Terakreditasi SINTA Peringkat 3
Surat Keputusan Direktur Jenderal Pendidikan Tinggi, Riset, dan Teknologi Nomor 225/E/KPT/2022
masa berlaku mulai Vol.7 No. 1 tahun 2022 s.d Vol. 11 No. 2 tahun 2026

Terbit online pada laman web jurnal:
http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs

JOINTECS

JOINTECS
(Journal of Information Technology and Computer Science)

Vol. 9 No. 1 (2024) 01 - 10 e-ISSN:2541-6448 p-ISSN:2541-3619

Perbandingan Analisis Sentimen PLN Mobile: *Machine Learning* vs. *Deep Learning*

Ismail Akbar[1], Muhammad Faisal[2]
Magister Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang
[1]ismaelakbar12@gmail.com, [2]mfaisal@ti.uin-malang.ac.id

Abstract

Play Store app ratings hold significant value as they offer critical insights for app developers to enhance digital service quality. The research centers on the PLN Mobile app, which has garnered mixed user opinions since its launch. These reviews come with challenges for users and developers when interpreting user comments. This study conducts tests, comparing several machine learning algorithms: logistic regression, decision trees, random forests, and specific deep learning algorithms, including neural network multi-layer perceptron (MLP) and long short-term memory (LSTM) for sentiment classification, i.e., positive or negative. The study collected 3,000 PLN Mobile user reviews, comprising 1,965 positive and 1,035 negative reviews. Logistic regression achieved an 84.47% accuracy rate, decision trees scored 79.30%, and random forests reached 83.64%. In contrast, deep learning models, particularly the Neural Network Multilayer Perceptron (MLP), reached an accuracy rate of 84.47%, while the LSTM achieved an accuracy rate of 78.83%. In the context of sentiment analysis of PLN Mobile user reviews, machine learning models using the logistic regression algorithm and deep learning models employing the multi-layer perceptron (MLP) neural network algorithm demonstrated higher accuracy compared to other methods.

Keywords: sentiment analysis; machine learning; deep learning; PLN Mobile.

Abstrak

Rating ulasan aplikasi *play store* memiliki nilai strategis karena merupakan informasi penting bagi pengembang aplikasi untuk meningkatkan kualitas layanan di dunia digital. Salah satu aplikasi yang dijadikan subjek penelitian ini adalah PLN *Mobile*. Sejak diluncurkannya aplikasi PLN *Mobile*, terbukti masih banyak opini masyarakat yang tidak puas dengan penggunaan aplikasi PLN *Mobile*. Oleh karena itu, masih memiliki kelemahan bagi pengguna aplikasi dan pengembang aplikasi saat menganalisis komentar penulis pengguna. Dalam penelitian ini, dilakukan pengujian dengan membandingkan beberapa algoritma *machine learning* terdiri dari *logistic regression*, *decision tree*, *random forest* serta algoritma *deep learning* terdiri *neural network multi-layer perceptron* (MLP) dan *long short-term memory* (LSTM) untuk mengklasifikasikan senitmen positif atau negatif. Penelitian ini menghasilkan 3.000 ulasan pengguna aplikasi PLN *Mobile*, yang terdiri dari 1.965 ulasan positif dan 1.035 ulasan negatif. Data tersebut kemudian diuji dengan menggunakan model *logistic regression* yang memiliki akurasi sebesar 84,47%, *decision tree* yang memiliki akurasi sebesar 79,30%, dan *random forest* yang memiliki akurasi sebesar 83,64%. Sedangkan model algoritma *deep learning* khususnya *Neural Network Multilayer Perceptron* (MLP) memiliki akurasi sebesar 84,47%, sedangkan pengujian dengan *Long Short Term Memory* (LSTM) memberikan akurasi sebesar 78,83%. Berdasarkan penelitian analisis sentimen dalam ulasan pengguna aplikasi PLN *Mobile*, model *machine learning* yang menggunakan algoritma *logistic regression* dan model *deep learning* yang menggunakan algoritma *neural network multi-layer perceptron* (MLP) memiliki keunggulan dalam akurasi dibandingkan algoritma lainnya.

Kata kunci: analisis sentimen; *machine learning*; *deep learning*; PLN *Mobile*.

Diterima Redaksi : 17-10-2023 | Selesai Revisi : 13-03-2024 | Diterbitkan Online : 28-05-2024

1. Pendahuluan

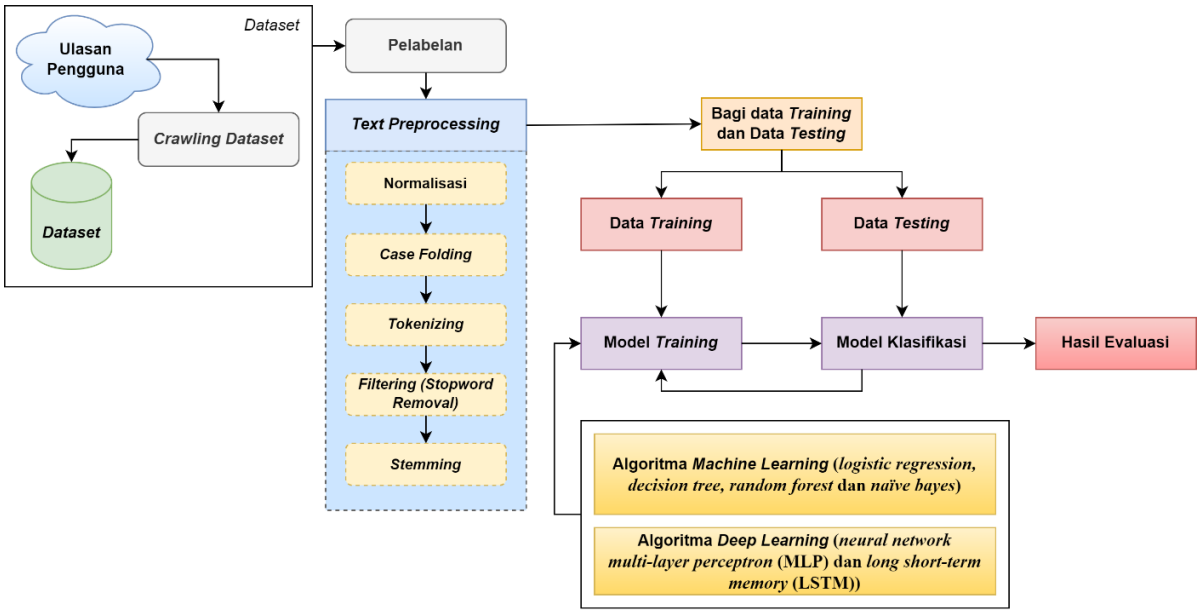
Di era teknologi yang semakin pesat, kebutuhan akan informasi sudah menjadi kebutuhan yang harus dipenuhi dengan cepat dan mudah. Bukan hanya sekedar informasi, akan tetap beberapa pelayanan harus mengalami peningkatan guna untuk memenuhi permintaan masyarakat yang mengharuskan pelayanan yang cepat dan efisien [1]. Hal ini beberapa perusahaan saling berlomba-lomba dalam meningkatkan kualitas pelayanan dengan memanfaatkan dunia digital, baik menggunakan layanan *website* ataupun *mobile*.

