*Presisi*	*Recall*	*F1-score*
1	1	0.921	0.920	0.921	0.920

Hasil evaluasi pada Tabel 13, data uji menunjukkan bahwa penggunaan kernel *Polynomial* dengan parameter $C = 1$ dan derajat *polinomial* $d = 1$ menghasilkan performa terbaik, ditunjukkan oleh *F1-score* sebesar 0.920. Nilai ini lebih tinggi dibandingkan *konfigurasi* lainnya, termasuk model dengan $C = 2$ dan $d = 2$ yang hanya mencatatkan *F1-score* 0.850. Hasil ini menunjukkan bahwa model dengan kompleksitas *polinomial* lebih rendah ($d = 1$) mampu memberikan performa klasifikasi yang lebih baik dan efisien, dengan waktu eksekusi yang lebih cepat (237.29 detik) dibandingkan model sebelumnya.

d. *Optimasi* Kernel *Sigmoid*

Kombinasi parameter *C* [0.5, 1, 2, 3, 4] dan *γ* [0.5, 1, 2] pada kernel *Sigmoid* dievaluasi untuk menentukan konfigurasi terbaik. Hasil validasi silang dan pengujian akhir terhadap data uji dapat dilihat pada Tabel 14 dan Tabel 15.

Tabel 14. Hasil validasi kernel *Sigmoid*

C	***y***	***Accuracy***	***Presisi***	***Recall***	***F1-score***
0.5	0.5	0.872	0.877	0.872	0.865
0.5	1	0.892	0.893	0.892	0.888
0.5	2	0.896	0.896	0.896	0.894
1	0.5	0.892	0.892	0.892	0.889
1	1	0.899	0.899	0.899	0.898
1	2	0.897	0.896	0.897	0.896
2	0.5	0.902	0.902	0.902	0.901
2	1	0.900	0.900	0.900	0.900
2	2	0.894	0.894	0.894	0.894
3	**0.5**	**0.904**	**0.905**	**0.904**	**0.904**
3	1	0.894	0.894	0.894	0.894
3	2	0.893	0.893	0.893	0.893
4	0.5	0.899	0.900	0.899	0.899
4	1	0.889	0.889	0.889	0.888
4	2	0.886	0.887	0.886	0.886

Dari hasil tersebut, dapat dilihat bahwa konfigurasi *C* = 3 dan *γ* = 0.5 memberikan performa terbaik dalam validasi silang dengan *F1-score* sebesar 0.904. Menariknya, tidak seperti kernel lainnya, peningkatan nilai γ dari 0.5 menjadi 2 hanya menurunkan *F1-score* sebesar 0.011, mengindikasikan bahwa kernel Sigmoid cukup stabil terhadap perubahan nilai γ dan tidak mudah mengalami *overfitting* akibat peningkatan kompleksitas.

Tabel 15. Evaluasi Model Terbaik Kernel *Sigmoid*

C	***y***	***Accuracy***	***Presisi***	***Recall***	***F1-score***
3	0.5	0.922	0.922	0.922	0.922

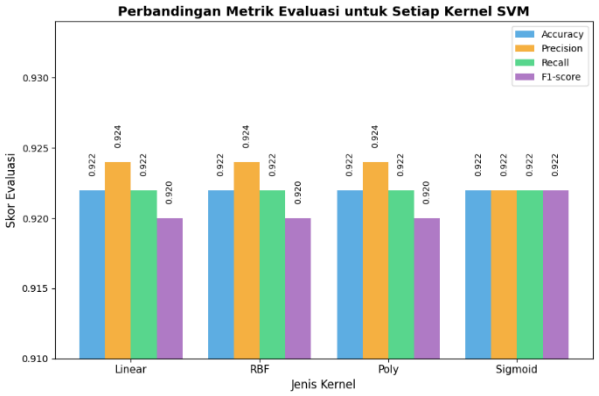
Pada Tabel 15, kombinasi kernel *Sigmoid* dengan *C* = 3 dan *γ* = 0,5 menghasilkan F1-score sebesar 0,922, dengan nilai Accuracy, presisi, dan recall yang identik. Hal ini menunjukkan model SVM yang seimbang, berkat distribusi data hasil pelabelan BERT dan fitur TF-IDF. Meskipun waktu eksekusinya (170,85 detik) lebih lambat dari kernel Linear, namun tetap lebih cepat dari RBF. Parameter ini dinilai ideal untuk analisis sentimen karena akurat dan efisien.

