Jurnal Ismetek *ISSN 2406-9841*
Vol. 18 No.2 Desember Tahun 2024 *e-IISN 2986-2973*

PERBANDINGAN PERFORMA ALGORITMA KLASIFIKASI MACHINE LEARNING UNTUK ANALISIS SENTIMEN PENGGUNA MOBILE BANKING LIVIN' BY MANDIRI PADA GOOGLE PLAY STORE

Faizal Riza, Bagas Kurniawan
Program Studi Teknik Informatika, Institut Teknologi Budi Utomo Jakarta,
faizalriza@itbu.ac.id, bagaskurniawan@educato.id

Abstrak
Abstrak—Penelitian ini melakukan analisis sentimen pada platform digital Livin' by Mandiri menggunakan empat algoritma klasifikasi machine learning untuk menilai opini pengguna terhadap layanan dan fitur yang ditawarkan. Data diperoleh dari ulasan pengguna di Google Playstore, yang kemudian diproses melalui tahap pra-pemrosesan teks, termasuk pembersihan data dan normalisasi, sebelum digunakan untuk melatih model klasifikasi. Hasil evaluasi menunjukkan bahwa XGBM memiliki akurasi tertinggi (0,868), diikuti oleh LGBM (0,862), yang unggul dalam efisiensi waktu eksekusi (4 detik). SVM dan CatBoost mencatatkan akurasi lebih rendah (0,856), dengan SVM memiliki waktu eksekusi yang sangat lama (1259 detik). Meskipun LGBM dan XGBM menunjukkan keunggulan dalam recall (0,856), LGBM lebih efisien dalam hal waktu, sementara XGBM lebih unggul dalam akurasi. Algoritma LGBM dan XGBM tersebut memberikan performa yang sangat baik, tergantung pada prioritas aplikasi yang lebih menekankan efisiensi atau akurasi.

Kata Kunci : analisis sentimen, machine learning, livin by mandiri, SVM, LGBM, XGBM, CatBoost.

1. PENDAHULUAN

Kemajuan teknologi informasi dan komunikasi saat ini telah berkembang pesat. Salah satu sektor yang mengalami transformasi signifikan adalah layanan perbankan, khususnya melalui aplikasi mobile banking. Mobile banking memungkinkan pengguna untuk melakukan berbagai transaksi keuangan menggunakan perangkat seluler, seperti ponsel pintar atau tablet. Salah satu aplikasi yang banyak digunakan dalam layanan ini adalah *Livin' by Mandiri*, yang dikembangkan oleh Bank Mandiri. Pertumbuhan pesat dalam jumlah transaksi perbankan digital ini didorong oleh peningkatan jumlah pengguna aplikasi tersebut. Tercatat, jumlah pengguna *Livin' by Mandiri* meningkat dari 15 juta pada tahun 2022 dan kini mendekati 19,5 juta pengguna. [1].

Selain menyediakan layanan transaksi finansial, *Livin' by Mandiri* juga mendukung perencanaan keuangan jangka panjang nasabah melalui berbagai fitur, seperti tabungan berjangka (*saving plan*), deposito, serta investasi dalam bentuk reksa dana. Selain itu, aplikasi ini juga memfasilitasi pemenuhan berbagai kebutuhan nasabah, termasuk pembelian tiket perjalanan ke Indonesia serta kebutuhan gaya hidup lainnya, yang dapat diakses melalui fitur *Sukha*. [2].

Pada bulan Desember tahun 2022, Bank Mandiri telah mengembangkan jaringan operasional yang luas di seluruh Indonesia, mencakup 4.050 unit kantor layanan, yang terdiri dari 2.363 kantor cabang dan 1.687 kantor mikro. Selain itu, Bank Mandiri juga memperkuat infrastruktur distribusinya dengan menyediakan 13.068 unit ATM yang terhubung dalam jaringan ATM Link, ATM Bersama, ATM Prima, serta Visa/Plus. Bank ini juga menyediakan layanan *Electronic Data Capture* (EDC) dan berbagai platform e-banking, termasuk aplikasi *New Livin' by Mandiri*, *SMS Banking*, serta *Call Center* 14000, yang memastikan aksesibilitas dan kenyamanan bagi nasabah dalam melakukan transaksi perbankan [1], [3].

