

Perbandingan Kinerja Naive Bayes, Support Vector Machine dan Random Forest Untuk Analisis Sentimen Aplikasi Brimo

Amelia Darwin*, Dinda Lestarini, Iin Seprina

Fakultas Ilmu Komputer, Sistem Informasi, Universitas Sriwijaya, Palembang, Indonesia

Email: ¹*09031282227120@student.unsri.ac.id, ²dinda.lestarini@gmail.com, ³afilasaja99@gmail.com

Email Penulis Korespondensi: 09031282227120@student.unsri.ac.id

Submitted: 13/11/2025; Accepted: 06/12/2025; Published: 08/12/2025

Abstrak—Perkembangan teknologi finansial mendorong peningkatan penggunaan *mobile banking*, termasuk BRImo milik Bank Rakyat Indonesia (BRI). Namun, ulasan pengguna di Google Play Store menunjukkan berbagai keluhan seperti kesulitan *login*, *error* sistem, dan transaksi gagal. Penelitian ini bertujuan menganalisis sentimen pengguna BRImo menggunakan tiga algoritma pembelajaran mesin, yaitu *Naive Bayes*, *Support Vector Machine* (SVM), dan *Random Forest*. Data diperoleh dari 4.996 ulasan melalui *web scrapping* dan diberi label berdasarkan *rating* dengan kategori 1-3 negatif dan 4-5 positif. Dari proses pelabelan diperoleh 4.123 ulasan positif dan 873 ulasan negatif, kemudian diseimbangkan menggunakan *Synthetic Minority Oversampling Technique* (SMOTE). Ekstraksi fitur dilakukan menggunakan TF-IDF. Hasil pengujian menunjukkan bahwa *Random Forest* memberikan performa terbaik dengan akurasi 0,87, *recall* 0.70 dan *F1-score* 0,65 pada kelas negatif, serta *F1-score* 0,92 pada kelas positif. Nilai *macro F1-score* mencapai 0,79 dan lebih tinggi dibandingkan SVM (0,69) dan *Naive Bayes* (0,70). Temuan ini menunjukkan bahwa *Random Forest* lebih efektif dalam mengelompokkan sentimen pengguna BRImo, terutama setelah penyeimbangan data, dan dapat menjadi acuan bagi pengembang dalam meningkatkan kualitas layanan aplikasi.

Kata Kunci: Analisis Sentimen; Aplikasi Brimo; Naive Bayes; Support Vector Machine; Random Forest; SMOTE; Ketidakseimbangan Data; Pembelajaran Mesin; Perbandingan Algoritma

Abstract—The development of financial technology has driven the increasing use of mobile banking, including BRImo, owned by Bank Rakyat Indonesia (BRI). However, user reviews on the Google Play Store show various complaints such as login difficulties, system errors, and failed transactions. This study aims to analyze BRImo user sentiment using three machine learning algorithms: Naive Bayes, Support Vector Machine (SVM), and Random Forest. Data were obtained from 4,996 reviews through web scraping and labeled based on ratings with categories 1-3 negative and 4-5 positive. The labeling process obtained 4,123 positive reviews and 873 negative reviews, which were then balanced using the Synthetic Minority Oversampling Technique (SMOTE). Feature extraction was performed using TF-IDF. Test results showed that Random Forest provided the best performance with an accuracy of 0.87, a recall of 0.70, and an F1-score of 0.65 in the negative class, and an F1-score of 0.92 in the positive class. The macro F1-score reached 0.79, higher than SVM (0.69) and Naive Bayes (0.70). This finding indicates that Random Forest is more effective in classifying BRImo user sentiment, especially after data balancing, and can serve as a reference for developers in improving the quality of application services.

Keywords: Sentiment Analysis; BRImo Application; Naive Bayes; Support Vector Machine; Random Forest; SMOTE; Data Imbalance; Machine Learning; Algorithm Comparison

1. PENDAHULUAN

Perkembangan teknologi digital telah membentuk banyak aspek kehidupan manusia, termasuk sektor keuangan. Salah satu bentuk nyata transformasi tersebut adalah munculnya *financial technology* (fintech) yang mengubah model bisnis perbankan konvensional menuju layanan berbasis digital [1]. Penerapan *fintech* tidak hanya memengaruhi cara bank berinteraksi dengan nasabah, tetapi juga meningkatkan efisiensi dan kenyamanan layanan [2]. Salah satu inovasi penting dalam *fintech* adalah *mobile banking* [3], yang memungkinkan nasabah melakukan transaksi secara mudah dan cepat, seperti *transfer* dana, pembayaran tagihan, dan pengelolaan rekening secara *online* [4]. Hampir seluruh bank besar di Indonesia kini telah mengintegrasikan layanan *mobile banking* guna meningkatkan aksesibilitas serta kepuasan pengguna [5].

Salah satu implementasi *mobile banking* tersebut adalah aplikasi BRImo milik Bank Rakyat Indonesia (BRI), yang menyediakan fitur *transfer*, pembayaran, pembelian, dan manajemen rekening [6]. Meskipun menawarkan kemudahan, banyak pengguna melaporkan kendala seperti kesulitan *login*, transaksi gagal, dan akun terblokir, yang berdampak pada menurunnya kepercayaan pengguna terhadap aplikasi [7]. Penelitian sebelumnya juga menunjukkan bahwa aspek efisiensi BRImo cukup baik, namun faktor *usability* dan stabilitas sistem masih menjadi keluhan utama [8].

Permasalahan tersebut menunjukkan pentingnya pemanfaatan ulasan pengguna sebagai dasar evaluasi kualitas layanan. Namun, tingginya jumlah ulasan menyebabkan analisis manual menjadi tidak efisien. Oleh karena itu, pendekatan otomatis seperti analisis sentimen diperlukan untuk mengidentifikasi persepsi pengguna berdasarkan metode *text mining* dan *Natural Language Processing* (NLP). Penelitian sebelumnya membuktikan bahwa analisis sentimen efektif untuk mengevaluasi aplikasi keuangan digital.

Berbagai algoritma *machine learning* seperti *Naive Bayes*, *Support Vector Machine* (SVM), dan *Random Forest* telah banyak digunakan dalam klasifikasi sentimen. *Naive Bayes* dikenal sederhana namun efektif untuk teks pendek, SVM unggul dalam memisahkan data berdimensi tinggi, sedangkan *Random Forest* menggabungkan banyak *decision tree* untuk meningkatkan akurasi dan mengurangi *overfitting*. Hasil penelitian sebelumnya

menunjukkan bahwa ketiga algoritma tersebut umum digunakan dalam studi perbandingan performa analisis sentimen, dan Random Forest diketahui menghasilkan akurasi tertinggi dibandingkan SVM dan Naive Bayes [9].