3.6 Perbandingan Pengujian Kernel

Berdasarkan hasil eksperimen terhadap keempat *kernel SVM*, yaitu *Linear*, *RBF*, *Polynomial*, dan *Sigmoid*, diperoleh performa yang bervariasi baik dari sisi *Accuracy* klasifikasi maupun efisiensi waktu komputasi. Tabel 16 dilihat ringkasan nilai *F1-score* tertinggi yang diperoleh masing-masing kernel beserta parameter optimalnya, sementara Gambar 7 memperlihatkan visualisasi perbandingan performa tersebut.

Tabel 16. Ringkasan performa tiap kernel

Kernel	***C***	**γ**	***d***	***F1-score***	**Waktu eksekusi (s)**
Linear	1	-	-	0.920	82.28
RBF	2	0.5	-	0.914	478.38
Polynomial	1	-	1	0.920	237.29
Sigmoid	3	0.5	-	0.922	170.85



Gambar 7. Hasil perbandingan kernel

Berdasarkan Tabel 16 dan Gambar 7, kernel *Sigmoid* memberikan performa terbaik dengan *F1-score* tertinggi (0,922) dan keseimbangan *presisi-recall* yang baik, serta efisiensi komputasi lebih baik dari RBF. Kernel *Linear* dan *Polynomial* (*degree* = 1) mencatatkan *F1-score* 0,920, dengan *Linear* unggul dalam waktu pelatihan tercepat (82,28 detik). Kernel RBF menghasilkan *F1-score* 0,914, namun membutuhkan waktu pelatihan yang lebih lama (478,38 detik). Pemilihan kernel ideal bergantung pada keseimbangan antara *Accuracy* dan efisiensi, di mana kernel *Sigmoid* unggul secara keseluruhan, sedangkan Linear cocok untuk kebutuhan komputasi cepat.

3.7 Word cloud dan Frekuensi Kata

Setelah mengidentifikasi kernel dengan performa terbaik berdasarkan nilai *F1-score*, analisis lebih lanjut dilakukan untuk mengeksplorasi kata-kata yang paling sering muncul dalam masing-masing kategori sentimen. Visualisasi ini bertujuan untuk memahami pola sentimen pengguna terhadap aplikasi *Wondr by* BNI secara lebih mendalam.

Pada tahap ini, dilakukan pemetaan kata-kata yang mendominasi sentimen positif dan negatif berdasarkan hasil pelabelan data pada tahap sebelumnya. Kata-kata yang sering muncul dalam ulasan positif mencerminkan aspek yang disukai pengguna, seperti kemudahan penggunaan atau fitur yang bermanfaat. Sebaliknya, kata-kata dominan dalam ulasan negatif dapat mengindikasikan permasalahan yang sering dihadapi, seperti bug aplikasi atau kendala dalam transaksi. Distribusi kata dalam ulasan positif dan negatif dapat dilihat pada Gambar 9 dan 10.

Tabel 17. Frekuensi kata positif

No	Kata	Jumlah
1	Mudah	419
2	Bagus	114
3	Cepat	103
4	Lengkap	93

Gambar 9. *Word Cloud* Positif

Berdasarkan data frekuensi pada Tabel 17, Gambar 9 menampilkan visualisasi word cloud dari kata-kata yang paling sering muncul dalam ulasan positif.

Tabel 18. Frekuensi kata negatif

No	Kata	Jumlah
1	Gagal	232
2	Verifikasi	170
3	Tolong	219
4	Susah	123

Gambar 10. *Word Cloud* Negatif

Merujuk pada Tabel 18, Gambar 10 menampilkan visualisasi kumpulan kata yang paling sering muncul dalam ulasan negatif. Empat kata dengan frekuensi tinggi, yaitu gagal, verifikasi, tolong, dan susah, mencerminkan isu-isu utama yang sering dihadapi pengguna dalam penggunaan aplikasi.

4. PENUTUP

4.1. Kesimpulan

Penelitian ini menunjukkan bahwa kernel Sigmoid pada algoritma SVM memberikan performa klasifikasi terbaik dengan nilai *F1-score* tertinggi sebesar 0,922 serta efisiensi komputasi yang baik. Kernel *Linear* dan *Polynomial* juga kompetitif, dengan *Linear* unggul dalam waktu pelatihan. Kernel RBF memiliki *Accuracy* tinggi namun memerlukan waktu pelatihan lebih lama. Dari hasil klasifikasi sentimen, kata yang paling sering muncul dalam ulasan negatif adalah "gagal" dan "verifikasi," menunjukkan kendala teknis atau masalah sistem verifikasi. Sementara itu, kata dominan dalam ulasan positif adalah "mudah" dan "bagus," mengindikasikan kemudahan penggunaan dan kepuasan terhadap fitur aplikasi *Wondr by* BNI.