Menurut data yang tercatat di situs Google Play pada 14 Agustus 2023, aplikasi *Livin' by Mandiri* memperoleh rating sebesar 3,7 berdasarkan 461.000 ulasan. Hal ini mengindikasikan bahwa masih terdapat potensi untuk aplikasi tersebut dalam melakukan perbaikan kualitas serta pengembangan inovasi guna mencapai rating yang lebih baik. [4].

Google Play menyediakan fitur ulasan atau komentar dari pengguna yang mengunduh aplikasi, termasuk *Livin' by Mandiri*. Dalam penelitian ini, data ulasan yang digunakan adalah ulasan yang diperbarui dalam rentang waktu antara 17 Agustus 2023 hingga 10 Maret 2024. Pemilihan rentang waktu tersebut didasarkan pada pembaruan aplikasi yang pertama kali dilakukan pada bulan Agustus dan pembaruan kedua pada 13 November 2023. 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 *Livin' by Mandiri*. Namun, untuk memahami lebih lanjut tentang pola tanggapan pengguna, ulasan-ulasan tersebut perlu dianalisis secara mendalam [3], [4].

Analisis sentimen bertujuan untuk mengolah dan menghasilkan informasi berdasarkan ulasan yang diberikan oleh pengguna aplikasi Livin' by Mandiri. 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 [5]. Dalam penelitian ini, eksperimen dilakukan dengan membandingkan performa beberapa algoritma, yaitu Support Vector Machine (SVM), Light Gradient Boosting (LGBM), eXtreme Gradient Boosting (XGB), 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 [6].

b. Light Gradient Boosting (LGB)
   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 [7].

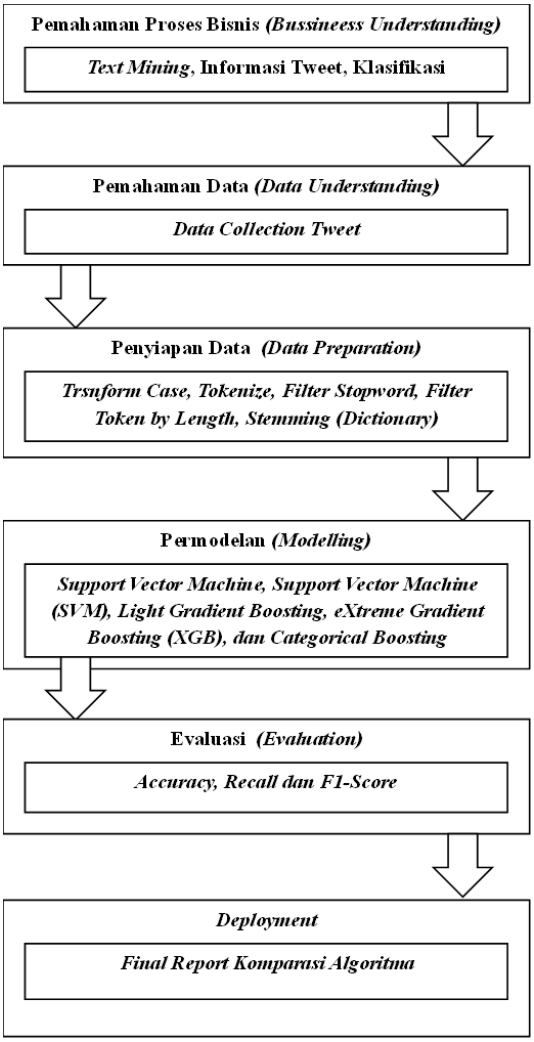
c. eXtreme Gradient Boosting (XGBM)
   Algoritma boosting berbasis pohon keputusan, seperti gradient boosting, sangat efisien dan mendukung penanganan missing value. Meskipun cepat dan efektif untuk dataset besar dan kompleks dengan banyak fitur, algoritma ini rentan terhadap overfitting dan memerlukan tuning parameter yang teliti. Selain itu, untuk data kategorikal, diperlukan encoding, dan parameter model harus disesuaikan secara hati-hati agar mencapai performa optimal [8].

d. 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 [9].

