Journal of Information Technology and Computer Science (INTECOMS)
Volume 8 Nomor 2, Tahun 2025
e-ISSN : 2614-1574
p-ISSN : 2621-3249

COMPARISON OF MACHINE LEARNING CLASSIFICATION ALGORITHM PERFORMANCE IN WONDR BY BNI MOBILE BANKING REVIEW SENTIMENT ANALYSIS

PERBANDINGAN KINERJA ALGORITMA KLASIFIKASI MACHINE LEARNING DALAM ANALISIS SENTIMEN ULASAN MOBILE BANKING WONDR BY BNI

Faizal Riza[1], Dannie Febrianto Hendrakusuma[2], Budi Wibowo[3], Dhian Yusuf Al Afghani[4], Abdurrahman[5]
Institut Teknologi Budi Utomo[1,2,3,4,5]
faizalriza@itbu.ac.id[1]

ABSTRACT

This study analyzes user review sentiments towards services and features offered by WONDR by BNI by applying various Machine Learning classification algorithms. Data were obtained from user reviews on the Google Play Store, which then went through a text pre-processing stage, including data cleaning and normalization, before being used in training the classification model. This study compares the performance of Support Vector Machine (SVM), Light Gradient Boosting Machine (LGBM), XGBoost, Random Forest, CatBoost, and Naïve Bayes in classifying user sentiments. The evaluation results show that SVM has the highest accuracy of 0.881, followed by LGBM with an accuracy of 0.877, which excels in execution time efficiency of 11.80 seconds. XGBoost obtained an accuracy of 0.872, followed by Random Forest with an accuracy of 0.866, while CatBoost and Naïve Bayes recorded accuracies of 0.85 and 0.72, respectively. Although SVM showed the best performance in sentiment classification, this algorithm has the disadvantage of a longer execution time of 57.5 seconds on a machine with 2 GB RAM and 2 vCPU specifications. In contrast, LGBM and XGBoost, although having slightly lower accuracy, showed a faster computational efficiency of 47.5 seconds, making it more optimal for implementation in large-scale systems.

Keywords: *Sentiment Analysis, Machine Learning, Mobile Banking, WONDR by BNI, Sentiment Classification*

ABSTRAK

Penelitian ini menganalisis sentimen ulasan pengguna terhadap layanan dan fitur yang ditawarkan oleh WONDR by BNI dengan menerapkan berbagai algoritma klasifikasi Machine Learning. Data diperoleh dari ulasan pengguna di Google Play Store, yang kemudian melalui tahap pra-pemrosesan teks, mencakup pembersihan data dan normalisasi, sebelum digunakan dalam pelatihan model klasifikasi. Studi ini membandingkan kinerja Support Vector Machine (SVM), Light Gradient Boosting Machine (LGBM), XGBoost, Random Forest, CatBoost, dan Naïve Bayes dalam mengklasifikasikan sentimen pengguna. Hasil evaluasi menunjukkan bahwa SVM memiliki akurasi tertinggi sebesar 0,881, diikuti oleh LGBM dengan akurasi 0,877, yang unggul dalam efisiensi waktu eksekusi sebesar 11,80 detik. XGBoost memperoleh akurasi sebesar 0,872, diikuti oleh Random Forest dengan akurasi 0,866, sementara CatBoost dan Naïve Bayes mencatatkan akurasi masing-masing sebesar 0,85 dan 0,72. Meskipun SVM menunjukkan kinerja terbaik dalam klasifikasi sentimen, algoritma ini memiliki kelemahan dalam hal waktu eksekusi yang lebih lama, yaitu 57,5 detik pada mesin dengan spesifikasi RAM 2 GB dan 2 vCPU. Sebaliknya, LGBM dan XGBoost, meskipun memiliki akurasi yang sedikit lebih rendah, menunjukkan efisiensi komputasi 47,5 detik lebih gegas, sehingga lebih optimal untuk implementasi dalam sistem berskala besar

Kata Kunci: Analisis Sentimen, Machine Learning, Mobile Banking, WONDR by BNI, Klasifikasi Sentimen

PENDAHULUAN

Kemajuan teknologi informasi telah mengubah sektor perbankan digital secara signifikan, terutama dengan hadirnya aplikasi mobile banking yang memudahkan transaksi keuangan melalui perangkat seluler (Basri, 2024). Salah satu inovasi terbaru dalam layanan ini adalah Wondr by BNI, aplikasi mobile banking yang diluncurkan oleh PT Bank Negara Indonesia (Persero) Tbk (BNI) pada 5 Juli 2024. Dirancang sebagai super app, Wondr menawarkan pengalaman yang lebih unggul dibandingkan aplikasi BNI Mobile Banking sebelumnya. Dalam waktu kurang dari enam bulan setelah peluncuran,

aplikasi ini mencapai 5,3 juta pengguna dan mengalami peningkatan volume transaksi hingga 200%, dengan rasio pengguna aktif mencapai 50%, lebih tinggi dibandingkan aplikasi sebelumnya yang hanya 30% (Liu et al., 2025).

Wondr by BNI menyediakan berbagai fitur inovatif untuk memenuhi kebutuhan finansial pengguna. Fitur Mobile Tunai memungkinkan penarikan uang tanpa kartu di ATM BNI, sementara Insight memberikan analisis pengeluaran dan rekomendasi finansial personal (Tondang et al., 2023). Aplikasi ini juga mendukung pembayaran digital QRIS, top-up e-wallet, layanan transfer internasional, dan virtual account. Selain aspek finansial, Wondr memiliki fitur Lifestyle yang memungkinkan pengguna membeli tiket transportasi dan hiburan langsung dari aplikasi, menjadikannya lebih dari sekadar layanan perbankan (Adiningtyas & Auliani, 2024).