Meskipun demikian, penelitian terdahulu belum membahas secara mendalam pengaruh teknik pra-pemrosesan dan penyeimbangan data terhadap performa model. Pada kasus ulasan aplikasi *mobile banking*, data sering kali tidak seimbang, di mana ulasan positif jauh lebih banyak dibandingkan ulasan negatif. Ketidakseimbangan ini dapat menyebabkan algoritma bias dan menurunkan kemampuan model dalam mengenali keluhan pengguna. Oleh karena itu, diperlukan teknik penyeimbangan data seperti *Synthetic Minority Oversampling Technique* (SMOTE), yang mampu memperbaiki representasi kelas minoritas melalui pembangkitan sampel sintetis dan telah terbukti efektif pada data teks tidak seimbang.

Berdasarkan celah tersebut, penelitian ini berkontribusi dengan (1) menerapkan dan membandingkan performa tiga algoritma *Naive Bayes*, SVM, dan *Random Forest* pada analisis sentimen aplikasi BRImo, (2) mengevaluasi pengaruh penggunaan teknik pra-pemrosesan dan penyeimbangan data menggunakan SMOTE, serta (3) menyajikan analisis metrik evaluasi yang lebih komprehensif, termasuk presisi, *recall*, dan *F1-score* pada kelas minoritas. Tujuan dari penelitian ini adalah menganalisis sentimen pengguna aplikasi BRImo dan membandingkan performa ketiga algoritma tersebut setelah menerapkan teknik SMOTE, guna mengetahui algoritma terbaik dalam mengklasifikasikan sentimen pada data ulasan yang tidak seimbang.

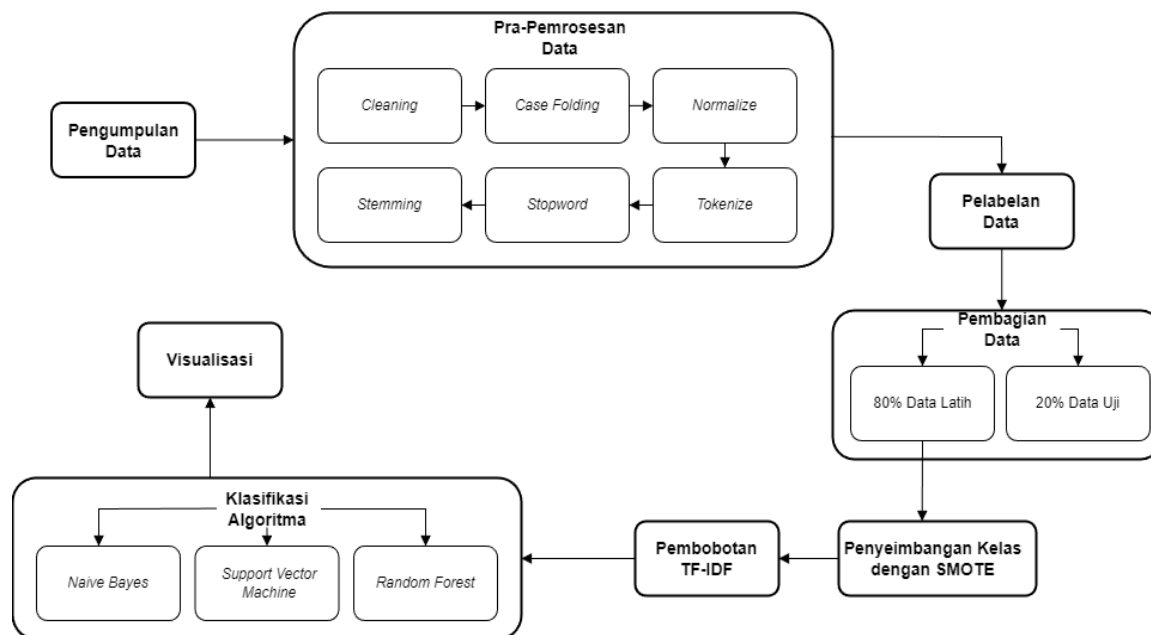
2. METODOLOGI PENELITIAN

2.1 Data Penelitian

Data yang dipergunakan diperoleh melalui teknik *web scrapping* Google Play Store, yang berisi ulasan pengguna aplikasi BRImo. Terdapat 4.996 ulasan dikumpulkan sebagai data utama untuk analisis sentimen. Data ini mencakup teks ulasan beserta *rating* yang diberikan pengguna, yang akan diproses dan dianalisis untuk mendukung tujuan penelitian dalam mengidentifikasi sentimen pengguna terhadap aplikasi BRImo.

2.2 Tahapan Penelitian

Penelitian ini meliputi beberapa tahapan penting yang mencakup proses pengumpulan data, pra-pemrosesan data (*cleaning*, *case folding*, *normalize*, *tokenize*, *stopword* dan *stemming*), pelabelan data, pembagian data menjadi data latih dan data uji, penyeimbangan kelas menggunakan SMOTE, pembobotan TF-IDF, klasifikasi menggunakan algoritma *Naive Bayes*, *Support Vector Machine*, dan *Random Forest*, tahap visualisasi, hingga pembahasan. Seluruh tahapan ini diimplementasikan menggunakan bahasa pemrograman python pada platform Google Colab berbasis web dan rangkaian proses tersebut ditampilkan secara menyeluruh pada gambar 1.



Gambar 1. Tahapan Penelitian

2.2.1 Pengumpulan Data

Tahap pengumpulan data dilakukan menggunakan teknik *web scrapping*, yaitu proses otomatisasi pengambilan data dari halaman web menggunakan skrip berbasis *Python* [10]. Data yang diambil mencakup teks ulasan, *rating* pengguna, serta tanggal unggahan. Pengumpulan data dilakukan pada tanggal 22 oktober, dengan mengambil ulasan

dalam rentang 3 bulan terakhir dari tanggal tersebut. Dengan demikian, data yang terkumpul berasal dari periode 22 juli 2025 hingga 22 oktober 2025. Seluruh data kemudian disimpan dalam format CSV untuk digunakan pada tahap prapemrosesan dan analisis berikutnya. Dalam penelitian ini terkumpul sebanyak 4.996 ulasan, yang terdiri atas teks ulasan serta rating numerik pengguna aplikasi BRImo. Rangkaian proses pengambilan data tersebut ditunjukkan pada Gambar 2.



Gambar 2. Tahap Pengumpulan Data

2.2.2 Prapemrosesan Data

Prapemrosesan data merupakan tahap penting dalam analisis sentimen karena berfungsi menyiapkan teks agar dapat diolah secara optimal oleh algoritma *machine learning* [11]. Tahap pertama yang dilakukan adalah *cleaning*, yaitu menghapus berbagai elemen yang tidak relevan seperti URL, *hashtag*, *mention*, emotikon, angka, dan simbol lain yang tidak memiliki kontribusi terhadap analisis [12]. Setelah itu dilakukan *case folding* dengan mengubah seluruh teks menjadi huruf kecil untuk menjaga konsistensi penulisan. Tahap berikutnya adalah *normalize*, yang pada penelitian ini dilakukan dengan mengubah kata-kata tidak baku atau slang menjadi bentuk baku menggunakan kamus slang/alay bahasa Indonesia, misalnya “gk” menjadi “tidak”, “udh” menjadi “sudah”, dan bentuk singkatan lain yang umum digunakan dalam ulasan pengguna. Setelah *normalize*, teks diproses melalui *tokenize* untuk memecah kalimat menjadi unit kata. Kemudian dilakukan *stopword* untuk menghilangkan kata-kata umum yang tidak berpengaruh pada penentuan sentimen, seperti “dan”, “atau”, dan “adalah”. Tahap terakhir adalah *stemming*, yaitu mengubah kata berimbuhan menjadi bentuk dasarnya menggunakan algoritma Sastrawi Stemmer [13]. Seluruh tahapan ini memastikan teks berada dalam kondisi yang bersih dan konsisten sebelum melalui proses pembobotan TF-IDF dan klasifikasi.