2. METODOLOGI

Penelitian ini menggunakan data dari Google Playstore berupa ulasan dari pengguna aplikasi Livin’ By Mandiri dengan topik pembahasan tentang pengalaman penggunaan aplikasi. Populasi yang ada dalam penelitian ini merupakan semua data ulasan pengguna Livin’ by Mandiri yang berasal dari website Google Play dengan total 200.000 ulasan. Sedangkan sampel data yang digunakan dalam penelitian ini adalah data ulasan mulai dari tanggal 11 Desember 2024 hingga 31 Januari 2025 dengan teknik scraping menggunakan Google Colab. Pelabelan dilakukan dengan machine learning menggunakan dictionary lexicon sentimen positive dan negative.

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 [10]. 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 [10].

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 [11].

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 [11]

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 [12].

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 [13].

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. [14].

3. ANALISIS 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 Livin' By Mandiri. 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.

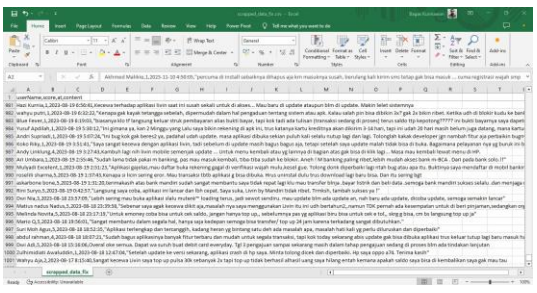
b. Pemahaman Data (*Data Understanding*)
Pada tahap Data Understanding, penelitian ini mengumpulkan data ulasan pengguna aplikasi mobile banking Livin' by Mandiri dari website Google Play melalui teknik web scraping menggunakan Google Colab. Data

yang diambil antara 11 Desember 2024 hingga 31 Januari 2025 terdiri dari 200.000 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 Livin' By Mandiri
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_livin.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_livin[['username','score', 'at','content']].head()*.
3. Menampilkan data secara berurutan dengan instruksi *my_df.head()*.
4. Menyimpan data *scraping* dengan format file csv menggunakan instruksi *my_df.to_csv("scrapped_data.csv", index = false)*. 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 Livin' By Mandiri 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 .

Gambar 3. Proses *Data Preparation*
Sumber : Hasil Olahan Penelitian

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
LIVIN sangat membantu terutama bagi nasabah mandiri. aplikasi ini besar sekali manfaat nya untuk mengurangi durasi waktu dan mempercepat kita melakukan transaksi tanpa harus ke ATM	LIVIN sangat membantu terutama bagi nasabah mandiri. aplikasi ini besar sekali manfaat nya untuk mengurangi durasi waktu dan mempercepat kita melakukan transaksi tanpa harus ke ATM

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 apa2. sudah di test lewat wifi dan ganti2 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
virtual akun nya ribet min pakek living coba cek bank swasta warna biru simpel penguna virtual akun nya, tinggal masukan kode pembayara virtual akun tanpa mencari" masukan pin ada notifikasi pembayaran berhasil (semudah itu min)	virtual akun ribet pakek living coba bank swasta warna biru simpel penguna virtual akun tinggal masukan kode pembayara virtual akun tanpa mencari masukan notifikasi pembayaran berhasil (semudah)

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), *Light Gradient Boosting* (LGBM), *eXtreme Gradient Boosting* (XGB), 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. [15].