Untuk mendukung layanan digital ini, BNI mengandalkan infrastruktur perbankan yang luas, termasuk 1.781 outlet, 13.390 ATM, serta hampir 186 ribu agen branchless banking (Agen46) yang menjangkau berbagai daerah. BNI juga menawarkan BNIDirect, platform digital yang mempermudah transaksi bagi perusahaan dan institusi bisnis (Safi'i et al., 2024). Selain itu, aplikasi Wondr mendapatkan beragam ulasan pengguna di platform mobile. Dengan rating sekitar 4,6/5 di Google Play Store dan 4,7-4,8/5 di Apple App Store, serta lebih dari 43 ribu ulasan di App Store, menunjukkan adanya pengalaman pengguna yang beragam dan perlu dianalisis lebih lanjut (Nurmakhlufi et al., 2024).

Aplikasi Wondr by BNI mendapatkan berbagai respons di platform mobile. Di Google Play Store, aplikasi ini memiliki rating sekitar 4,6/5 bintang, sementara di Apple App Store, ratingnya berkisar antara 4,7 hingga 4,8/5 bintang. Dengan lebih dari 43 ribu ulasan di App Store, respons pengguna mencerminkan beragam pengalaman dalam menggunakan fitur-fitur yang ditawarkan oleh Wondr. Dalam masa awal implementasi, variasi ulasan ini menjadi aspek penting untuk dianalisis guna memahami preferensi dan kendala yang dialami pengguna (Tondang et al., 2023).

Analisis sentimen bertujuan untuk mengolah dan menghasilkan informasi berdasarkan ulasan yang diberikan oleh pengguna aplikasi Wondr by BNI. Informasi yang diekstraksi dari ulasan-ulasan tersebut akan dijadikan sebagai sumber atau acuan untuk melakukan perbaikan terhadap aplikasi. Selanjutnya, persepsi pengguna perlu dikategorikan menjadi ulasan negatif atau ulasan positif (Ramadhan & Andarsyah, 2022). Dalam penelitian ini, eksperimen dilakukan dengan membandingkan performa beberapa algoritma, yaitu Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), *Light Gradient Boosting Machine* (LGBM), *eXtreme Gradient Boosting* (XGBoost), dan *Categorical Boosting* (CatBoost). Tiap algoritma memiliki karakter yang berbeda-beda dan dapat diringkas menjadi narasi seperti berikut :

a. *Support Vector Machine* (SVM)
   Support Vector Machine (SVM) adalah algoritma pembelajaran mesin berbasis pemrograman linier yang menggunakan prinsip margin maksimal dalam supervised learning untuk memisahkan kelas-kelas data. SVM efektif dalam menangani data berdimensi tinggi dan tahan terhadap noise, namun membutuhkan waktu pelatihan yang lama pada dataset besar dan sulit untuk menyesuaikan parameter, terutama pemilihan kernel. Algoritma ini juga tidak mendukung data kategorikal secara langsung, sehingga memerlukan konversi terlebih dahulu. Proses tuning parameter kernel harus dilakukan dengan hati-hati, terutama pada dataset yang kompleks (Pisner & Schnyer, 2020).
b. Naïve Bayes (NB)

Naïve Bayes adalah algoritma klasifikasi berbasis probabilitas yang didasarkan pada Teorema Bayes dengan asumsi independensi antar fitur. Algoritma ini bekerja dengan baik pada data teks, seperti analisis sentimen dan klasifikasi email, serta memiliki kecepatan komputasi yang tinggi. Meskipun sederhana, Naïve Bayes sering memberikan hasil yang kompetitif, terutama untuk dataset dengan jumlah fitur besar. Namun, asumsi independensi antar fitur dapat menjadi kelemahan jika terdapat korelasi yang kuat antar variabel, yang dapat mengurangi akurasi model. Algoritma ini juga membutuhkan sedikit tuning dan lebih efisien dibandingkan dengan metode berbasis pohon keputusan atau deep learning (Insan et al., 2023; Nadira et al., 2023).

c. *Random Forest* (RF)
Random Forest adalah algoritma ensemble berbasis pohon keputusan yang menggunakan pendekatan bagging untuk meningkatkan akurasi dan mengurangi *overfitting*. Dengan membangun banyak pohon keputusan dari subset data yang berbeda, algoritma ini menghasilkan prediksi dengan cara voting untuk klasifikasi atau rata-rata untuk regresi. *Random Forest* memiliki keunggulan dalam menangani dataset dengan banyak fitur dan mampu menangani data yang tidak terstruktur dengan baik. Meskipun lebih lambat dibandingkan algoritma Decision Tree, *Random Forest* lebih cepat dibandingkan model berbasis boosting seperti XGBM atau CatBoost (Larasati et al., 2022).
d. *Light Gradient Boosting Machine* (LGBM)
Algoritma boosting berbasis pohon keputusan, seperti gradient boosting, menggunakan pohon keputusan untuk meningkatkan performa model. Algoritma ini lebih cepat dan efisien dalam menangani dataset besar dibandingkan dengan metode boosting lainnya. Namun, ia memiliki risiko overfitting, terutama ketika diterapkan pada data yang mengandung noise tinggi, yang dapat menurunkan efektivitasnya (Alzamzami et al., 2020).
e. *eXtreme Gradient Boosting* (XGBoost)
Algoritma boosting berbasis pohon keputusan, seperti gradient boosting, sangat efisien dan mendukung penanganan missing value. Meskipun cepat dan efektif untuk dataset besar dan kompleks dengan banyak fitur, algoritma ini rentan terhadap overfitting dan memerlukan tuning parameter yang teliti. Selain itu, untuk data kategorikal, diperlukan encoding, dan parameter model harus disesuaikan secara hati-hati agar mencapai performa optimal (Arif Ali et al., 2023).
f. *Categorical Boosting* (CatBoost)
Algoritma boosting berbasis pohon keputusan, seperti gradient boosting, menggunakan pohon keputusan dan memiliki built-in handling untuk data kategorikal, sehingga lebih mudah diterapkan tanpa perlu encoding. Meskipun lebih lambat dibandingkan dengan LGB, namun tetap lebih cepat daripada SVM dan cocok untuk dataset dengan banyak fitur kategori dan ukuran yang tidak terlalu besar. Algoritma ini memerlukan komputasi yang lebih tinggi dan waktu pelatihan yang lebih lama dibandingkan dengan LGB, namun membutuhkan sedikit tuning dibandingkan dengan XGB dan LGB (Prokhorenkova et al., 2018).