2.2.3 Pelabelan Data

Pelabelan Data dilakukan terhadap data ulasan berdasarkan *rating* pengguna. Ulasan dengan rating 1,2 dan 3 ditetapkan sebagai kategori sentimen negatif, sementara rating 4 dan 5 dikategorikan sebagai sentimen positif. Tahapan ini menghasilkan data yang telah diberi label berupa kategori positif serta negatif, selanjutnya dipergunakan pada proses pelatihan model [14].

2.2.4 Pembagian Data

Tahap pembagian data penelitian menggunakan proporsi 80% data latih serta 20% data uji. Pemecahan tersebut tujuannya supaya model mampu belajar pada data cukup representatif serta mampu memberikan performa optimal ketika diuji menggunakan data baru belum pernah dilihat terdahulu [8]. Beberapa penelitian melihat pemilihan rasio terdapat pengaruh cukup besar atas performa model, rasio 80:20 menjadi salah satu standar umum diterapkan di beragam penelitian serupa [15].

2.2.5 Penyeimbangan Kelas dengan SMOTE

Karena terdapat ketidakseimbangan total data kelas positif serta negatif, penelitian ini menerapkan teknik SMOTE (Synthetic Minority Oversampling Technique) guna memperbaiki proporsi data antar kelas. Metode ini berfungsi menyelaraskan sebaran data secara mendapat sampel sintesis dari kelas minoritas. Penting untuk diperhatikan bahwa penerapan SMOTE dilakukan hanya pada data latih, agar model dapat mempelajari pola dari data lebih seimbang sehingga mendapat prediksi lebih akurat dan tidak bias terhadap kelas mayoritas.

2.2.6 Pembobotan TF-IDF

Metode pembobotan kata yang digunakan dalam penelitian ini adalah TF-IDF (Term Frequency–Inverse Document Frequency) untuk merepresentasikan teks dalam bentuk vektor numerik [16]. Teknik ini bekerja dengan menghitung frekuensi kemunculan suatu kata pada sebuah dokumen serta membandingkannya dengan seluruh dokumen dalam korpus, sehingga dapat menggambarkan tingkat kepentingan kata tersebut [17]. Pada penelitian ini, fitur TF-IDF dibangun menggunakan unigram ($ngram_range = (1,1)$), yaitu pembobotan berbasis kata tunggal. Pemilihan *unigram* dilakukan karena jenis fitur ini paling umum digunakan pada analisis sentimen dan memberikan representasi yang stabil terhadap teks pendek seperti ulasan pengguna aplikasi. Nilai TF-IDF yang lebih tinggi menunjukkan bahwa suatu kata sering muncul pada satu dokumen tetapi jarang muncul pada dokumen lain, sehingga kata tersebut berpotensi menjadi penanda penting dalam proses klasifikasi sentimen.

2.2.7 Klasifikasi Algoritma

Penelitian ini memanfaatkan tiga algoritma *machine learning*, yakni *Naïve Bayes*, *Support Vector Machine* (SVM), *Random Forest*. Algoritma *Naïve Bayes* didasarkan pada konsep probabilistik yang bertujuan menghitung



kemungkinan suatu kategori diberi ke sebuah dokumen teks mempertimbangkan probabilitas gabungan antara kata serta kategori [18]. *Support Vector Machine* berguna untuk menemukan fungsi pemisah terbaik di antara fungsi yang ada untuk memisahkan dua macam obyek [19]. Terakhir *random forest* merupakan metode yang membangun beberapa pohon keputusan dengan menggunakan sampel dan data latih secara acak[20]. Setiap algoritma memiliki prinsip kerja berbeda, sehingga perlu dilaksanakan evaluasi kinerja memanfaatkan metrik berupa akurasi, presisi, *recall*, *F1-score* guna mengukur performanya secara menyeluruh.

2.2.8 Visualisasi

Selanjutnya tahap visualisasi meningkatkan pemahaman data melalui pembuatan *word cloud* yang merepresentasikan frekuensi kata pada ulasan pengguna, ukuran *font* menunjukkan tingkat kemunculan kata tersebut. Tahap visualisasi dapat membantu mendemonstrasikan distribusi setiap kategori dalam kumpulan data, misalnya, menggunakan diagram lingkaran, diagram batang, atau plot hitungan [8].

2.3 Evaluasi Kinerja Model

Evaluasi kinerja model dilaksanakan guna mengukur sejauh mana model mampu memenuhi tujuan yang sudah ditentukan. Satu dari metode umum dipergunakan saat mengukur performa model klasifikasi yakni *Confusion Matrix*. *Confusion Matrix* berfungsi sebagai alat evaluasi menggambarkan kinerja model melihatkan perbandingan antara hasil prediksi yang benar serta yang salah pada tiap kelas, maka memberi deskripsi visual tentang kompetensi model melaksanakan kelompok secara akurat. sehingga dapat dihitung berapa metrik penting seperti akurasi, presisi, *recall*, dan *F1-score*. Nilai nilai ini membantu peneliti memahami sejauh mana model mampu mengenali setiap kelas dengan benar serta mendeteksi kemungkinan kesalahan klasifikasi yang terjadi.

Tabel 1. Confusion Matrix

| Actual | Predicted | |
|----------|---------------------|---------------------|
| | Prediksi Positive | Prediksi Negative |
| Positive | True Positive (TP) | False Negative (FN) |
| Negative | False Positive (FP) | True Negative (TN) |

Pada Tabel 1 menampilkan rincian mengenai tingkat akurasi model mengelompokkan sentimen pengguna ke kategori positif maupun negatif. Bagian “Actual” menunjukkan label asli dari data, sementara kolom “Predicted” menampilkan hasil prediksi yang dihasilkan oleh model. Informasi yang disajikan matriks menjadi dasar dalam perhitungan metrik evaluasi berupa akurasi, *recall*, presisi, *F1-score*, berfungsi menilai performa model menyeluruh. Merujuk nilai-nilai pada *confusion matrix*, keempat metrik tersebut dihitung menggunakan persamaan (1) hingga (4).