Perusahaan Listrik Negara (PLN) (Persero) yang menyediakan jasa dalam bidang tenaga listrik yang disalurkan kepada masyarakat. PLN terus melakukan berbagai terobosan inovasi untuk melayani masyarakat dalam dunia digital, agar tetap megikuti perkembangan zaman teknologi dengan memunculkan sebuah aplikasi berbasis *mobile* yakni Aplikasi PLN *Mobile*. Menurut Putri ayu Lestari, Aplikasi PLN *Mobile* merupakan suatu aplikasi yang terintegrasi dengan Aplikasi Pengaduan dan Keluhan Terpadu (APKT) dan Aplikasi Pelayanan Pelanggan Terpusat (AP2T) [2]. Aplikasi PLN *Mobile* dipilih sebagai objek penelitian ini karena setiap rumah di Indonesia menggunakan instalasi listrik khusus buatan PT. PLN (persero). Dengan banyaknya pengguna yang menggunakan instalasi PLN, tentunya dalam pekerjaan pelayanan dan pemeliharaan PLN juga perlu menyediakan layanan yang maksimal kepada masyarakat baik *offline* maupun *online* untuk menerima instalasi, perintah pemeliharaan, layanan perbaikan dan memberikan informasi tentang layanan yang diberikan.

Sejak peluncuran aplikasi PLN *Mobile* pada tahun 2016, hampir 10 juta telah diunduh oleh masyarakat, namun dilihat dari rating aplikasi tersebut, aplikasi ini masih memiliki banyak ruang untuk perbaikan. Hal ini terbukti dengan masih banyaknya opini masyarakat yang kurang puas dalam menggunakan aplikasi PLN Mobile. Ulasan pengguna biasanya memiliki dua bagian, yaitu rating (skor ulasan) dan komentar tertulis. Skor ulasan adalah representasi numerik dari keseluruhan pengalaman pengguna, namun ulasan teks dapat menceritakan kisah yang lebih detail. Namun ulasan pengguna di *Google Play Store* bisa memiliki nilai kumulatif yang sangat tinggi dalam waktu singkat. Oleh karena itu, jenis data ulasan ini memiliki beberapa kelemahan bagi pengguna aplikasi dan pengembang aplikasi saat menganalisis komentar penulis pengguna. Kerugian pertama adalah karena orang dapat dengan bebas mempublikasikan kontennya, kualitas opininya tidak dapat dijamin. Kerugian kedua adalah data evaluasi ini tidak selalu tersedia. Kebenaran dasar ibarat label suatu opini tertentu, yang menunjukkan apakah opini tersebut positif atau negatif. Untuk memahami review atau ulasan pengguna diperlukan analisis sentimen atau *opinion mining* yang merupakan salah satu cabang dari *Natural Language Processing* (NLP) yang mengukur dan memprediksi emosi pengguna [3]. Bidang keilmuan yang digunakan dalam pengembangan algoritma dan model komputasi dalam *Natural Language Processing* (NLP) termasuk *machine learning* dan *deep learning*.

Beberapa penelitian yang menggunakan bidang *machine learning* dalam analisis sentimen telah banyak diterapkan. Seperti yang dikemukakan oleh Youga Pratama dengan menggunakan algoritma *logistic regression* untuk analisis sentimen kendaraan listrik pada media sosial *twitter*, yang menghasilkan akurasi 87,9% dari 86,9% opini positif dan 13,1% opini negatif [4]. Namun menurut [5], penerapan *logistic regeression* masih memiliki kelemahan meski menghasilkan akurasi yang cukup baik, yaitu model yang dihasilkan mengalami *overfitting* karena skor uji yang dihasilkan memiliki gap yang cukup tinggi. Dalam penelitian lain yang menggunakan algoritma *decission tree* [6] memiliki hasil yang lebih baik dari *naïve bayes* dan *k-nearest neighbor* dengan akurasi dari *decission tree* mencapai 89,12% dari 1000 data ulasan, sedangkan hasil *naïve bayes* dari penelitian [6] hanya memperoleh akurasi 73,95%. Sedangkan pada penelitian [7] dengan jumlah data yang sama, hasil *naïve bayes* memperoleh akurasi 94,16%, pada penelitian yang dilakukan Evita Fitri, selain itu juga melakukan pengujian menggunakan *random forest* dengan akurasi 97,16%. dan *support vector machine* memiliki akurasi 96,01% [7].