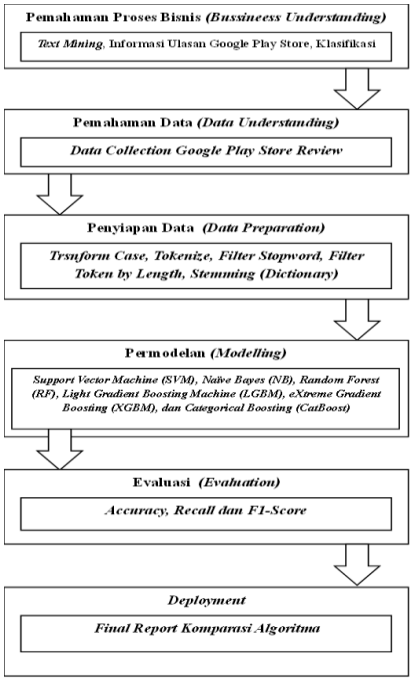
METODE

Penelitian ini menggunakan data dari Google Playstore berupa ulasan dari pengguna aplikasi Wondr by BNI dengan topik pembahasan tentang pengalaman penggunaan aplikasi. Populasi yang ada dalam penelitian ini merupakan semua data ulasan pengguna Wondr by BNI yang berasal dari website Google Play dengan total 163000 ulasan. Dalam penelitian ini, data ulasan yang digunakan adalah ulasan yang diperbarui dalam rentang waktu

antara 17 Agustus 2023 hingga 10 Desember 2024. Pemilihan rentang waktu tersebut didasarkan pada pembaruan aplikasi yang pertama kali dilakukan pada bulan September dan pembaruan kedua pada bulan Juli 2024. Sedangkan sampel data yang digunakan dalam penelitian ini sebanyak 27928 ulasan. Data ulasan yang digunakan sebagai sampel adalah ulasan dari tanggal 5 Desember 2024 sampai dengan 31 Februari 2025 dengan teknik *scraping* menggunakan Google Colab. Pelabelan dilakukan dengan *machine learning* menggunakan dictionary lexicon sentimen positif dan negatif.

Ulasan yang diberikan oleh pengguna dapat bersifat positif, berupa saran atau rekomendasi, maupun negatif, berupa keluhan terhadap aplikasi. Ulasan-ulasan tersebut berpotensi memberikan pengaruh signifikan terhadap upaya perbaikan dan pengembangan aplikasi Wondr by BNI. Namun, untuk memahami lebih lanjut tentang pola tanggapan pengguna, ulasan-ulasan tersebut perlu dianalisis secara mendalam (Nurmakhlufi et al., 2024; Tondang et al., 2023).

Metodologi penelitian yang digunakan dalam penelitian eksperimen ini dengan menggunakan metode *Cross-Industry Standard Process for Data Mining* (CRISP-DM) terdiri dari enam tahap yaitu *Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, dan Deployment* (Singgalen, 2023). Model penelitian ditunjukkan pada gambar 1.



Gambar 1. Kerangka Penelitian
Sumber : Hasil Olahan Penelitian

a. Pemahaman Proses Bisnis (*Bussiness Understanding*)
   Setiap proyek penambangan data dimulai dengan penentuan tujuan yang jelas, yang mencakup fase pertama yaitu pemahaman terhadap kebutuhan bisnis. Tujuan bisnis ini berfokus pada upaya untuk memaksimalkan waktu operasional dan efisiensi mesin melalui penerapan analitik prediktif. Tujuan tersebut kemudian diterjemahkan ke dalam proses penambangan data dengan mengidentifikasi komponen-komponen mesin yang relevan untuk dianalisis (Riza et al., 2020).
b. Pemahaman Data (*Data Understanding*)
   Tujuan dari proyek data mining ditetapkan berdasarkan pengalaman dan asumsi yang kuat. Pada fase Data Understanding, informasi terkait skenario perawatan prediktif disembunyikan untuk mendeteksi potensi kesalahan, dengan konsep yang valid digunakan untuk mencari pola frekuensi baru dalam aliran data yang diperoleh dari sensor gerakan (Kusrini & Taufiq Emha, 2009).
c. Penyiapan Data (*Data Preparation*)

Pada tahap Data Preparation, peneliti mengumpulkan data yang relevan dan menyiapkan dataset untuk keperluan data mining dengan melakukan preprocessing. Proses ini meliputi reduksi data, pemfilteran, serta pembuatan fitur-fitur yang berkaitan dengan tujuan proyek data mining (Kusrini & Taufiq Emha, 2009)

d. Permodelan (*Modelling*)
Pada fase Pemodelan data mining, alur kerja dibangun untuk menemukan pengaturan parameter yang diinginkan dan algoritma yang dipilih untuk dieksekusi. Tugas data mining adalah pada data yang telah diproses sebelumnya (Amra & Maghari, 2017).