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Presisi = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} = TPR \tag{3}$$

$$F1 - score = \frac{2x(RecallxPrecision)}{Recall+Precision} \tag{4}$$

3. HASIL DAN PEMBAHASAN

3.1 Data Ulasan

Sebanyak 4.996 data ulasan pengguna aplikasi BRImo dikumpul lewat proses *web scraping* dari Google Play Store. *Web scraping* merupakan teknik pengambilan data otomatis disuatu situs web melalui proses ekstraksi informasi. Data yang diperoleh mencakup berbagai jenis ulasan dengan rentang penilaian yang beragam, mulai dari yang bernada positif hingga negatif. Contoh hasil data yang diperoleh ditunjukkan pada Tabel 2, yang memuat beberapa sampel ulasan pengguna beserta *rating* dan waktu unggahannya.

Tabel 2. Hasil Scrapping Data

| Username | Content | Score | at |
|-----------------|---|-------|------------------|
| Pengguna Google | BRImo sering error, saldo tidak muncul! | 1 | 22/10/2025 07:07 |
| Pengguna Google | Registrasi mudah, tapi login kadang gagal | 3 | 12/09/2025 09:10 |
| Pengguna Google | Aplikasi BRImo saya berputar terus pas mau login, lama banget 😞 | 5 | 10/09/2025 14:12 |

3.2 Prapemrosesan Data

Prapemrosesan, ialah langkah membersihkan data meliputi penghilangan data duplikat, verifikasi ketidaksesuaian data dan pemulihan kesalahan data. Tahapan yang dilakukan meliputi penghapusan karakter tidak relevan, konversi huruf menjadi kecil, *normalize*, *Tokenize*, *stopword*, dan *stemming* [21]. Dengan melakukan prapemrosesan secara menyeluruh, kualitas data meningkat, noise dapat diminimalkan, dan akurasi hasil klasifikasi sentimen menjadi lebih optimal.

3.2.1 Cleaning

Tahap *cleaning* tujuannya bersihkan teks pada elemen yang tidak sesuai, berupa tanda baca, simbol, emotikon, atau angka yang tidak memiliki makna kontekstual. Proses ini memastikan data ulasan menjadi lebih terstruktur dan siap digunakan pada tahap analisis berikutnya. Sebagai contoh, teks “BRImo sering error, saldo tidak muncul!” diubah menjadi “BRImo sering error saldo tidak muncul”. Tahapan ini sangat penting karena kualitas hasil analisis teks sangat bergantung pada kebersihan dan konsistensi data yang digunakan. Contoh hasil data yang telah melalui proses *cleaning* ditunjukkan pada Tabel 3.

Tabel 3. Hasil *Cleaning*

| Ulasan Asli | Hasil <i>Cleaning</i> |
|---|--|
| BRImo sering error, saldo tidak muncul! | BRImo sering error saldo tidak muncul |
| Registrasi mudah, tapi login kadang gagal | Registrasi mudah tapi login kadang gagal |
| Aplikasi BRImo saya berputar terus pas mau login, lama banget 😞 | Aplikasi BRImo saya berputar terus pas mau login lama banget |

3.2.2 Case Folding

Case Folding adalah tahap yang tujuannya menyeragamkan format teks dengan mengganti seluruh huruf besar (uppercase) jadi bentuk kecil (lowercase). Proses ini dilakukan supaya sistem dapat memperlakukan kata yang serupa dengan bentuk penulisan berbeda secara konsisten, misalnya “BRImo”, “BRIMO”, dan “brimo” dianggap identik dalam analisis. Contoh hasil proses *casefolding* ditunjukkan pada Tabel 4.

Tabel 4. Hasil *Casefolding*

| Hasil <i>Casefolding</i> |
|--|
| brimo sering error saldo tidak muncul |
| registrasi mudah tapi login kadang gagal |
| aplikasi brimo saya berputar terus pas mau login lama banget |

3.2.3 Normalize

Tahap *normalize* dilakukan untuk menyeragamkan kata-kata yang ditulis secara tidak baku atau menggunakan singkatan menjadi bentuk kata baku agar lebih konsisten dalam proses analisis. *Normalize* pada penelitian ini difokuskan pada penggantian kata slang atau singkatan yang umum digunakan pengguna, seperti “gk” menjadi “tidak”, “udh” menjadi “sudah”, atau “bgt” menjadi “banget”. Proses ini diterapkan setelah tahap *cleaning* dan *case folding*, sebelum memasuki proses *tokenize*, *stopword* dan *stemming*. Contoh hasil *normalize* dapat dilihat pada Tabel 5.

Tabel 5. Hasil *Normalize*

| Hasil <i>Normalize</i> |
|---|
| brimo sering error saldo hilang |
| registrasi mudah tetapi login sering gagal |
| aplikasi brimo saya berputar terus saat mau login lama sangat |

3.2.4 Tokenize

Merupakan proses memecah teks menjadi unit terkecil berupa kata agar dapat dianalisis secara terpisah. Pada tahap ini, setiap kalimat hasil *normalize* diubah menjadi daftar kata yang mewakili isi ulasan pengguna. Contoh hasil proses *tokenize* ditunjukkan pada Tabel 6.

Tabel 6. Hasil *Tokenize*

| Hasil <i>Tokenize</i> |
|--|
| ["brimo", "sering", "error", "saldo", "hilang"] |
| ["registrasi", "mudah", "tetapi", "login", "sering", "gagal"] |
| ["aplikasi", "brimo", "saya", "berputar", "terus", "saat", "mau", "login", "lama", "sangat"] |

3.2.5 Stopword

Stopword merupakan proses menghapus kata yang dianggap tidak relevan atau kata-kata umum dianggap tidak terdapat arti signifikan dalam proses analisis teks, seperti “dan”, “atau”, serta “adalah”. Penghapusan *stopword* dilaksanakan menurunkan *noise* pada data serta memusatkan analisis ke kata-kata terdapat makna relevan terhadap konteks pembahasan. Hasil proses *stopword* diperlihatkan pada Tabel 7.

Tabel 7. Hasil *Stopword*

| Hasil <i>Stopword</i> |
|--|
| ["brimo", "error", "saldo", "hilang"] |
| ["registrasi", "mudah", "login", "gagal"] |
| ["aplikasi", "brimo", "berputar", "login", "lama", "sangat"] |

3.2.6 Stemming

Pada tahap ini, dilakukan upaya untuk mengembalikan berbagai bentuk kata ke bentuk dasar atau akarnya, menghapus imbuhan tertentu. Misalnya, kata "berlari", "pelari", dan "berlari-lari" akan dikonversi menjadi bentuk dasar "lari", sehingga membantu meningkatkan efektivitas dalam pengolahan teks karena variasi kata disederhanakan menjadi satu bentuk dasar. Contoh hasil *stemming* ditampilkan pada Tabel 8.