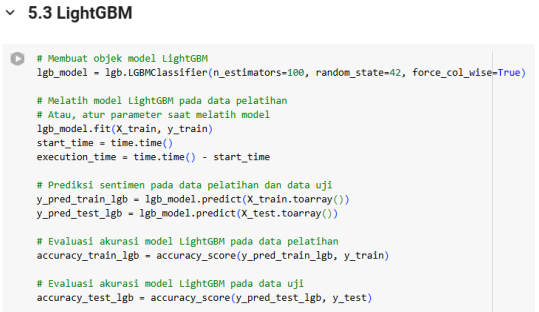
1. Pengujian model algoritma Support Vector Machine (SVM)

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 disajikan pada gambar 4.

Gambar 4. Training Model Algoritma SVM
Sumber : Hasil Olahan Penelitian

2. Pengujian model algoritma Light Gradient Boosting (LGBM)

Model ini menggunakan algoritma untuk Light Gradient Boosting (LGBM). LGBM ini 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 lainnyaseperti disajikan pada gambar 5.

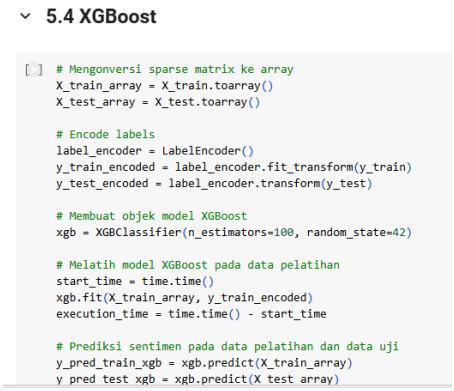


Gambar 5. Training Model LGM
Sumber : Hasil Olahan Penelitian

3. Pengujian model algoritma eXtreme Gradient Boosting (XGBM)

Model ini menggunakan algoritma boosting berbasis pohon keputusan, seperti gradient boosting, sangat efisien dan mendukung

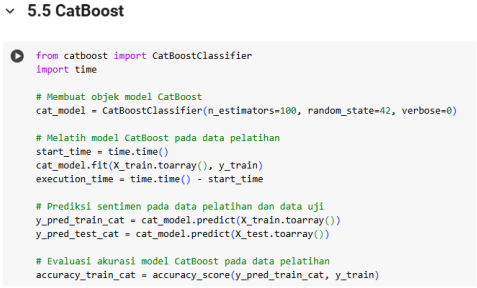
penanganan missing value. Meskipun cepat dan efektif untuk dataset besar dan kompleks dengan banyak fitur. Training terlihat pada gambar 6.



Gambar 6. Training Model XGBM
Sumber : Hasil Olahan Penelitian

4. 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 terlihat pada gambar 7.



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 [16].

Hasil dari confusion matrix akan menghasilkan nilai accuracy, precision, recall dan F1-Score yang diambil dari data test. maka dapat dirangkum hasilnya seperti tabel 6.

Table 5. Stopword removal

Algoritma	Accuracy	Recall	F1-Score	Execution Time
SVM	0,856	0,852	0,856	1259s
LGBM	0,862	**0,856**	**0,857**	**4s**
XGBM	**0,868**	**0,856**	**0,857**	7s
CatBoost	0,856	0854	0,855	14s

Sumber Data : Hasil Olahan Data Penelitian

Berdasarkan data yang tercantum dalam Tabel 5, terdapat perbedaan signifikan dalam hal akurasi, recall, F1-Score, dan waktu eksekusi antara empat algoritma yang diuji: SVM, LGBM, XGBM, dan CatBoost. XGBM (eXtreme Gradient Boosting) mencatatkan akurasi tertinggi yaitu 0,868, mengungguli LGBM (0,862) dan CatBoost serta SVM (kedua mengdapatkan akurasi yang sama yaitu 0,856). Meskipun selisih antara LGBM dan XGBM tidak begitu signifikan, perbedaan ini menunjukkan bahwa XGBM lebih unggul dalam mengenali pola dari data yang lebih kompleks, menghasilkan prediksi yang lebih akurat. SVM, meskipun memiliki performa yang baik dalam hal akurasi dan recall, tetap kalah dibandingkan dengan kedua algoritma boosting dalam hal efisiensi, baik dari sisi waktu eksekusi maupun fleksibilitas dalam menangani data yang lebih besar atau lebih kompleks.