e. Evaluasi (*Evaluation*)
Pada fase Evaluasi, model diuji terhadap kumpulan data nyata dalam konteks skenario produksi dan hasil penambangan data dinilai berdasarkan tujuan bisnis yang telah ditetapkan. Untuk tujuan ini, kumpulan data uji dihasilkan dengan mengikuti langkah-langkah yang telah dikembangkan pada fase sebelumnya, dengan pengecualian pada langkah pelabelan data (Zy, 2017).

f. Implementasi (*Deployment*)
Setelah evaluasi berhasil, model yang telah dilatih diterapkan dalam lingkungan produksi pada fase Penerapan. Proses penyebaran model ini memerlukan pengaturan yang stabil untuk akuisisi data, termasuk penyediaan infrastruktur pemrosesan data yang memadai untuk memastikan kelancaran operasional. (Bustami, 2014).

HASIL DAN PEMBAHASAN

Berikut adalah tahapan tahapan penelitian yang dilakukan :

a. Pemahaman Proses Bisnis (*Bussiness Understanding*)
Pada tahap ini, dilakukan pemahaman terhadap objek penelitian dengan cara melakukan scraping pada aplikasi Google Playstore untuk memperoleh ulasan pengguna dari aplikasi Wondr by BNI. Tujuannya adalah untuk mengungkapkan berbagai pendapat, baik yang bersifat negatif maupun positif, yang terdapat dalam ulasan pengguna di platform tersebut. Implementasi pemahaman bisnis ini berfungsi untuk menentukan pendekatan analisis sentimen yang paling tepat serta model yang sesuai, berdasarkan perbandingan hasil dari berbagai algoritma yang diuji.

b. Pemahaman Data (*Data Understanding*)
Pada tahap Data Understanding, penelitian ini mengumpulkan data ulasan pengguna aplikasi mobile banking Wondr by BNI dari website Google Play melalui teknik web scraping menggunakan Google Colab. Data yang diambil antara 5 Desember 2024 sampai dengan 31 Februari 2025 terdiri dari 27.928 ulasan pengguna dalam bentuk teks. Proses scraping dilakukan dengan menginstal *Google Play Scraper* pada Google Colab, mengurutkan ulasan berdasarkan relevansi, dan menampilkan semua rating mulai dari 1 hingga 5. Penggunaan Google Colab untuk melakukan web scraping ditunjukkan pada Gambar 2.

Gambar 1. *Scraping* Ulasan Wondr by BNI

Sumber : Hasil Olahan Penelitian

Untuk mendapatkan dataset yang sesuai maka peneliti melakukan tahap *Bussiness Understanding* dengan langkah sebagai berikut :

1. Menampilkan data *scraping* dengn instruksi *df_womdr.head()* kemudian data ulasan yang sudah didapat

dihitung menggunakan instruksi *len(df_livin. index)*.

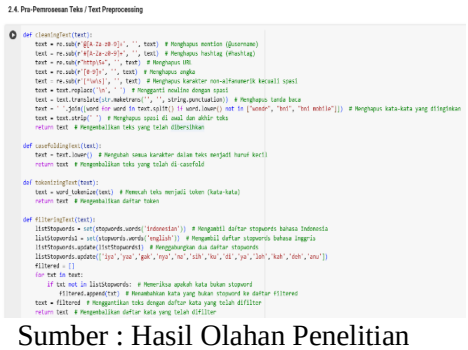
2. Menampilkan data *username*, *rating*, *date time* dan *review* menggunakan instruksi *df_womdr[['username','score', 'at','content']].head()*.
3. Menampilkan data secara berurutan dengan instruksi *my_df.head()*.
4. Menyimpan data *scraping* dengan format file csv dengan nama *hasil_scraping_ wondrbybni.csv*. Data csv yang telah siap untuk dijadikan dataset analisis sentimen ditunjukkan pada Gambar 3.

Gambar 2. Dataset Hasil Proses Scraping

Sumber : Hasil Olahan Penelitian

c. Penyiapan Data (*Data Preparation*)

Tahap Persiapan Data diperoleh dari scraping data menggunakan Google Colab untuk mendapatkan data ulasan pengguna Wondr by BNI pada wahana Google Playstore, kemudian di lakukan pelabelan kategori positif dan negatif. Selanjutnya pembersihan data dilaksanakan untuk mengurangi duplikasi dan redundansi data, lalu transform case, transform remove url, tokenizing, @anotation removal, stopword dan removal seperti pada gambar 4 .



Sumber : Hasil Olahan Penelitian

Gambar 3. Proses Data Preparation

1) Transform Case

Operator yang digunakan pada tahapan ini adalah untuk mengubah huruf kapital yang masih ada pada text akan diubah menjadi huruf kecil semua. Hal ini dilakukan agar ketika dilakukan proses ke dalam model klasifikasi terdapat keseragaman huruf dan tidak terjadi kesalahan dalam proses tokenize.

Tabel 1. *Transform Case*

Data Sebelum	Data Sesudah
Jujur agak downgrade ya dibanding seblumnya, utk modelnya okelah eye catching, tp utk history udh ga se transparan dlu yang lengkap keluar/masuk berapa saldo akhir brp tiap per transaksi. Skrg cuma ada history saldo keluar masuk tp gaada informasi saldo akhir tiap transaksi. Sama perubahan dari sandi huruf min.8 karakter ke sandi yang cuma 6 angka agak diragukan keamanan nya, tp ya semoga aja aman deh.	jujur agak downgrade ya dibanding seblumnya, utk modelnya okelah eye catching, tp utk history udh ga se transparan dlu yang lengkap keluar/masuk berapa saldo akhir brp tiap per transaksi. skrg cuma ada history saldo keluar masuk tp gaada informasi saldo akhir tiap transaksi. sama perubahan dari sandi huruf min.8 karakter ke sandi yang cuma 6 angka agak diragukan keamanan nya, tp ya semoga aja aman deh.