Analisis sentimen pada bidang *deep learning* tidak kalah banyak juga yang telah melakukan penelitian dengan algoritma yang ada pada *deep learning*. Salah satunya yang dilakukan oleh Jelita Asian, dengan menggunakan algoritma *neural network multi-layer perceptron* (MLP) untuk analisis sentimen pendapat masyarakat terhadap kasus pelecahan seksual yang dilakukan oleh ahli aneestasi asal brazil, dengan menghasilkan nilai akurasi 94.44%. Akan tetap hasil algoritma *neural network multi-layer perceptron* (MLP) masih lebih baik dengan *random forest* yang menghasilkan akurasi 96,42% [8]. Hal ini dikarenakan mungkin karena dimensi *tweet* yang tinggi, jumlah teks yang sedikit dalam *tweet*, dan ukuran sampel. Pada penelitian lain yang menggunakan algoritma *long short-term memory* (LSTM) yang dilakukan oleh [9], menghasilkan nilai akurasi 97% untuk menganalisis komentar nitizen terhadap tayangan suatu acara di stasiun telivisi. Akan tetapi pada penelitian lain [10], hasil LSTM menghasilkan nilai akurasi sebesar 88% dari 1.511 dataset yang digunakan.



Gambar 1. Tahapan penelitian.

Tabel 1. Komentar ulasan pengguna PLN *Mobile*.

Index	Ulasan	Label
1	Gak jelas, dua kali belik token melalui Aplikasi ke BRI tp tokennya di PLN mobile tidak keluar. ...	0
2	Aplikasi sangat membantu, mudah dan cepat membuat pelaporan. Praktis, simpel dan hemat waktu karena bisa diakses kapanpun dan dimanapun.	1
3	Aplikasi sudah bagus, pas ada perbaikan di daerah saya ternyata dapat notif …	1
4	Untuk layanan dilapangan cepat tanggap, tapi aplikasinya masih banyak error	0

Dengan demikian, pada penelitian ini akan melakukan pengujian dengan membandingkan beberapa algoritma *machine learning* (*logistic regression, decision tree, random forest* dan *naïve bayes*) dan *deep learning* (*neural network multi-layer perceptron* (MLP) dan *long short-term memory* (LSTM)). Dengan data ulasan aplikasi PLN mobile yang diambil pada *Google Play Store*. Sehingga diharapkan memperoleh metode yang paling baik dalam melakukan analisis sentimen.

2. Metode Penelitian

Ada beberapa langkah yang harus dilakukan untuk mendapatkan hasil terbaik dalam penelitian ini. Pengumpulan dan pelabelan data merupakan proses yang perlu dilakukan terlebih dahulu, dilanjutkan dengan tahap pra-pemrosesan. Algoritma *machine learning* dan *deep learning* akan digunakan untuk mengklasifikasikan data yang diproses sebelumnya. Langkah-langkah yang dilakukan dalam penelitian ini digambarkan pada Gambar 1.

2.1. Pengumpulan Data

Data untuk penelitian ini diperoleh melalui pengambilan ulasan pengguna aplikasi PLN Mobile di Google Play Store menggunakan teknik crawling. Proses ini menghasilkan sebanyak 3.000 data ulasan yang mencakup periode waktu dari tahun 2018 hingga 2023. Pengumpulan data dilakukan secara menyeluruh untuk memastikan representasi yang akurat dari beragam pengguna dan pengalaman menggunakan aplikasi PLN Mobile selama periode tersebut.

2.2. Pelabelan

Pada penelitian ini, proses pelabelan setiap kalimat menggunakan metode berbasis kosakata dengan menggunakan *Vader Sentiment Library*. Menurut Taboada, dkk pada penelitian Nadhif Sanggara Fathullah menyatakan *Lexicon Based* adalah proses pemilihan kata-kata penting dari suatu dokumen berdasarkan kamus/kosakata yang ada. Dalam penerapannya terdapat dua kamus yang digunakan untuk menjadi daftar kata. Kamus berisi kelompok kata dengan positif dan kamus berisi kelompok kata dengan negatif [11]. Sedangkan VADER (*Valence Aware Dictionary and Sentiment Reasoner*) adalah metode analisis berbasis kosakata. VADER akan mengurai teks dari kosakata (*a library*) yang menciptakan kelas sentimen sebagai positif, negatif, dan netral dengan menambahkan skor total atau skor gabungan [12]. ulasan pengguna aplikasi PLN *Mobile* telah dikategorikan menjadi dua label. Label 1 menandakan bahwa komentar ulasan pengguna berkomentar positif. Sedangkan label 0 menandakan komentar ulasan pengguna berkomentar negatif. Contoh data pada setiap label terlihat pada Tabel 1.

2.3. *Preprocessing*

Langkah selanjutnya adalah data *preprocessing* dengan tujuan mendapatkan data ulasan pengguna yang jelas dan terstruktur sehingga mendapatkan hasil klasifikasi sentimen yang lebih akurat. Proses yang digunakan dalam *preprocessing* meliputi normalisasi untuk mengubah beberapa baris menjadi satu baris. Kemudian proses *case folding*, yaitu proses penyeragaman huruf besar diubah menjadi huruf kecil dan menghilangkan angka serta tanda baca [13]. Kemudian tokenisasi digunakan untuk memisahkan kalimat menjadi kata [14]. Proses keempat adalah *filtering* untuk menghilangkan kata-kata yang tidak penting [15]. Proses terakhir yaitu *stemming* digunakan untuk mengubah kata menjadi kata dasar [16].

2.4. Pembagian Data *Training* dan Data *Testing*

Data latih (*training*) digunakan untuk melatih model klasifikasi, model ini merupakan representasi pengetahuan yang akan digunakan untuk memprediksi kelas data baru yang belum ada, semakin banyak data latih yang digunakan semakin baik mesin memahami pola data tersebut. Pada saat yang sama, data uji (*testing*) digunakan untuk mengukur seberapa baik kinerja pengklasifikasi. Data yang digunakan untuk data latih dan data uji adalah data yang mempunyai label kelas. Dalam penelitian ini, kelas dibagi menjadi dua, yaitu positif dan negatif. Jumlah data latih dan data uji memiliki perbandingan 80%:20% seperti terlihat pada Tabel 2.

Tabel 2. Perbandingan data latih dan data uji.