Tabel 8. Hasil *Stemming*

| Hasil <i>Stemming</i> |
|---|
| [brimo, error, saldo, hilang] |
| [registrasi, mudah, login, gagal] |
| ["aplikasi", "brimo", "putar", "login", "lama", "sangat"] |

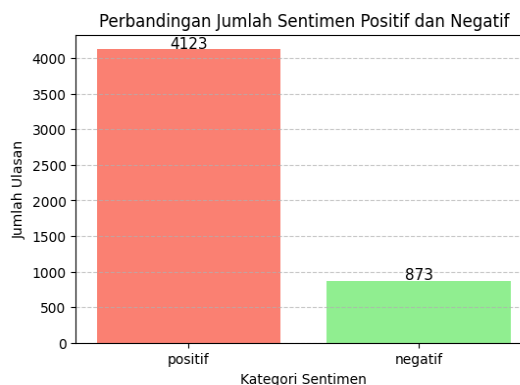
3.3 Pelabelan Data

Dalam penelitian ini, proses pelabelan dilakukan dengan mengacu pada *rating* yang diberikan pengguna pada ulasan aplikasi BRImo. Ulasan dengan *rating* 4 dan 5 dikategorikan sebagai sentimen positif, sedangkan ulasan dengan *rating* 1, 2, dan 3 dimasukkan ke dalam kategori sentimen negatif. Berdasarkan proses tersebut, dari total 4.996 ulasan yang diperoleh, terdapat 4.123 ulasan positif dan 873 ulasan negatif. Perbedaan jumlah yang cukup besar ini menunjukkan bahwa dataset awal bersifat tidak seimbang, sehingga pada tahap berikutnya diperlukan penerapan teknik penyeimbangan data seperti SMOTE agar model tidak bias terhadap kelas mayoritas. Contoh hasil proses pelabelan ditunjukkan pada Tabel 9.

Tabel 9. Proses Pelabelan Data

| Content | Label |
|--|---------|
| aplikasi sangat membantu untuk melakukan transaksi dimana dan kapan saja ðŸ™ | Positif |
| gagal bayar qris... ktanya eror smpe 3x...tp saldo terpotong n hilang.... udh tanya Sabrina.. kirim email ga ada kejelasan sma sekali... bnr" mengecewakan! | Negatif |
| bri yang terbaik, sudah cukup lama saya pengguna brimo. bertransaksi juga mudah sekali | Positif |
| hadeh kok bisa ada admin fee sama fee monthly, mana 18500 lagi lu kira tidak sedikit tuh, harusnya satu saja cukup admin fee atau fee monthly, potongannya sangat sangat menyusahkan | Negatif |

Perbedaan jumlah ulasan positif dan negatif yang cukup signifikan terlihat lebih jelas melalui visualisasi pada grafik distribusi sentimen. Grafik ini menggambarkan proporsi masing-masing kategori, sehingga memudahkan peneliti dalam memahami ketidakseimbangan data sebelum dilakukan proses penyeimbangan menggunakan SMOTE. Visualisasi tersebut ditunjukkan pada Gambar 3.



Gambar 3. Grafik Label Ulasan

3.4 Penerapan Teknik SMOTE

Masalah ketidakseimbangan data pada ulasan pengguna ditangani dengan menerapkan teknik SMOTE (*Synthetic Minority Oversampling Technique*). Dari total 4.996 ulasan, data terlebih dahulu dibagi menjadi 80% data latih dan 20% data uji. Pembagian dilakukan secara proporsional berdasarkan kelas sehingga distribusi tiap kelas pada data latih mengikuti rasio data sebenarnya. Dari hasil pembagian tersebut, diperoleh 3.298 data latih kelas positif dan 698 data latih kelas negatif.

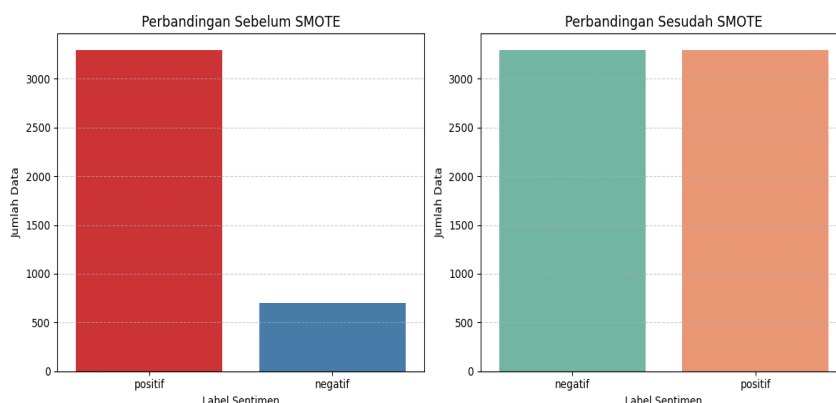
Teknik SMOTE hanya diterapkan pada data latih, sesuai praktik standar *machine learning* untuk menghindari terjadinya *data leakage*. Proses SMOTE bekerja dengan menghasilkan sampel sintesis pada kelas minoritas (negatif) hingga jumlahnya seimbang dengan kelas mayoritas[9]. Setelah proses ini diterapkan, jumlah sampel pada kelas negatif meningkat dari 698 menjadi 3.298, sehingga kedua kelas memiliki distribusi yang seimbang sebagaimana yang ditunjukkan pada tabel 10.

Sementara itu, data uji tidak melalui proses SMOTE dan dipertahankan dalam bentuk distribusi aslinya, yaitu 825 ulasan positif dan 175 ulasan negatif sebagaimana terlihat pada *confusion matrix* di bagian 3.5. Dengan demikian, SMOTE berfungsi meningkatkan representasi kelas minoritas selama pelatihan tanpa memengaruhi data uji, sehingga model dapat dievaluasi secara objektif pada data yang mencerminkan kondisi sebenarnya.

Tabel 10. Hasil Penyeimbangan Kelas dengan SMOTE

| Label | Sebelum SMOTE | Sesudah SMOTE |
|---------|---------------|---------------|
| Positif | 3.298 | 3.298 |
| Negatif | 698 | 3.298 |

Perbandingan visual antara distribusi data sebelum dan sesudah penerapan SMOTE ditampilkan pada Gambar 4, yang memperlihatkan perubahan signifikan dari ketidakseimbangan menuju kondisi yang setara antara kelas positif dan negatif.



Gambar 4. Grafik Perbandingan Hasil Sebelum dan Sesudah SMOTE

3.5 Hasil Klasifikasi

3.5.1 Algoritma *Naive Bayes*

Untuk menilai performa model setelah melalui proses pelatihan, dilakukan evaluasi menggunakan metrik *precision*, *recall*, dan *F1-score*. Ketiga metrik ini digunakan untuk melihat kemampuan model dalam mengenali kelas negatif maupun positif secara seimbang, mengingat data uji tetap memiliki distribusi yang tidak setara. Hasil evaluasi tersebut disajikan pada Gambar 5, yang menampilkan laporan klasifikasi lengkap berupa nilai *precision*, *recall*, *F1-score*, serta jumlah data uji (*support*) pada masing-masing kelas.

```

♦ Classification Report:
      precision    recall  f1-score   support

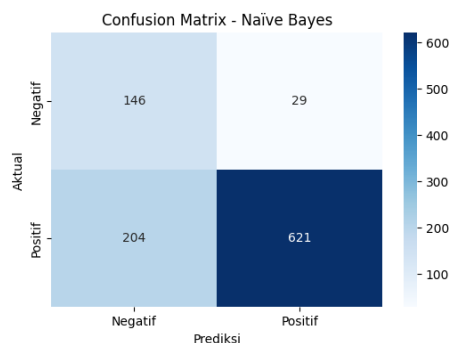
   negatif      0.42      0.83      0.56       175
   positif      0.96      0.75      0.84       825

 accuracy              0.77       1000
 macro avg              0.69      0.79      0.70       1000
 weighted avg          0.86      0.77      0.79       1000
    
```