Sumber Data : Hasil Olahan Data Penelitian

2) Transformation remove url

Transformation Remove URL, dalam proses ini link atau URL yang terkandung pada teks akan dihilangkan. Hal ini bertujuan untuk menjadikan kata atau komentar terseleksi hanya teksnya saja.

Tabel 2. Transformation remove url

Data Sebelum	Data Sesudah
kenapa setelah saya ganti hp aplikasi tidak bisa dibuka kembali. setiap setelah verifikasi wajah langsung kembali ke menu pendaftaran terus. Kunjungi kami di https://t.co/K9PnzXDWQ	kenapa setelah saya ganti hp aplikasi tidak bisa dibuka kembali. setiap setelah verifikasi wajah langsung

	kembali ke menu pendaftaran terus. kunjungi kami di

Sumber Data : Hasil Olahan Data Penelitian

3) Tokenization

Kemudian hasil dari proses *Transformation Remove URL* dilanjutkan oleh proses *Tokenization* (Regexp) yaitu semua kata yang ada didalam tiap dokumen dikumpulkan dan dihilangkan tanda baca, angka, simbol, karakter khusus atau apapun yang bukan huruf.

Table 3. *Tokenization* (Regexp)

Data Sebelum	Data Sesudah
kenapa akhir" ini mandiri error ya susah banget untuk transaksi yg katanya pengaturan jam lah atau ulangi lagi dll bener bener mengganggu ktifitas bertransaksi tolong dong dibenahi segera mungkin :(	kenapa akhir ini mandiri error ya susah banget untuk transaksi yg katanya pengaturan jam lah atau ulangi lagi dll bener bener mengganggu ktifitas bertransaksi tolong dong dibenahi segera mungkin

Sumber Data : Hasil Olahan Data Penelitian

4) @ anotation removal

Teks diurai berdasarkan white space. Dalam proses ini, semua anotasi (@) yang terkandung dalam teks dihilangkan dan mengubah seluruh huruf kapital menjadi huruh kecil. Tujuannnya adalah karena annotation (@) biasanya merujuk pada yang melakukan komentar.

Table 4. @Anotation removal

Data Sebelum	Data Sesudah
@admin untuk konek ke shopee pay masih bugs, hanya blank putih ga muncul apa2. sudah di test lewat wifi dan ganti2 sim juga tetap blank putih. mohon di perbaiki	admin untuk konek ke shopee pay masih bugs, hanya blank putih ga muncul apa. sudah di test lewat wifi dan ganti sim juga tetap blank putih. mohon di perbaiki

Sumber Data : Hasil Olahan Data Penelitian

5) Filter stopword removal

Selanjutnya adalah penggunaan operator *Stopword Removal* (*by Dictionary*) yang berfungsi untuk menghilangkan kata-kata yang tidak hubungan dengan isi text. Maka dengan operator *Stopword Removal* (*by Dictionary*) peneliti dapat mendaftarkan kata yang harusnya dihapus dari text.

Table 5. Stopword removal

Data Before	Data After
Kenapa cuma ada Dana keluar dan masuk saja. Sedangkan potongan saldo, penambahan saldo tidak tertulis dalam mutasi rekening. Dulu yg BNI mobile banking ada. Tolonglah kasih catanan saldo min)	kenapa cuma dana keluar dan masuk saja. sedangkan potongan saldo, penambahan saldo tidak tertulis dalam mutasi rekening. dulu bni mobile banking. tolonglah kasih catanan saldo

Sumber Data : Hasil Olahan Data Penelitian

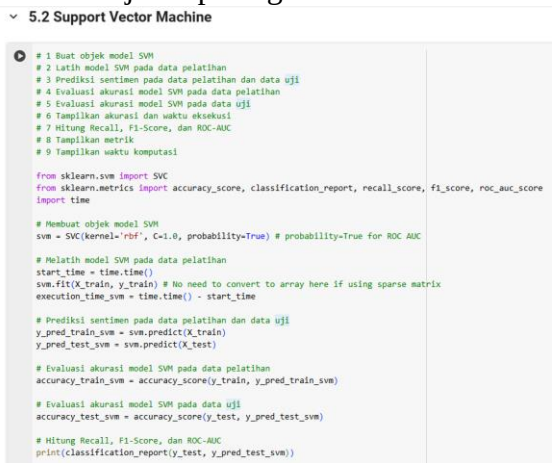
d. Modelling

Pada tahap pemilihan teknik mining, penelitian ini menentukan algoritma yang akan digunakan dengan perbandingan performa antara algoritma *Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), Light Gradient Boosting Machine (LGBM), eXtreme Gradient Boosting (XGBoost), dan Categorical Boosting (CatBoost)*. Tool yang digunakan adalah Google Colab. Pengaturan dan penggunaan operator serta parameter dalam framework Google Colab sangat berpengaruh terhadap akurasi dan model yang terbentuk. (Cristianini & Shawe-Taylor, 2000).

1. Pengujian model algoritma *Support Vector Machine* (SVM)

Peneliti membagi dataset menjadi dua bagian yaitu data latih dan data uji dengan rasio test dan train yaitu 20% data

uji dan 80% data latih. Pengujian model SVM disajikan pada gambar 4.