Kelas	Data latih	Data uji
Positif	1586	379
Negatif	806	220

2.5. *Logistic Regression*

Logistic regression adalah bagian dari metode penambangan data yang digunakan untuk menganalisis data yang menggambarkan suatu variabel respon (dependen) atau beberapa variabel *predictor* [17]. Regresi logistik dapat bekerja dengan baik dalam menangani hubungan linier antar variabel [18]. Regresi logistik merupakan metode klasifikasi yang sangat cocok untuk data dengan dua label, positif dan negatif, namun metode ini tetap dapat digunakan untuk data dengan banyak label atau dua label [19]. Rumus dari *logistic regression* dapat dilihat pada rumus 1 [20].

$$W = b + w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n \quad (1)$$

$$P = \frac{1}{1+e^{-(w)}} \quad (2)$$

Pada rumus 1 nilai *b* adalah *intercept* atau bias, yang merupakan konstanta. Nilai W_1, W_2 ... W_n adalah koefisien atau bobot yang sesuai dengan masing-masing fitur (atribut) X_1, X_2, X_3, ..., X_n. Setiap fitur memiliki bobot yang mempengaruhi sejauh mana fitur tersebut memengaruhi nilai *W* atau probabilitas kelas positif. Nilai X_1, X_2, X_3, ..., X_n adalah nilai-nilai fitur atau atribut yang digunakan dalam prediksi. Nilai *w* yang dihasilkan kemudian dipetakan menggunakan fungsi *logistic regression* seperti rumus 2.

2.6. *Decision Tree*

Decision tree (pohon keputusan) adalah metode yang mengorganisir atribut untuk memprediksi hasil. Di dalamnya, setiap node internal menguji atribut, dengan cabang yang mewakili hasil tes dan node yang menunjukkan kelas. Simpul paling atas merupakan akar, dipilih berdasarkan gain tertinggi atau entropi terendah dari atribut. Proses ini dimulai dengan menghitung entropi menggunakan rumus entropi pada rumus 3, kemudian menentukan gain dengan rumus gain yang disajikan pada rumus 4. Pendekatan ini memfasilitasi pemahaman tentang bagaimana atribut berkontribusi terhadap keputusan, mengoptimalkan pemilihan atribut untuk prediksi yang lebih tepat dan efisien dalam struktur pohon keputusan. [21].

$$Entropy\ (S) = \sum_{i=0}^{n} - \pi * log^2\ \pi \quad (3)$$

$$Gain\ (S,A) = Entropy(S) = \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy\ (Si) \quad (4)$$

Dari rumus 3, rumus 4 dapat diketahui *Gain (S,A)* adalah *information gain* yang ingin kita hitung. Ini mengukur sejauh mana pemilihan atribut (A) akan mengurangi ketidakpastian (*entropi*) dalam *dataset* (S) ketika dibagi menjadi *subset* (Si). *Entropy (S)* mengukur tingkat ketidakpastian atau kekacauan dalam dataset. Semakin tinggi entropi, semakin banyak ketidakpastian. Σ (*i=1 to n*) digunakan untuk menjumlahkan kontribusi dari setiap subset (Si) yang dihasilkan saat kita membagi dataset (S) dengan menggunakan atribut (A). |Si| / |S| merupakan fraksi atau rasio ukuran subset (Si) terhadap ukuran dataset awal (S). Ini mengukur sejauh mana subset ini menyumbang terhadap dataset keseluruhan. *Entropi (Si)* merupakan entropi dari subset (Si) yang dihasilkan saat kita membagi dataset menggunakan atribut (A). Entropi ini mengukur ketidakpastian dalam subset tersebut setelah pembagian.

2.7. *Random Forest*

Algoritma *random forest* adalah sekumpulan metode pembelajaran yang menggunakan pohon keputusan sebagai pengklasifikasi dasar yang membangun, dan menggabungkan, beberapa aspek penting dari metode *random forest*, termasuk melakukan pengambilan sampel terbimbing untuk membangun pohon prediksi. Setiap pohon keputusan menggunakan prediktor acak, dan hutan acak itu sendiri membuat prediksi dengan menggabungkan hasil dari setiap pohon keputusan menggunakan pemungutan suara mayoritas untuk klasifikasi dan rata-rata untuk regresi. *Random forest* memiliki kinerja yang baik dengan akurasi tinggi, tahan terhadap outlier dan kebisingan, serta lebih cepat dibandingkan *bagging* dan *boosting* [22].

2.8. *Neural Network Multi-Layer Perceptron*

Perceptron dikenal sebagai jaringan saraf yang diperkenalkan pada tahun 1950-an, *Multi-Layer Perceptron* (MLP) adalah jaringan saraf yang disebut perceptron. *Neuron* disusun secara hierarki menjadi beberapa lapisan yang terhubung. Jaringan MLP dimulai dengan lapisan masukan, diikuti oleh lapisan tersembunyi, dan diakhiri dengan lapisan keluaran. Lapisan tersembunyi menyediakan pemrosesan komputasi dalam jaringan untuk menghasilkan keluaran jaringan [23]. *Perceptron* tergolong linier, artinya digunakan untuk memisahkan kategori menggunakan garis lurus. *Input*-nya biasanya berupa vektor *x*, yang kemudian dikalikan dengan bobot *w* dan ditambahkan ke bias atau *y* pada rumus 5, rumus 6 dan rumus 7.

$$y = w \,.\, x + b \tag{5}$$

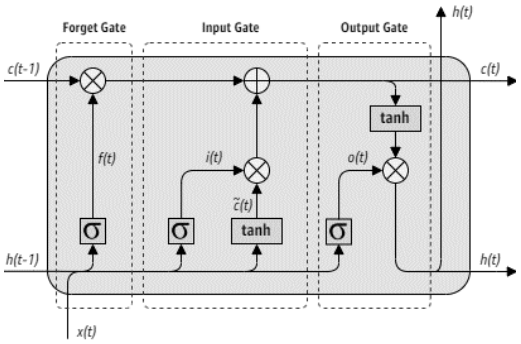
$$y = \varphi \left(\sum_{i=1}^{n} w_1 x_1 + b\right) \tag{6}$$

$$y = \varphi \left(w^T x + b\right) \tag{7}$$

Dalam rumus rumus 5, rumus 6 dan rumus 7, *y* adalah rumus bias, *w* adalah bobot, *x* adalah vektor masukan, *b* adalah bias, *phi* adalah fungsi aktivasi *nonlinier*. Dalam implementasi algoritma *multi-layer perceptron* untuk tugas klasifikasi, kita menggunakan metode *backpropagation*. Metode ini membantu model dalam proses pembelajaran dan penyesuaian bobot dan bias pada jaringan saraf tiruan yang memiliki beberapa lapisan [24].