4.2. Saran

Penelitian selanjutnya disarankan mengeksplorasi pendekatan klasifikasi yang lebih kompleks seperti *ensemble learning* atau *deep learning* guna meningkatkan *Accuracy* dan generalisasi model. Selain itu, penerapan analisis sentimen berbasis aspek *(aspect-based sentiment analysis)* dapat memberikan wawasan lebih mendalam terhadap opini pengguna pada fitur tertentu. Perlu juga dilakukan evaluasi terhadap kualitas data latih dan metode pelabelan sentimen, serta mempertimbangkan integrasi teknik preprocessing yang lebih komprehensif agar hasil klasifikasi lebih optimal.

5. DAFTAR PUSTAKA

Rusman, J., Haryati, B. Z., & Michael, A. (2023) 'Optimisasi hiperparameter tuning pada metode Support Vector Machine untuk klasifikasi tingkat kematangan buah kopi', *JICON: Jurnal Informatika dan Komputer*, 11(2), pp. 89–97. https://doi.org/10.35508/jicon.v11i2.12571

Chazar, C., Erawan, B. & Widhiaputra, F. (2020) *Machine learning diagnosis kanker payudara menggunakan algoritma support vector machine*.

Darwis, D. et al. (2020) 'Penerapan algoritma SVM untuk analisis sentimen pada data Twitter Komisi Pemberantasan Korupsi Republik Indonesia', *Jurnal Ilmiah Edutic*, X(X), pp. xx–xx.

Fajri, M. & Primajaya, A. (2023) 'Komparasi teknik hyperparameter optimization pada SVM untuk permasalahan klasifikasi dengan menggunakan grid search dan random search', *Journal of Applied Informatics and Computing (JAIC)*. Available at: http://jurnal.polibatam.ac.id/index.php/JAIC

Idris, M. and Mussalimun, (2024) 'Sentiment Analysis of Google Play Store Reviews using Support Vector Machines', *International Journal of Applied Information Systems*, 12(42), pp. 48–53. Available at: https://www.ijais.org/archives/volume12/number42/sentiment-analysis-ofgoogle-play-store-reviews-using-supportvector-machines

Ranjan, S. and Mishra, S. (2020) 'Comparative Sentiment Analysis of App Reviews', *arXiv preprint*, arXiv:2006.09739. Available at: https://arxiv.org/abs/2006.09739

Fharyana Otang (2024) 'Lonjakan 54,89%, transaksi digital jadi andalan keuangan di Indonesia, 2024', *Bank Indonesia Report*. Available at: https://www.bi.go.id

Che Mohd Safawi, N.U. and Shafie, N.A. (2023) 'Performance of TF-IDF for Text Classification Reviews on Google Play Store: Shopee', *Journal of Computing Research and Innovation*, 9(2). Available at: https://jcrinn.com/index.php/jcrinn/article/view/410

Singgalen, Y.A., 2025. *Improved sentiment classification using multilingual BERT with enhanced performance evaluation for hotel guest review analysis*. Journal of Computer System and Informatics (JoSYC), 6(2), pp.508–520. Available at: https://doi.org/10.47065/josyc.v6i2.6870

Fransiska, S. & Gufroni, A. I. (2020) 'Sentiment analysis provider by U on Google Play Store reviews with TF-IDF and support vector machine (SVM) method', *Scientific Journal of Informatics*, **7**(2), pp. 2407–7658. Available at: http://journal.unnes.ac.id/nju/index.php/sji

Fremmuzar, P. & Baita, A. (2023) 'Uji kernel SVM dalam analisis sentimen terhadap layanan Telkomsel di media sosial Twitter', *Komputika: Jurnal Sistem Komputer*, 12(2), pp. 57–66. https://doi.org/10.34010/komputika.v12i2.9460

Işık, M. & Dağ, H. (2020) 'The impact of text preprocessing on the prediction of review ratings', *Turkish Journal of Electrical Engineering and Computer Sciences*, Turkiye Klinikleri, pp. 1405–1421. https://doi.org/10.3906/elk-1907-46