Sumber : Hasil Olahan Penelitian

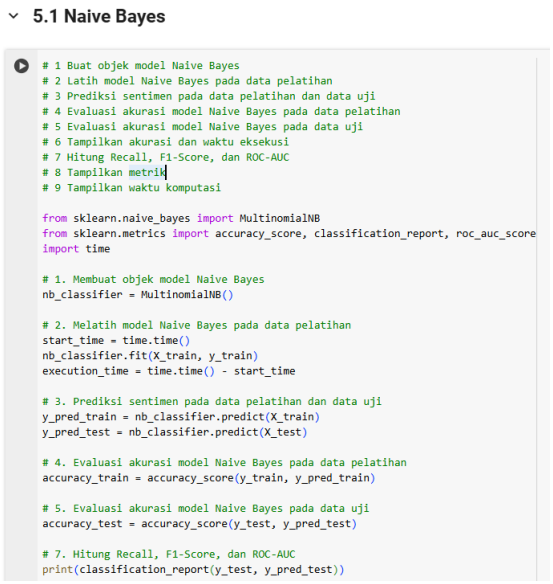
Gambar 4. Training Model SVM

2. Pengujian Model Naïve Bayes (NB)

Model Naive Bayes yang digunakan adalah Multinomial Naïve Bayes yang cocok untuk data kategori atau teks seperti analisis frekuensi kata. Model ini memiliki kecepatan tinggi, membutuhkan sedikit data untuk dilatih, dan tidak mudah overfitting. Kelemahan model Naive Bayes adalah asumsi independensi antar fitur, yang bisa menyebabkan akurasi menurun jika fitur-fitur dalam dataset memiliki korelasi tinggi. Pengujian model Naive Bayes disajikan pada gambar 5.

3. Pengujian Model Random Forest (RF)

Model Radnom Forest Random Forest adalah algoritma ensemble berbasis pohon keputusan yang menggunakan pendekatan bagging untuk meningkatkan akurasi dan mengurangi overfitting. Dengan membangun banyak pohon keputusan dari subset data yang berbeda, algoritma ini menghasilkan prediksi dengan cara voting untuk klasifikasi atau rata-rata untuk regresi. Pengujian model Random Forest disajikan pada gambar 6.



Gambar 6. Training Model Naive Bayes

Sumber : Hasil Olahan Penelitian

Gambar 7. Training Model Random Forest

Sumber : Hasil Olahan Penelitian

4. Pengujian model algoritma Light Gradient Boosting (LGBM)

Model ini menggunakan algoritma *Light Gradient Boosting Machine* (LGBM) yang berbasis pohon keputusan, seperti gradient boosting, menggunakan pohon keputusan untuk meningkatkan performa model. Algoritma ini lebih cepat dan efisien dalam menangani dataset besar dibandingkan dengan metode boosting lainnya. Pengujian model LGBM disajikan pada gambar 7.

5.4 LightGBM

```
# 1 Buat objek model LightGBM
# 2 Latih model LightGBM pada data pelatihan
# 3 Prediksi sentimen pada data pelatihan dan data uji
# 4 Evaluasi akurasi model LightGBM pada data pelatihan
# 5 Evaluasi akurasi model LightGBM pada data uji
# 6 Tampilkan akurasi dan waktu eksekusi
# 7 Hitung Recall, F1-Score, dan ROC-AUC
# 8 Tampilkan metrik
# 9 Tampilkan waktu komputasi

import warnings
warnings.filterwarnings('ignore', category=FutureWarning, message="

# 1. Membuat objek model LightGBM
lgbm_classifier = lgb.LGBMClassifier(random_state=42)

# 2. Melatih model LightGBM pada data pelatihan
start_time = time.time()
lgbm_classifier.fit(X_train, y_train)
execution_time = time.time() - start_time

# 3. Prediksi sentimen pada data pelatihan dan data uji
y_pred_train_lgbm = lgbm_classifier.predict(X_train)
y_pred_test_lgbm = lgbm_classifier.predict(X_test)

# 4. Evaluasi akurasi model LightGBM pada data pelatihan
accuracy_train_lgbm = accuracy_score(y_train, y_pred_train_lgbm)

# 5. Evaluasi akurasi model LightGBM pada data uji
accuracy_test_lgbm = accuracy_score(y_test, y_pred_test_lgbm)

# 7. Hitung Recall, F1-Score, dan ROC-AUC
print(classification_report(y_test, y_pred_test_lgbm))
```

Gambar 7. Training Model LGBM
Sumber : Hasil Olahan Penelitian

5. Pengujian model algoritma eXtreme Gradient Boosting (XGBM)

Model ini menggunakan algoritma boosting berbasis pohon keputusan, seperti gradient boosting, sangat efisien dan mendukung penanganan missing value. Meskipun cepat dan efektif untuk dataset besar dan kompleks dengan banyak fitur. Training model XGBM disajikan pada gambar 8.

5.4 XGBoost

```
# Mengonversi sparse matrix ke array
X_train_array = X_train.toarray()
X_test_array = X_test.toarray()

# Encode labels
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)

# Membuat objek model XGBoost
xgb = XGBClassifier(n_estimators=100, random_state=42)

# Melatih model XGBoost pada data pelatihan
start_time = time.time()
xgb.fit(X_train_array, y_train_encoded)
execution_time = time.time() - start_time

# Prediksi sentimen pada data pelatihan dan data uji
y_pred_train_xgb = xgb.predict(X_train_array)
y_pred_test_xgb = xgb.predict(X_test_array)
```

Gambar 6. Training Model XGBM
Sumber : Hasil Olahan Penelitian

6. Pengujian model algoritma Categorical Boosting (CatBoost)

Model ini berbasis pohon keputusan, seperti gradient boosting, menggunakan pohon keputusan dan memiliki built-in handling untuk data kategorikal. Training model CatBoostterlihat pada gambar 9.