2.9. *Long Short-Term Memory*

Long Short-Term Memory (LSTM) baru-baru ini menjadi teknik penting di kalangan peneliti *Natural Language Processing*. LSTM bertujuan untuk memecahkan masalah *vanishing gradient* pada RNN (*Recurrent Neural Network*) dimana ketika memproses data sekuensial panjang, fungsi gradien mengalami penurunan secara eksponensial yang mengakibatkan kerugian [25], sehingga mengurangi prediksi kinerja [26]. Setiap unit dalam jaringan LSTM memiliki komponen penting yang disebut sel memori. Dalam Gambar 2, kita dapat melihat bahwa pada waktu *t*, keadaan dari unit LSTM direpresentasikan sebagai *ct*. *Sigmoid gate* memainkan peran kunci dalam mengontrol pembacaan dan modifikasi sel memori, dan memengaruhi tiga hal utama: *input gate* (*it*), *forget gate* (*ft*), dan *output gate* (*Ot*). Pada titik ini, model menerima informasi dari dua sumber eksternal, yaitu *ht-1* (keadaan tersembunyi sebelumnya) dan *xt* (vektor input pada waktu (*t*). Untuk menghitung keadaan tersembunyi *ht* pada waktu *t* kita menggunakan informasi dari *input gate, output gate, forget gate,* dan juga *xt*. Semua komponen ini berkontribusi pada perhitungan keadaan *node* lapisan tersembunyi dalam jaringan LSTM. Sehingga melalui mekanisme ini, LSTM dapat mempertahankan informasi relevan dan membuang yang tidak perlu, memungkinkan prediksi yang lebih akurat dalam urutan data yang panjang.



Gambar 2. Arsitektur *Long Short-Term Memory*.

2.10. Evaluasi Kinerja Model

Untuk menghitung tingkat prediksi yang benar dan salah serta memahami jenis kesalahan, dapat digunakan untuk mengukur kinerja model menggunakan *confusion matrix*. *confusion matrix* adalah metode mengukur dan mengevaluasi keakuratan model [27]. Berdasarkan Tabel 3 TP mewakili hasil positif sebenarnya, yaitu jumlah sampel positif yang diprediksi dengan benar; TN mewakili hasil negatif sebenarnya, yaitu jumlah sampel negatif yang diprediksi dengan benar; FP mewakili positif palsu, yaitu jumlah sampel negatif yang salah diprediksi sebagai positif; dan FN mewakili hasil negatif palsu, yaitu jumlah sampel positif yang salah diprediksi menjadi negatif [28]. Untuk menghitung keakuratan prediksi yang benar bisa menggunakan rumus 8. Dengan TP menunjukkan nilai positif sebenarnya ditambah TN yang merupakan nilai sentimen benar dibagi total TP ditambah TN ditambah FP ditambah FN. *Precison* digunakan untuk mengukur sampel kelas positif yang diklasifikasikan dengan benar dan didefinisikan dalam rumus 9, TP dan FP menunjukkan jumlah positif benar dan positif salah. *Recall* pada rumus 10 digunakan untuk menghitung seluruh sampel positif, TP sebagai positif sebenarnya, dan dibagi dengan total TP ditambah FN.

Tabel 3. *Confusion matrix*.

	Kelas Aktual	
Kelas Prediksi	TP	FN
	FP	TN

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{8}$$

$$precison = \frac{TP}{TP+FP} \tag{9}$$

$$recall = \frac{TP}{TP+FN} \tag{10}$$

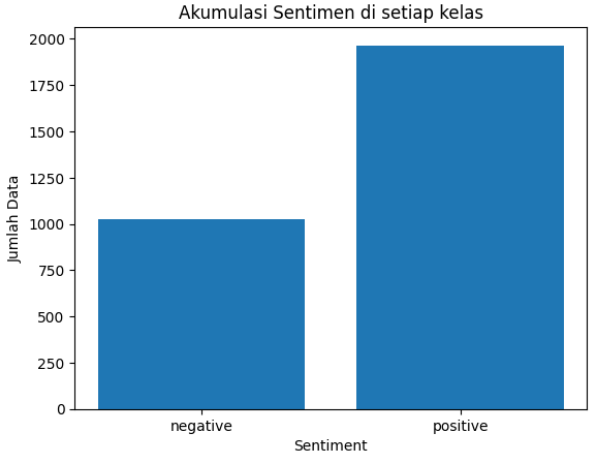
3. Hasil dan Pembahasan

Pada penelitian ini, dilakukan pengujian menggunakan beberapa model *machine learning* dan *deep learning*. Model *machine learning* yang digunakan meliputi *logistic regression*, *decision tree*, dan *random forest*. Selain itu, dalam rangka meningkatkan kompleksitas dan kemampuan model, kami juga menggunakan model *deep learning* seperti neural *network multi-layer*

perceptron dan *long short-term memory*. Dengan mengintegrasikan kedua jenis model ini, kami dapat mengamati dan membandingkan kinerja serta karakteristik masing-masing model dalam memprediksi variabel target. Dengan demikian, penelitian ini membandingkan pendekatan *machine learning* dan *deep learning* untuk memberikan pemahaman yang lebih komprehensif tentang masalah yang diteliti serta memberikan gambaran yang lebih lengkap tentang keunggulan dan kelemahan dari masing-masing model yang digunakan.