Isnain, A. R. et al. (2021) 'Sentimen analisis publik terhadap kebijakan lockdown pemerintah Jakarta menggunakan algoritma SVM', *JDMSI*, **2**(1), pp. 31–37. Available at: https://t.co/NfhnfMjtXw

Judijanto, L. et al. (2024) *Transformasi Digital (Teori & Implementasi Menuju Era Society 5.0)*. Available at: https://www.researchgate.net/publication/380462238

Kavabilla, F. E., Widiharih, T. & Warsito, B. (2023) 'Analisis sentimen pada ulasan aplikasi investasi online Ajaib pada Google Play menggunakan metode support vector machine dan maximum entropy', *Jurnal Gaussian*, 11(4), pp. 542–553.

https://doi.org/10.14710/j.gauss.11.4.542-553

Khotimah, A. C. & Utami, E. (2022) 'Comparison Naïve Bayes classifier, K-nearest neighbor and support vector machine in the classification of individual on Twitter account', *Jurnal Teknik Informatika (JUTIF)*, 3(3). https://doi.org/10.20884/1.jutif.2022.3.3.254

Kulkarni, A. & Mhaske, S. (2020) 'Tweet sentiment analysis and study and comparison of various approaches and classification algorithms used', *International Research Journal of Engineering and Technology* [Preprint]. Available at: www.irjet.net

Kurniawan, R. & Apriliani, A. (2020) 'Analisis sentimen masyarakat terhadap virus Corona', *Informatika Sains dan Teknologi*

Meisya, T. et al. (2021) 'Perbandingan kernel support vector machine (SVM) dalam penerapan analisis sentimen vaksinasi COVID-19', *SINTECH*, 4, pp. 139–145. https://doi.org/10.31598

Munandar, D. et al. (2024) 'Analisis sentimen ulasan pengguna aplikasi mobile banking menggunakan algoritma K-nearest neighbor', *Jurnal Teknologi Sistem Informasi dan Aplikasi*, **7**(3), pp. 1309–1318. https://doi.org/10.32493/jtsi.v7i3.41409

Nuraeni Herlinawati et al. (2020) 'Analisis sentimen Zoom Cloud Meetings di Play Store menggunakan Naïve Bayes dan support vector machine'.

Pooja, P. & Bhalla, R. (2022) 'A review paper on the role of sentiment analysis in quality education', *SN Computer Science*. https://doi.org/10.1007/s42979-022-01366-9

Putri, C. M. et al. (2024) 'Perbandingan evaluasi kernel support vector machine dalam analisis sentimen chatbot AI pada ulasan Google Play Store', *Jurnal Teknologi Sistem Informasi dan Aplikasi*, **7**(3), pp. 1236–1245. https://doi.org/10.32493/jtsi.v7i3.41354

Rabbani, S. et al. (2023) 'Perbandingan evaluasi kernel SVM untuk klasifikasi sentimen dalam analisis kenaikan harga BBM', *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 3(2), pp. 153–160. https://doi.org/10.57152/malcom.v3i2.897

Salsabila, A. dan Muntahanah, 2024. Analisis Sentimen Ulasan Aplikasi ChatGPT pada Google Play Menggunakan Metode Support Vector Machine. *Progresif: Jurnal Ilmiah Komputer*, 20(2), pp.714–723.

Sholihah, E. R. M., Diyase, I. G. S. & Puspanigrum, E. Y. (2024) 'Perbandingan kinerja kernel linear dan RBF support vector machine untuk analisis sentimen ulasan pengguna KAI Access pada Google Play Store', *Jurnal Mahasiswa Teknik Informatika*, 8, pp. 728–733

Syahputra, H. (2021) 'Sentiment analysis of community opinion on online store in Indonesia on Twitter using support vector machine algorithm (SVM)', in *Journal of Physics: Conference Series*, IOP Publishing Ltd. https://doi.org/10.1088/1742-6596/1819/1/012030

Wankhade, M., Rao, A. C. S. & Kulkarni, C. (2022) 'A survey on sentiment analysis methods, applications, and challenges', *Artificial Intelligence Review*, **55**(7), pp. 5731–5780. https://doi.org/10.1007/s10462-022-10144-1

Rahardi, M., Aminuddin, A., Abdulloh, F.F. & Nugroho, R.A., 2022. *Sentiment Analysis of Covid-19 Vaccination using Support Vector Machine in Indonesia*. International Journalof Advanced Computer Science and Applications (IJACSA), 13(6), pp.534–539