Gambar 5. Laporan Klasifikasi *Naive Bayes*

Hasil pada Gambar 5 menunjukkan bahwa model *Naive Bayes* memperoleh tingkat akurasi sebesar 76,7% pada data uji dengan menggunakan representasi fitur TF-IDF. Berdasarkan hasil *classification report*, kinerja model pada kelas positif menunjukkan performa yang sangat baik, bernilai *precision* 0,96, *recall* 0,75, *F1-score* 0,84. Hal yang

mengindikasikan model dapat mengetahui ulasan positif tingkat ketepatan tinggi. Tetapi, kelas negatif, nilai *precision* relatif rendah (0,42) meskipun *recall*-nya cukup tinggi (0,83). Kondisi ini menunjukkan bahwa sebagian besar ulasan negatif dapat terdeteksi dengan baik, tetapi masih terdapat kesalahan klasifikasi terhadap beberapa ulasan positif yang teridentifikasi sebagai negatif. Nilai *macro average F1-score* 0,70 menandakan performa model cukup seimbang antar kelas, meskipun terdapat perbedaan jumlah data yang cukup mencolok.



Gambar 6. Confusion Marix Metode Naive Bayes

Berdasarkan hasil *confusion matrix* pada Gambar 6, dari 175 data ulasan negatif, sebanyak 146 data berhasil diklasifikasikan dengan benar, sedangkan 29 data keliru dikategorikan sebagai ulasan positif. Pada kelas positif yang berjumlah 825 data, sebanyak 621 data teridentifikasi secara tepat, sementara 204 data salah dikelompokkan jadi negatif. Temuan melihatkan model *Naive Bayes* terdapat kecenderungan lebih kuat mengenali ulasan positif dibandingkan ulasan negatif, kemungkinan dipicu ketidakseimbangan distribusi data (*imbalanced dataset*).

3.5.2 Algoritma Support Vector Machine

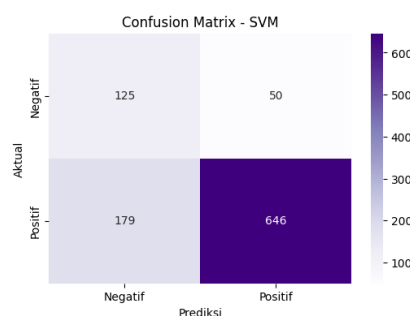
Untuk menilai performa model *Support Vector Machine* (SVM), dilakukan analisis menggunakan metrik evaluasi berupa *precision*, *recall*, dan *F1-score*. Ketiga metrik ini digunakan untuk mengukur sejauh mana model dapat memprediksi sentimen positif dan negatif secara akurat[22]. Hasil evaluasi tersebut ditampilkan pada gambar 7, yang menyajikan laporan klasifikasi lengkap sebagai dasar interpretasi performa SVM.

◆ Classification Report (SVM):

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| negatif | 0.41 | 0.71 | 0.52 | 175 |
| positif | 0.93 | 0.78 | 0.85 | 825 |
| accuracy | | | 0.77 | 1000 |
| macro avg | 0.67 | 0.75 | 0.69 | 1000 |
| weighted avg | 0.84 | 0.77 | 0.79 | 1000 |

Gambar 7. Laporan Klasifikasi Support Vector Machine

Hasil pada Gambar 7 menunjukkan, model *Support Vector Machine* (SVM) representasi fitur TF-IDF mencapai tingkat akurasi 77,1% data uji. Berdasarkan hasil *classification report*, kinerja model pada kelas negatif menunjukkan nilai *precision* 0,41, *recall* 0,71, *F1-score* 0,52. Sementara itu, kelas positif menunjukkan performa yang lebih baik *precision* 0,93, *recall* 0,78, *F1-score* 0,85. Dengan keseluruhan, nilai *macro average F1-score* sebesar 0,69 mengindikasikan model dapat melindungi keseimbangan performa antar kelas walaupun data yang digunakan tidak seimbang. Selain itu, nilai *weighted average F1-score* 0,79 konsisten dengan tingkat akurasi keseluruhan, yang menunjukkan stabilitas model dalam memprediksi kedua jenis sentimen, baik mayoritas maupun minoritas.



Gambar 8. Confusion Matrix Metode Support Vector Machine

Berdasarkan hasil *confusion matrix* pada Gambar 8, dari 175 data ulasan negatif, sebanyak 125 data sukses dikelompokkan secara benar, sementara 50 data keliru diperkirakan menjadi ulasan positif. Kelas positif yang berjumlah 825 data, sebanyak 646 data teridentifikasi dengan benar, sementara 179 data salah dikategorikan sebagai negatif. Temuan menandakan model SVM terdapat kompetensi cukup baik guna mengetahui ulasan positif, namun tetap mengalami kesulitan dalam membedakan ulasan negatif akibat adanya ketidakseimbangan data. Meskipun demikian, performa model ini tetap menunjukkan hasil yang lebih stabil serta nilai F1-score lebih baik dibanding metode *Naive Bayes*.

3.5.3 Algoritma *Random Forest*

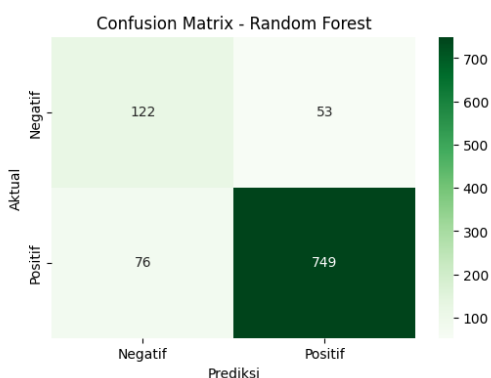
Untuk melihat performa model *Random Forest* secara lebih detail, evaluasi dilakukan menggunakan metrik *precision*, *recall*, dan *F1-score* pada masing-masing kelas. Hasil evaluasi tersebut disajikan pada laporan klasifikasi lengkap yang menjadi dasar analisis kemampuan model dalam mengklasifikasikan sentimen positif dan negatif, sebagaimana ditunjukkan pada Gambar 9.