3.1. Pengumpulan Data

Pada tahap pengumpulan data ini, dilakukan proses pengambilan data dari ulasan pengguna aplikasi PLN *Mobile* pada situs Google Play Store menggunakan teknik *web scraping* dengan bahasa pemrograman *python*. Proses pengumpulan data dilakukan dalam rentang waktu antara tahun 2018 hingga tahun 2023, dan berhasil mengumpulkan total 3.000 data ulasan pengguna. Data yang diambil mencakup ulasan-ulasan dengan berbagai tingkat rating, mulai dari bintang 1 hingga 5. Selanjutnya, data-data ini diberi label sentimen menggunakan metode *Vader Lexicon*, yang mengelompokkan ulasan-ulasan tersebut menjadi dua kategori berdasarkan Gambar 3, yaitu sentimen positif dengan jumlah 1.965 data ulasan dan sentimen negatif dengan jumlah 1.035 data ulasan.



Gambar 3. Ulasan pengguna berdasarkan label.

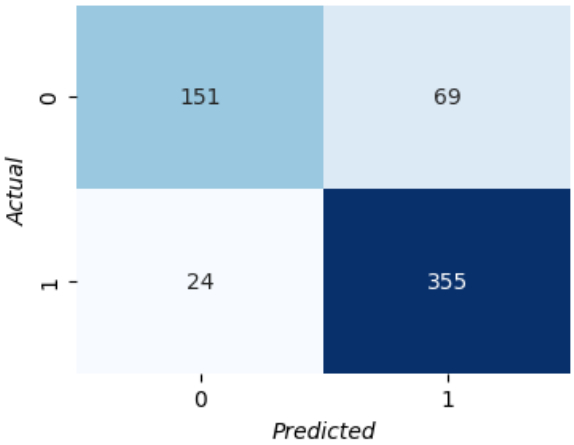
3.2. Pengujian menggunakan *Logistic Regression*

Pengujian pertama dalam penelitian ini menggunakan model *machine learning* dengan algoritma *logistic regression*. Berdasarkan pembagaian data yang ada pada tabel 2 dan hasil pengujian menggunakan *logistic regression* pada Tabel 4, mendapatkan *accuracy* model pada data latih sebesar 90.83%, *precision* 91.00% dan *recall* 90.00%. Sedangkan *accuracy* model pada data uji 84.47%, *precision* 85.00% dan *recall* 81.00%.

Dari Gambar 4, kita dapat melihat hasil dari *confusion matrix* yang dihasilkan dari pengujian dengan model *logistic regression*. Terdapat 355 ulasan positif yang telah diprediksi dengan benar, yang berarti model berhasil mengidentifikasi ulasan positif tersebut. Namun, ada 24 ulasan positif yang diprediksi dengan salah oleh model. Di sisi lain, ada 151 ulasan negatif yang telah diprediksi dengan benar oleh model, menunjukkan kemampuan model dalam mengenali ulasan negatif. Namun, terdapat 69 ulasan negatif yang diprediksi dengan salah oleh model. Dengan kata lain, model *logistic regression* berhasil memprediksi dengan benar sebagian besar ulasan positif dan negatif, namun terdapat beberapa kesalahan dalam prediksi ulasan.

Tabel 4. Hasil Pengujian *Logistic Regression*.

Data	*Accuracy*	*Precision*	*Recall*
Data latih	90.83%	91.00%	90.00%
Data Uji	84.47%	85.00%	81.00%



Gambar 4. *Confusion matrix logistic regression*.

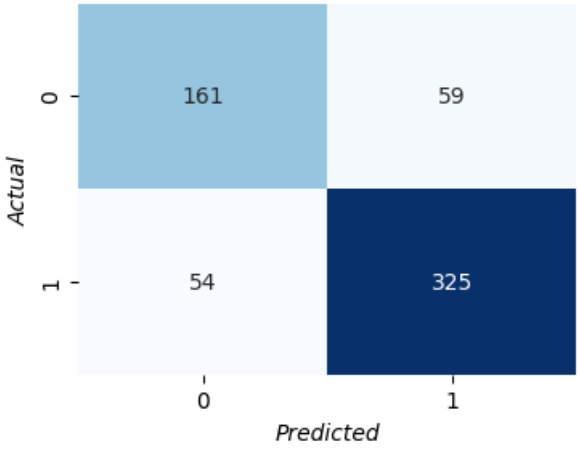
3.3. Pengujian menggunakan *Decision Tree*

Pengujian menggunakan algoritma *Decision Tree*, sebagaimana terdokumentasi pada Tabel 5, menunjukkan bahwa model ini berhasil mencapai tingkat akurasi sebesar 85.00%, presisi 86.00%, dan recall 86.00% pada data latih. Hasil ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mengklasifikasikan data latih dengan tepat, memberikan prediksi yang konsisten dan akurat. Namun, saat dihadapkan pada data uji, performa model menunjukkan sedikit penurunan dengan tingkat akurasi sebesar 79.30%, presisi 78.00%, dan *recall* 78.00%. Meskipun demikian, mesin pembelajaran ini masih mampu memberikan prediksi yang layak dengan tingkat keakuratan yang tetap dalam kisaran yang dapat diterima. Dalam konteks ini, meskipun terdapat penurunan performa pada data uji, model *Decision Tree* tetap dapat dianggap efektif dalam menghadapi variasi data yang berbeda dengan hasil yang cukup memuaskan.

Tabel 5. Hasil Pengujian *Decision Tree*.