5.5 CatBoost

```
from catboost import CatBoostClassifier
import time

# Membuat objek model CatBoost
cat_model = CatBoostClassifier(n_estimators=100, random_state=42, verbose=0)

# Melatih model CatBoost pada data pelatihan
start_time = time.time()
cat_model.fit(X_train.toarray(), y_train)
execution_time = time.time() - start_time

# Prediksi sentimen pada data pelatihan dan data uji
y_pred_train_cat = cat_model.predict(X_train.toarray())
y_pred_test_cat = cat_model.predict(X_test.toarray())

# Evaluasi akurasi model CatBoost pada data pelatihan
accuracy_train_cat = accuracy_score(y_pred_train_cat, y_train)
```

Gambar 7. Training Model CatBoost
Sumber : Hasil Olahan Penelitian

e. Evaluation

Tahapan evaluasi bertujuan untuk menentukan nilai kegunaan dari model yang telah berhasil dibuat pada langkah sebelumnya. Metode matrix kinerja (*confution matrix*) yang digunakan pada penelitian ini untuk mengevaluasi kinerja model dengan membagi dataset menjadi subset pelatihan dan pengujian secara berulang (Onantya et al., 2019).

Hasil dari confusion matrix akan menghasilkan nilai accuracy, precision, recall dan F1-Score yang diambil dari data test. maka dapat dirangkum hasilnya seperti tabel 5.

Table 6. Perbandingan Confution Matrix

Algoritma	Accuracy	Recall	F1-Score	Execution Time
SVM	**0,881**	**0,881**	**0,881**	57,45s
NB	0,723	0,810	0,691	11,80s
RF	0,866	0,861	0,866	11,80s
LGBM	0,876	0,871	0,872	11,80s
XGBM	0,872	0,871	0,872	11,80s
CatBoost	0,845	0854	0,848	11,80s

Sumber Data : Hasil Olahan Data Penelitian

Berdasarkan hasil pengujian, Support Vector Machine (SVM) menunjukkan performa terbaik dengan akurasi, recall, dan F1-score sebesar 0,881. Namun, keunggulan ini disertai dengan waktu eksekusi tertinggi, yaitu 57,45 detik, yang dapat menjadi tantangan dalam

implementasi pada skala besar. Di sisi lain, algoritma berbasis ensemble seperti *Random Forest* (RF), *LightGBM* (LGBM), dan *XGBoost* (XGBM) menunjukkan performa yang kompetitif dengan akurasi berkisar antara 0,866 hingga 0,876, serta waktu eksekusi yang lebih efisien dibandingkan SVM. LGBM memiliki akurasi tertinggi di antara model ini (0,876), sedikit lebih baik dari XGBM (0,872) dan RF (0,866), dengan eksekusi hanya dalam 11,80 detik, menjadikannya alternatif yang lebih optimal dalam situasi yang memerlukan keseimbangan antara akurasi dan efisiensi waktu pemrosesan.

Naïve Bayes (NB) menunjukkan performa terendah, dengan akurasi 0,723 dan recall yang juga paling rendah (0,810), serta F1-score sebesar 0,691. Hal ini mengindikasikan bahwa NB kurang mampu mengenali pola yang kompleks dibandingkan model lainnya, kemungkinan disebabkan oleh asumsi independensi fitur yang tidak selalu sesuai dalam data nyata. CatBoost, meskipun termasuk dalam algoritma boosting, menunjukkan performa yang lebih rendah dibandingkan dengan LGBM dan XGBM, dengan akurasi 0,845 dan recall 0,854. Meskipun waktu eksekusinya sama dengan LGBM dan XGBM (11,80 detik), CatBoost tidak menawarkan keunggulan yang signifikan dalam hal akurasi, sehingga pemilihannya harus disesuaikan dengan karakteristik data yang digunakan.

Berdasarkan analisis komparatif ini, pemilihan algoritma sebaiknya mempertimbangkan tujuan spesifik dari implementasi model. Jika akurasi tinggi menjadi prioritas utama, maka SVM tetap menjadi pilihan terbaik, meskipun memiliki keterbatasan dalam efisiensi waktu. Namun, dalam kondisi yang mengharuskan keseimbangan antara akurasi tinggi dan efisiensi pemrosesan, LGBM dan XGBM lebih direkomendasikan. Jika model yang lebih ringan dan cepat dibutuhkan, Random Forest atau CatBoost dapat menjadi alternatif yang lebih seimbang antara performa dan efisiensi komputasi. Naïve Bayes, meskipun unggul dalam hal kecepatan, kurang direkomendasikan dalam skenario yang membutuhkan prediksi dengan tingkat akurasi yang tinggi, karena keterbatasannya dalam menangani dependensi antar fitur. Dengan demikian, pemilihan algoritma harus disesuaikan dengan karakteristik data, kebutuhan komputasi, serta tujuan akhir dari implementasi model.

SIMPULAN

Berdasarkan analisis hasil pengujian, pemilihan algoritma harus mempertimbangkan keseimbangan antara akurasi, efisiensi waktu eksekusi, dan kompleksitas komputasi. SVM menunjukkan akurasi tertinggi (0,881), tetapi memiliki waktu eksekusi yang jauh lebih lama, sehingga kurang efisien untuk implementasi berskala besar. LGBM dan XGBM menawarkan akurasi tinggi dengan efisiensi waktu yang lebih baik, menjadikannya alternatif yang optimal untuk aplikasi yang membutuhkan keseimbangan antara kinerja dan waktu pemrosesan. Random Forest dan CatBoost juga menunjukkan performa yang kompetitif, namun CatBoost cenderung kurang unggul dibandingkan LGBM dan XGBM. Naïve Bayes memiliki performa terendah dalam hal akurasi dan recall, meskipun unggul dalam efisiensi komputasi, sehingga kurang cocok untuk skenario yang membutuhkan prediksi yang akurat. Oleh karena itu, dalam pemilihan algoritma, penting untuk mempertimbangkan karakteristik data, tujuan model, serta keterbatasan sumber daya komputasi, guna memastikan bahwa model yang digunakan dapat memberikan hasil yang optimal sesuai dengan kebutuhan aplikasi.