Recall adalah parameter yang mengukur sejauh mana model dapat mengidentifikasi instance positif, semua algoritma menunjukkan performa yang baik. LGBM dan XGBM memiliki recall yang sama yaitu 0,856, sedikit lebih tinggi dibandingkan dengan SVM yang mencatatkan nilai recall 0,852, sementara CatBoost sedikit lebih rendah dengan recall 0,854. Hal ini menunjukkan bahwa meskipun SVM memiliki sedikit keunggulan dalam akurasi, LGBM dan XGBM lebih efisien dalam menangani jumlah instance positif yang lebih banyak. Dengan kata lain, meskipun ada perbedaan kecil antara algoritma-algoritma tersebut dalam hal recall, tidak ada perbedaan yang sangat signifikan yang dapat memengaruhi pengambilan keputusan dalam hal deteksi instance positif. Sebagai hasilnya, LGBM dan XGBM dapat dianggap sebagai algoritma yang lebih handal dalam hal recall, sementara SVM juga masih menunjukkan hasil yang sangat kompetitif meskipun sedikit lebih rendah.

Namun, perbedaan yang paling mencolok terlihat pada waktu eksekusi. LGBM unggul dengan waktu eksekusi tercepat, yaitu hanya 4 detik, diikuti oleh XGBM yang membutuhkan 7 detik, sementara CatBoost membutuhkan 14 detik dan SVM sangat lambat dengan waktu eksekusi mencapai 1259 detik. Meskipun SVM menawarkan akurasi yang baik, waktu eksekusinya yang sangat lama membuatnya kurang efisien dibandingkan dengan LGBM dan XGBM yang sangat cepat dan efisien. Keunggulan LGBM dalam hal waktu eksekusi sangat penting untuk aplikasi dunia nyata yang memerlukan pemrosesan cepat dalam jumlah data yang besar, sementara XGBM memberikan keseimbangan antara akurasi yang tinggi dan waktu eksekusi yang masih dalam batas wajar. CatBoost,

meskipun lebih cepat dari SVM, masih kalah dari LGBM dan XGBM dalam hal waktu eksekusi, meskipun hasil yang diberikan cukup kompetitif. Oleh karena itu, meskipun semua algoritma memberikan hasil yang baik dalam hal akurasi dan recall, pemilihan algoritma terbaik harus mempertimbangkan waktu eksekusi, di mana LGBM menjadi pilihan terbaik untuk aplikasi yang membutuhkan efisiensi waktu, sementara XGBM menawarkan performa yang sedikit lebih unggul dari segi akurasi.

4. KESIMPULAN

Berdasarkan hasil eksperimen dan pengujian, dapat disimpulkan bahwa XGBM (*eXtreme Gradient Boosting*) menawarkan akurasi terbaik di antara keempat algoritma yang diuji, dengan nilai akurasi 0,868, meskipun LGBM (*Light Gradient Boosting Machine*) mencatatkan akurasi yang hampir setara (0,862) dan lebih unggul dalam hal efisiensi waktu eksekusi, hanya membutuhkan 4 detik. Sementara itu, SVM (*Support Vector Machine*) dan CatBoost memiliki akurasi yang lebih rendah (0,856), namun performa SVM tercatat sangat lambat dengan waktu eksekusi mencapai 1259 detik, yang membuatnya kurang praktis dalam aplikasi yang membutuhkan pemrosesan cepat. Dalam hal recall, LGBM dan XGBM menunjukkan nilai yang sangat baik (0,856), sedangkan SVM sedikit lebih rendah (0,852). Meskipun perbedaan dalam recall tidak terlalu signifikan, LGBM dan XGBM memiliki keunggulan dalam kecepatan pemrosesan, menjadikannya pilihan yang lebih efisien untuk aplikasi dunia nyata yang memerlukan keseimbangan antara akurasi, recall, dan waktu eksekusi. Secara keseluruhan, LGBM adalah pilihan terbaik dalam hal waktu eksekusi, sedangkan XGBM unggul dalam hal akurasi, dengan keduanya memberikan performa yang sangat baik dan dapat dipilih sesuai dengan kebutuhan aplikasi yang lebih menekankan pada efisiensi waktu atau keakuratan hasil.