Data	*Accuracy*	*Precision*	*Recall*
Data latih	85.00%	86.00%	86.00%
Data Uji	79.30%	78.00%	78.00%

Hasil dari *confusion matrix* pada Gambar 5 adalah hasil dari pengujian model *decision tree*. Dalam hasil ini, terdapat 325 ulasan positif yang berhasil diidentifikasi secara benar oleh model, yang menunjukkan bahwa model sangat baik dalam mengenali ulasan positif tersebut. Namun, perlu diperhatikan bahwa terdapat 54 ulasan positif yang diprediksi dengan salah oleh model, artinya ada beberapa kasus positif yang model tidak berhasil mengenali. Di sisi lain, model juga memiliki keahlian dalam mengenali ulasan negatif, yang ditunjukkan oleh 161 ulasan negatif yang telah diprediksi dengan benar. Ini mengindikasikan kemampuan model untuk mengidentifikasi ulasan negatif. Namun, ada 59 ulasan negatif yang diprediksi dengan salah oleh model, artinya ada beberapa ulasan negatif yang pada kenyataannya merupakan positif menurut model.



Gambar 5. *Confusion matrix decision tree*.

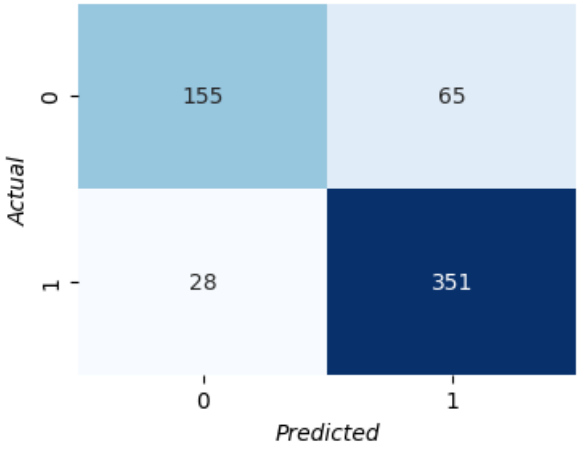
3.4 Pengujian menggunakan *Random Forest*

Pengujian selanjutnya pada penelitian ini menerapkan model *machine learning* dengan algoritma *Random Forest*. Hasil percobaan menggunakan algoritma ini, sebagaimana tercatat dalam Tabel 6, mengungkapkan performa yang menjanjikan. Pada data latih, model mencapai tingkat akurasi sebesar 90,53%, dengan nilai *precision* dan *recall* masing-masing mencapai 91,06%. Namun demikian, ketika diterapkan pada data uji, performa model sedikit menurun dengan akurasi sebesar 83,64%, dan precision serta *recall* berturut-turut sebesar 83,00% dan 81,00%. Meskipun demikian, hasil ini tetap menunjukkan kecenderungan model untuk mempertahankan performa yang baik meskipun pada data uji yang berbeda.

Tabel 6. Hasil Pengujian *Random Forest*.

Data	*Accuracy*	*Precision*	*Recall*
Data latih	90.53%	91.06%	91.06%
Data Uji	83.64%	83.00%	81.00%

Dengan mengukur seberapa baik model *random forest* mampu membedakan hasil positif dan negatif, hal ini dapat dilihat pada hasil *confusion matrix* yang ditunjukkan pada Gambar 6. Dalam hasil tersebut, 351 ulasan positif diidentifikasi dengan benar oleh model, yang menunjukkan bahwa model ini sangat efektif dalam menangkap ulasan positif. Namun, perlu diperhatikan bahwa 28 ulasan positif salah diprediksi oleh model, yang berarti model tersebut gagal menangkap beberapa kasus positif. Di sisi lain, model tersebut juga memiliki keahlian dalam mengenali ulasan negatif, seperti yang ditunjukkan oleh 155 ulasan negatif yang diprediksi dengan tepat. Hal ini menunjukkan kemampuan model untuk mengidentifikasi ulasan negatif. Namun, 65 ulasan negatif diprediksi secara keliru oleh model tersebut.



Gambar 6. *Confusion matrix random forest*.

3.5 Pengujian menggunakan *Multi-Layer Perceptron*

Pengujian menggunakan model *deep learning* melibatkan penerapan algoritma *neural network multi-layer perceptron* (MLP) sebagai salah satu metode evaluasi. Dari hasil percobaan yang terdokumentasi dalam Tabel 7, diperoleh sejumlah metrik evaluasi kinerja model. Pada tahap ini, model berhasil mencapai tingkat akurasi sebesar 92,36% terhadap data latih, dengan nilai *precision* mencapai 93,26% dan *recall* mencapai 92,54%. Namun demikian, ketika diuji menggunakan data uji, performa model menurun menjadi 84,47% akurasi. Hal ini mengindikasikan adanya perbedaan kinerja yang signifikan antara data latih dan data uji, yang mungkin disebabkan oleh *overfitting* atau variasi dalam *dataset*.

Tabel 7. Hasil Pengujian *Multi-Layer Perceptron*.

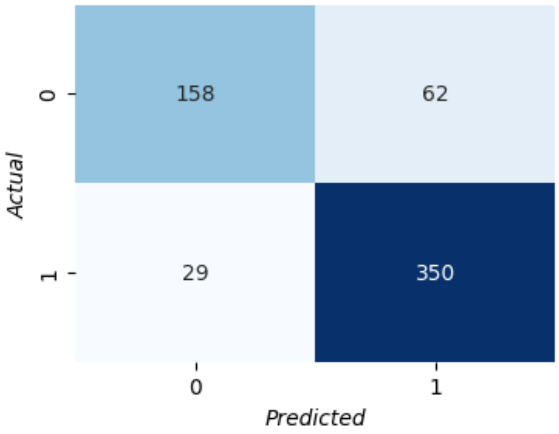
Data	*Accuracy*	*Precision*	*Recall*
Data latih	92.36%	93.26%	92.54%
Data Uji	84.47%	84.00%	82.00%

Hasil *confusion matrix* dari algoritma *neural network multi-layer perceptron* (MLP) ditampilkan pada Gambar 7. Terdapat 350 ulasan positif yang telah diprediksi dengan benar, yang berarti model berhasil mengidentifikasi ulasan positif tersebut. Namun, ada 29 ulasan positif yang diprediksi dengan salah oleh model. Kemudian terdapat 158 ulasan negatif yang telah diprediksi dengan benar oleh model, menunjukkan kemampuan model dalam mengenali ulasan negatif.