DAFTAR PUSTAKA

Jurnal Ilmiah

Adiningtyas, H., & Auliani, A. S. (2024). Sentiment analysis for mobile banking service quality

measurement. *Procedia Computer Science*, *234*, 40–50.

Alzamzami, F., Hoda, M., & Saddik, A. El. (2020). Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation. *IEEE Access*, *8*, 101840–101858. https://doi.org/10.1109/ACCESS.2020.2997330

Amra, I. A. A., & Maghari, A. Y. A. (2017). Students performance prediction using KNN and Naïve Bayesian. *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*. https://doi.org/10.1109/ICITECH.2017.8079967

Arif Ali, Z., H. Abduljabbar, Z., A. Tahir, H., Bibo Sallow, A., & Almufti, S. M. (2023). eXtreme Gradient Boosting Algorithm with Machine Learning: A Review. *Academic Journal of Nawroz University*, *12*(2), 320–334. https://doi.org/10.25007/AJNU.V12N2A1612

Basri, M. H. (2024). Pemodelan Topik dan Analisis Sentimen pada Teks Ulasan Pengguna Aplikasi Perbankan Seluler di Indonesia. *The Indonesian Journal of Computer Science*, *13*(4). http://ijcs.net/ijcs/index.php/ijcs/article/view/4200

Bustami. (2014). Penerapan Algoritma Naive Bayes. *Jurnal Informatika*.

Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. In *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. https://doi.org/10.1017/cbo9780511801389

Insan, M. K. K., Hayati, U., & Nurdiawan, O. (2023). Analisis Sentimen Aplikasi Brimo Pada Ulasan Pengguna Di Google Play Menggunakan Algoritma Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(1), 478–483.

Kusrini, & Taufiq Emha, L. (2009). Algoritma Data Mining Yogyakarta. In *Algoritma Data Mining* (Issue February). Andi.

Larasati, F. A., Ratnawati, D. E., & Hanggara, B. T. (2022). Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *6*(9), 4305–4313.

Liu, G. R. F., Wea, A., & Kabba, M. N. (2025). Analisis Peran dan Fungsi Bank BNI dalam Meningkatkan Perekonomian Masyarakat dengan Menerapkan Sistem Kredit. *JEMSI (Jurnal Ekonomi, Manajemen, Dan Akuntansi)*, *11*(1), 120–128.

Nadira, A., Setiawan, N. Y., & Purnomo, W. (2023). Analisis Sentimen Pada Ulasan Aplikasi Mobile Banking Menggunakan Metode Naïve Bayes Dengan Kamus Inset. *Indexia: Informatics and Computational Intelligent Journal*, *5*(01), 35–47.

Nurmakhlufi, A., Arsyad, M. R. H., Mulyani, W. S., & Nugroho, K. (2024). Sentiment Analysis on BNI Mobile Application Review Using K-Nearest Neighbors Algorithm. *Sinkron: Jurnal Dan Penelitian Teknik Informatika*, *8*(4), 2490–2502.

Onantya, I., … P. I.-T. I. dan I. K. e, & 2019, undefined. (2019). Analisis Sentimen Pada Ulasan Aplikasi BCA Mobile Menggunakan BM25 Dan Improved K-Nearest Neighbor. *J-Ptiik.Ub.Ac.Id*, *3*(3), 2575–2580.

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. *Machine Learning: Methods and Applications to Brain Disorders*, 101–121. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A.

(2018). CatBoost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, *31*.

Ramadhan, M. A., & Andarsyah, R. (2022). *Klasifikasi Text Spam Menggunakan Metode Support Vector Machine dan Naive Bayes*. Penerbit Buku Pedia.

Riza, F., Rifai, S., Dirgantara, A., Sfenrianto, Rasenda, & Herdyansyah, S. (2020). Information Retrieval Technique for Indonesian PDF Document with Modified Stemming Porter Method Using PHP. *Journal of Physics: Conference Series*, *1477*(3), 1–7. https://doi.org/10.1088/1742-6596/1477/3/032016

Safi'i, M. A., Wijayanti, R. A., Firmansyah, R. Z., & Oktafia, R. (2024). Analisis Optimalisasi Efiensi Operasional Bank BNI Berdasarkan Rasio Biaya Dana Pada Tahun 2019-2021. *Jurnal Ekonomi Bisnis Dan Manajemen*, *2*(2), 58–61.

Singgalen, Y. A. (2023). Penerapan CRISP-DM dalam Klasifikasi Sentimen dan Analisis Perilaku Pembelian Layanan Akomodasi Hotel Berbasis Algoritma Decision Tree (DT). *Jurnal Sistem Komputer Dan Informatika (JSON) Hal*, *237*, 248.

Tondang, B. A., Fadhil, M. R., Perdana, M. N., Fauzi, A., & Janitra, U. S. (2023). Analisis pemodelan topik ulasan aplikasi BNI, BCA, dan BRI menggunakan latent dirichlet allocation. *INFOTECH: Jurnal Informatika & Teknologi*, *4*(1), 114–127.

Zy, A. T. (2017). Comparison Algorithm Classification Naive Bayes, Decission Tree and Neural Network for Analysis Sentiment. *Jurnal Pelita Teknologi*.