◆ Classification Report (Random Forest):

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| negatif | 0.62 | 0.70 | 0.65 | 175 |
| positif | 0.93 | 0.91 | 0.92 | 825 |
| accuracy | | | 0.87 | 1000 |
| macro avg | 0.78 | 0.80 | 0.79 | 1000 |
| weighted avg | 0.88 | 0.87 | 0.87 | 1000 |

Gambar 9. Laporan Klasifikasi Algoritma *Random Forest*

Berdasarkan hasil pada Gambar 9, model *Random Forest* dengan representasi fitur TF-IDF menunjukkan kinerja paling unggul dibandingkan dengan model lainnya, dengan tingkat akurasi sebesar 87,1% pada data uji. Berdasarkan *hasil classification report*, kelas negatif terdapat nilai *precision* 0,62, *recall* 0,70, *F1-score* 0,65, sedangkan kelas positif melihatkan performa sangat baik *precision* 0,93, *recall* 0,91, *F1-score* 0,92. Nilai *macro average F1-score* 0,79 serta *weighted average F1-score* 0,87 menunjukkan model ini mampu mempertahankan keseimbangan performa antar kelas meskipun data yang digunakan tidak seimbang. Temuan menandakan *Random Forest* tidak hanya efektif mengenali sentimen mayoritas (positif), tetapi juga mampu meningkatkan ketepatan prediksi pada kelas minoritas (negatif).



Gambar 10. *Confusion Matrix* Metode *Random Forest*

Berdasarkan hasil *confusion matrix* pada Gambar 10, dari 175 data ulasan negatif, sebanyak 122 data berhasil diklasifikasikan secara benar, sementara 53 data keliru diperkirakan menjadi ulasan positif. Sedangkan dari 825 data ulasan positif, sebanyak 749 data teridentifikasi dengan benar, dan 76 data salah diklasifikasikan sebagai negatif. Hasil tersebut menunjukkan bahwa model *Random Forest* memiliki kemampuan generalisasi lebih baik dibandingkan SVM maupun *Naive Bayes*. Kombinasi antara proses *bagging* dan pemilihan fitur secara acak menjadikan model ini lebih tahan terhadap *overfitting* serta mampu mengolah variasi data teks dengan lebih efektif.

Keunggulan *Random Forest* ini tidak hanya tampak dari nilai akurasi dan *F1-score* yang lebih tinggi, tetapi juga dapat dijelaskan dari karakteristik algoritmanya serta kondisi dataset yang telah melalui proses SMOTE. Sebagai metode *ensemble* berbasis *bagging*, *Random Forest* cenderung lebih robust terhadap *noise* dan sampel sintesis hasil *oversampling*, sehingga mampu mempelajari pola dari kelas minoritas dengan lebih baik. Sebaliknya, SVM yang berbasis *kernel* lebih sensitif terhadap perubahan distribusi data sintesis dan dapat mengalami pergeseran margin pada data berdimensi tinggi seperti TF-IDF. Adapun *Naive Bayes* memiliki keterbatasan karena asumsi independensi fitur,

pada dataset berbasis TF-IDF karena mampu menangani variasi kata yang cukup kompleks. Dengan demikian, hasil penelitian ini konsisten dengan literatur yang relevan dan memperkuat temuan bahwa metode *ensemble* lebih unggul dalam tugas klasifikasi sentimen.

3.7.2 Analisis Efek SMOTE terhadap Peningkatan *Recall* Kelas Negatif

Penerapan SMOTE memberikan pengaruh signifikan terutama pada peningkatan *recall* kelas negatif. Kondisi awal menunjukkan bahwa data negatif jumlahnya jauh lebih sedikit, sehingga model cenderung bias dan sulit mengenali ulasan berisi keluhan. Setelah SMOTE diterapkan pada data latih, jumlah sampel negatif menjadi lebih seimbang dengan kelas positif, sehingga model mendapatkan representasi yang lebih baik untuk mempelajari pola keluhan pengguna. Hal ini terbukti dari peningkatan *recall* pada ketiga model, yang menunjukkan bahwa model semakin mampu mendeteksi ulasan negatif. Namun demikian, peningkatan *recall* ini biasanya diikuti sedikit penurunan *precision* karena kemunculan sampel sintesis yang menambah variasi data minoritas. Meskipun begitu, peningkatan *recall* lebih penting dalam konteks penelitian ini karena ulasan negatif merupakan sumber utama informasi untuk perbaikan layanan BRImo.

3.7.3 Alasan Teknis Mengapa *Random Forest* Paling Cocok

Random Forest menjadi algoritma paling unggul secara teknis karena karakteristiknya yang sesuai dengan kondisi dataset. Sebagai metode *ensemble* berbasis *bagging*, *Random Forest* lebih tahan terhadap *noise* dan sampel sintesis hasil SMOTE, berbeda dengan SVM yang sensitif terhadap perubahan margin akibat data sintesis. *Random Forest* juga melakukan pemilihan fitur secara acak (*random feature selection*), sehingga mengurangi risiko *overfitting* pada data berdimensi tinggi seperti TF-IDF. Selain itu, algoritma pohon keputusan di dalam *Random Forest* dapat menangkap hubungan antar kata yang bersifat non-linear, sedangkan *Naive Bayes* terbatas oleh asumsi independensi fitur sehingga tidak mampu mengenali hubungan kata dalam ulasan negatif. Kombinasi faktor ini membuat *Random Forest* lebih stabil dan akurat dibandingkan dua algoritma lainnya.

3.7.4 Keterkaitan Hasil Analisis Sentimen dengan Kebutuhan Pengembang BRImo

Hasil *wordcloud* sentimen negatif menunjukkan dominasi kata seperti “tidak”, “kenapa”, “saldo”, “error”, dan “login”. Kata-kata tersebut mengindikasikan bahwa sebagian besar keluhan pengguna bukan terkait penambahan fitur baru, melainkan masalah teknis yang mengganggu penggunaan aplikasi, seperti kendala *login*, saldo tidak tampil, transaksi gagal, dan *error* sistem. Temuan ini memberikan rekomendasi yang jelas bahwa pengembang BRImo perlu memprioritaskan stabilitas sistem, keandalan transaksi, serta perbaikan proses autentikasi, dibandingkan fokus pada pengembangan fitur baru. Dengan memanfaatkan pola keluhan ini, pengembang dapat menyesuaikan arah pengembangan aplikasi agar lebih responsif terhadap kebutuhan pengguna.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa algoritma *Random Forest* memberikan performa terbaik dalam klasifikasi sentimen ulasan BRImo. Selain mencapai akurasi 87%, model ini juga memperoleh *F1-Score* makro tertinggi yaitu 0,79, sehingga lebih andal dalam menangani data yang tidak seimbang dibandingkan SVM dan *Naive Bayes*. Keunggulan tersebut didukung oleh mekanisme *ensemble* berbasis *bagging* yang membuat *Random Forest* lebih stabil terhadap *noise* dan data sintesis hasil SMOTE. Setelah penerapan SMOTE, akurasi tidak lagi menjadi indikator utama untuk menilai keberhasilan model karena distribusi data menjadi lebih seimbang. Oleh karena itu, penggunaan metrik tambahan seperti *Recall* kelas negatif dan *F1-Score* makro menjadi penting untuk menunjukkan kemampuan model dalam mengenali ulasan negatif yang jumlahnya lebih sedikit. Nilai *F1-Score* makro 0,79 pada *Random Forest* memperkuat bahwa model ini tidak hanya unggul secara akurasi, tetapi juga efektif dalam mendeteksi kelas minoritas, sehingga hasil klasifikasinya lebih kredibel. Penerapan SMOTE terbukti membantu menyeimbangkan distribusi kelas pada data latih sehingga meningkatkan kemampuan model dalam mengenali ulasan negatif. Kombinasi antara pra-pemrosesan teks, pembobotan TF-IDF, dan teknik penyeimbangan data menjadikan model lebih efektif dalam menggambarkan persepsi pengguna. Dari sisi akademis, kontribusi utama penelitian ini adalah evaluasi komparatif tiga algoritma utama (NB, SVM, RF) yang dioptimalkan dengan SMOTE pada konteks aplikasi mobile banking BRImo, yang sebelumnya belum banyak dikaji secara mendalam. Secara praktis, temuan ini dapat membantu BRI untuk memprioritaskan peningkatan layanan terutama pada aspek *login*, saldo, dan stabilitas sistem sesuai pola keluhan pada ulasan negatif.