DAFTAR PUSTAKA

[1] B. Mandiri, "Profil Perusahaan," *Retrieved from Bank Syariah Mandiri http//www. syariahmandiri. co. id/category/infoperusahaan/profilperusahaan/profil perusahaan-profilperusahaan*, 2020.

[2] K. A. Ayuningtyas, "Pengaruh Kualitas Pelayanan dan Kepercayaan Terhadap Kepuasan Nasabah (Studi Pada Bank Mandiri Cabang Alam Sutera Tangerang)," *J. Manaj.*, vol. 11, no. 1, pp. 63–76, 2021.

[3] N. Suryani and F. Kusumawati, "The Influence of Service Features and Sales Promotions on Intention to Use Livin'By Mandiri Application," *J. Manag. Energy Bus.*, vol. 2, no. 2, 2022.

[4] R. N. Pradana and others, "Pengaruh Manfaat, Gaya Hidup Dan Kepercayaan Terhadap Keputusan Penggunaan Aplikasi Livin By Mandiri," *J. Manaj. dan Ekon. Kreat.*, vol. 1, no. 2, pp. 46–60, 2023.

[5] M. A. Ramadhan and R. Andarsyah, *Klasifikasi Text Spam Menggunakan Metode Support Vector Machine dan Naive Bayes*. Penerbit Buku Pedia, 2022.

[6] D. A. Pisner and D. M. Schnyer, "Support vector machine," *Mach. Learn. Methods Appl. to Brain Disord.*, pp. 101–121, Jan. 2020, doi: 10.1016/B978-0-12-815739-8.00006-7.

[7] F. Alzamzami, M. Hoda, and A. El Saddik, "Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation," *IEEE Access*, vol. 8, pp. 101840–101858, 2020, doi: 10.1109/ACCESS.2020.2997330.

[8] Z. Arif Ali, Z. H. Abduljabbar, H. A. Tahir, A. Bibo Sallow, and S. M. Almufti, "eXtreme Gradient Boosting Algorithm with Machine Learning: a Review," *Acad. J. Nawroz Univ.*, vol. 12, no. 2, pp. 320–334, May 2023, doi: 10.25007/AJNU.V12N2A1612.

[9] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, Accessed: Jan. 31, 2025. [Online]. Available: https://github.com/catboost/catboost.

[10] F. Riza, S. Rifai, A. Dirgantara, Sfenrianto, Rasenda, and S. Herdyansyah, "Information Retrieval Technique for Indonesian PDF Document with Modified Stemming Porter Method Using PHP," *J. Phys. Conf. Ser.*, vol. 1477, no. 3, pp. 1–7, 2020, doi: 10.1088/1742-6596/1477/3/032016.

[11] Kusrini and L. Taufiq Emha, *Algoritma Data Mining Yogyakarta*, no. February. Yogyakarta: Andi, 2009.

[12] I. A. A. Amra and A. Y. A. Maghari, "Students performance prediction using KNN and Naïve Bayesian," 2017, doi: 10.1109/ICITECH.2017.8079967.

[13] A. T. Zy, "Comparison Algorithm Classification Naive Bayes, Decission Tree and Neural Network for Analysis Sentiment," *J. Pelita Teknol.*, 2017.

[14] Bustami, "Penerapan Algoritma Naive Bayes," *J. Inform.*, 2014.

[15] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. 2000.

[16] I. Onantya, … P. I.-T. I. dan I. K. e, and undefined 2019, "Analisis Sentimen Pada Ulasan Aplikasi BCA Mobile Menggunakan BM25 Dan Improved K-Nearest Neighbor," *J-Ptiik.Ub.Ac.Id*, vol. 3, no. 3, pp. 2575–2580, 2019.