REFERENCES

- [1] N. F. Tsakila, M. A. Wirahadi, A. A. Fadilah, and H. Simanjuntak, “Analisis Dampak Fintech terhadap Kinerja dan Inovasi Perbankan di Era Ekonomi Digital,” *Indonesian Journal of Law and Justice*, vol. 1, no. 4, p. 11, Jun. 2024, doi: 10.47134/ijlj.v1i4.2787.
- [2] S. Nijjer, K. Sood, S. Grima, R. Rupeika-Apoga, and P. Varma, “Thematic Analysis of Financial Technology (Fintech) Influence on the Banking Industry,” *Risks*, vol. 10, p. 186, Sep. 2022, doi: 10.3390/risks10100186.



- [3] N. S. Elysa, L. Arini, D. F. Murad, and R. Leandros, “User Experience Satisfaction Analysis of Customers on the BRI Mobile Application (BRImo),” in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 680–689. doi: 10.1016/j.procs.2023.10.572.
- [4] Q. Zhou *et al.*, “A study on factors affecting service quality and loyalty intention in mobile banking A study on factors affecting service quality and loyalty intention in mobile,” *Journal of Retailing and Consumer Services*, vol. 60, May 2021, doi: 10.1016/j.jretconser.2020.102424.
- [5] Hani Harlan, “Perkembangan Financial Technology (Fintech) dalam Digital Perbankan di Indonesia,” *Al-Kharaj: Jurnal Ekonomi, Keuangan & Bisnis Syariah*, vol. 7, no. 5, May 2025, doi: 10.47467/alkharaj.v7i5.7529.
- [6] Yulia Nur Afifah, A. Kamilah, A. I. Nugroho, and A. A. Mustafidah, “Optimalisasi BRI Mobile Sebagai Bentuk Privilege Kepada Nasabah Di BRI Unit Puger,” *Journal Of Indonesian Social Society (JISS)*, vol. 2, no. 2, pp. 64–68, Jun. 2024, doi: 10.59435/jiss.v2i2.232.
- [7] N. F. Rokhmawati and A. B. Arifa, “Analisis User Interface (UI) pada BRIMO (BRI Mobile) menggunakan Pendekatan Metode Pengembangan System Usability Scale (SUS),” *Jurnal Ilmiah ILKOMINFO -Jurnal Ilmu Komputer dan Informatika*, vol. 7, p. 77, Jan. 2024, doi: 10.47324/ilkominfo.v7i1.233.
- [8] S. Rahayu and M. S. Hasibuan, “Sentiment Analysis of Mobile Banking Reviews Using Machine Learning Models,” *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, vol. 13, no. 1, pp. 69–78, Mar. 2025, doi: 10.33558/piksel.v13i1.10677.
- [9] S. A. S. Mola, D. L. B. Baun, I. O. Nunes, and M. M. A. R. Sani, “Analisis Sentimen Aplikasi Halo Bca Di Google Play Store Menggunakan Metode Naive Bayes, Support Vector Machine Dan Random Forest,” *HOAQ (High Education of Organization Archive Quality): Jurnal Teknologi Informasi*, vol. 15, no. 2, pp. 69–79, Dec. 2024, doi: 10.52972/hoaq.vol15no2.p69-79.
- [10] M. A. Khder, “Web scraping or web crawling: State of art, techniques, approaches and application,” *International Journal of Advances in Soft Computing and its Applications*, vol. 13, no. 3, pp. 144–168, 2021, doi: 10.15849/ijasca.211128.11.
- [11] W. Andriyani *et al.*, *PERANGKAT LUNAK DATA MINING*. Bandung: Widina Media Utama, 2024.
- [12] H. T. Duong and T. A. Nguyen-Thi, “A review: preprocessing techniques and data augmentation for sentiment analysis,” *Comput Soc Netw*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40649-020-00080-x.
- [13] M. A. Palomino and F. Aider, “Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis,” *Applied Sciences (Switzerland)*, vol. 12, no. 17, Sep. 2022, doi: 10.3390/app12178765.
- [14] A. A. Pratiwi and M. Kamayani, “Perbandingan Pelabelan Data dalam Analisis Sentimen Kurikulum Proyek di platform TikTok: Pendekatan Naive Bayes,” *Jurnal Eksplora Informatika*, vol. 14, no. 1, pp. 96–107, Sep. 2024, doi: 10.30864/eksplora.v14i1.1093.
- [15] V. R. Joseph, “Optimal ratio for data splitting,” *Stat Anal Data Min*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.
- [16] I. Irawan, Mh. Wathan, and M. Bagus Prayogi, “Analisis Sentimen Pada Aplikasi Capcut Di Google Play Store,” *Jurnal Pengembangan Sistem Informasi dan Informatika*, vol. 5, no. 4, pp. 152–164, Oct. 2024, doi: 10.47747/jpsii.v5i4.1959.
- [17] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” *Artif Intell Rev*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.
- [18] S. D. Parameswari *et al.*, “Studi Perbandingan Naive Bayes dan Support Vector Machine (SVM) dalam Analisis Sentimen Pengguna Metaverse,” *Jurnal Teknologi dan Manajemen Industri Terapan (JTMIT)*, vol. 4, no. 3, 2025, doi: 10.55826/jtmit.v4i3.1122.
- [19] N. Hendrastuty, A. Rahman Isnain, and A. Yanti Rahmadhani, “Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine,” *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*, vol. 6, no. 3, Sep. 2021, doi: 10.30591/jpit.v6i3.2870.
- [20] S. A. S. Mola, N. D. Rumlaklak, and D. P. N. Polly, *Analisis Sentimen dengan Metode Random Forest*. Bandung: Kaizen Media Publishing, 2024.
- [21] N. Pavitha *et al.*, “Movie recommendation and sentiment analysis using machine learning,” *Global Transitions Proceedings*, vol. 3, no. 1, pp. 279–284, Jun. 2022, doi: 10.1016/j.gltp.2022.03.012.
- [22] A. Roy and S. Chakraborty, “Support vector machine in structural reliability analysis: A review,” May 01, 2023, *Elsevier Ltd*. doi: 10.1016/j.res.2023.109126.