Akan tetapi, terdapat 62 ulasan negatif yang diprediksi dengan salah oleh model.

3.6 Pengujian menggunakan *Long Short-Term Memory*

Percobaan berikut menerapkan arsitektur *Long Short-Term Memory* yang ditunjukkan pada Gambar 2. Kemudian, kumpulan data teks hasil *preprocessing* dibagi menjadi tiga bagian. Pertama, dataset untuk data latih yaitu 80% dan sisanya untuk data uji. Data pelatihan kemudian dibagi dua untuk data validasi 75%. Setelah berhasil membagi data, selanjutnya adalah proses pemberian nilai vektor pada setiap kata. Kemudian, di dalam daftar token kata diubah menjadi rangkaian angka dengan mengganti indeks kata dengan nilai *integer*.



Gambar 7. *Confusion matrix multi-layer perceptron*.

Tabel 8. Parameter *Long Short-Term Memory*.

Nama Parameter	*Nilai Parameter*
Embedding Size	32
Optimizer	Adam
Activation	Sigmoid
Batch Size	64

Tabel 9. Hasil Pengujian *Long Short-Term Memory*.

Data	*Accuracy*	*Precision*	*Recall*
Data latih	95.50%	97.75%	95.60%
Data Validasi	80.00%	90.75%	81.39%
Data Uji	78.83%	89.81%	79.01%

Pada pengujian dengan algoritma *Long Short-Term Memory*, parameter yang digunakan diatur sesuai Tabel 8. Kemudian pada tahap pengujian ditentukan jumlah *epoch* sebanyak 20 kali untuk setiap pengujian. Dari data uji latih, data validasi, dan data uji diperoleh nilai *accuracy*, *precision*, dan *recall* sesuai Tabel 9.

3.7 Perbandingan Penelitian Terdahulu

Dalam penelitian ini melihat dan menganalisis temuan dan metode yang telah diusulkan dalam penelitian sebelumnya yang relevan dengan topik yang sedang dibahas. Perbandingan dengan penelitian terdahulu merupakan langkah penting untuk mengevaluasi kontribusi baru yang ditawarkan oleh penelitian ini dan memahami posisi relatifnya dalam konteks pengetahuan yang sudah ada. Melalui perbandingan ini, dapat diidentifikasi keunggulan, kelemahan, dan kontribusi unik dari penelitian ini dalam memperluas atau meningkatkan pemahaman kita tentang topik tertentu. Seperti yang ada pada Tabel 10.

Tabel 10. Perbandingan Penelitian Terdahulu.

Reference	*Algoritma*	*Matrix Performance*		
		Accuracy	*Precision*	*Recall*
Penulis	LR	84.47%	85.00%	81.00%
	DT	79.30%	78.00%	78.00%
	RF	83.64%	83.00%	81.00%
	MLP	84.47%	84.00%	82.00%
	LSTM	78.83%	89.81%	79.01%
[4]	LR	87.90%	-	-
	LR+PCA	90.00%	-	-
[6]	KNN	85.03%	84.98%	100%
	NB	73.95%	100%	69.26%
	DT	89.12%	88.62%	100%
[29]	NB	77.69%	59.84%	53,14%
	KNN	76.40%	56.84%	49.64%
[30]	SVM	83.75%	-	-

Tabel 10 di atas menyajikan hasil kinerja beberapa algoritma *machine learning* maupun *deep learning* yang telah diuji dalam penelitian, serta hasil yang dilaporkan oleh beberapa referensi lainnya. Setiap baris mewakili satu algoritma, dengan kolom-kolom yang menunjukkan metrik evaluasi kinerja seperti akurasi, presisi, dan *recall*. Hasil kinerja beberapa algoritma yang ada dipenelitian ini, di antaranya adalah *Logistic Regression* (LR), *Decision Tree* (DT), *Random Forest* (RF), *Multi-Layer Perceptron* (MLP), dan *Long Short-Term Memory* (LSTM). Hasil yang ditampilkan adalah rata-rata kinerja dari model yang dihasilkan. Dari hasil tersebut, dapat dilihat bahwa penelitian ini mencatat akurasi tertinggi pada algoritma *Logistic Regression* (84.47%) dan MLP (84.47%), serta presisi tertinggi pada LSTM (89.81%). Bila dibandingkan dengan data dari referensi lain, tabel menunjukkan adanya variasi pada hasil. Misalnya, referensi [4] hanya menyediakan nilai untuk *Logistic Regression* (LR) dengan akurasi sebesar 87.90%. Sementara itu, referensi [6] menampilkan kinerja yang cukup impresif dari *K-Nearest Neighbors* (KNN) dengan presisi 84.98% dan *recall* 100%, serta *Naive Bayes* (NB) dengan *recall* 69.26%. Referensi [29] dan [30] memberikan gambaran tentang kinerja KNN dan *Support Vector Machine* (SVM) dengan variasi nilai yang berbeda. Dengan mengamati nilai-nilai akurasi dan presisi yang tinggi untuk algoritma *Logistic Regression* dan *Multi-Layer Perceptron*, serta presisi yang sangat baik untuk LSTM yang tercatat dalam penelitian ini, bahwa model-model tersebut menawarkan hasil yang konsisten dan berdaya saing tinggi. Hasil ini menandakan bahwa penelitian ini berhasil mengembangkan atau menerapkan model-model dengan kemampuan prediktif yang kuat, yang mungkin memiliki potensi untuk memberikan wawasan

yang lebih akurat dalam berbagai aplikasi praktis dari pembelajaran mesin. Namun demikian, penelitian ini memiliki kekurangan dalam hal data *recall* yang tidak disajikan, yang membuat kita tidak dapat mengevaluasi sepenuhnya seberapa baik model-model ini dalam menemukan semua kasus positif yang relevan dalam dataset. *Recall* adalah metrik penting, terutama dalam kasus di mana konsekuensi dari melewatkan kasus positif bisa signifikan, seperti dalam pengaturan medis atau keamanan. Tanpa data ini, kita tidak dapat membandingkan secara langsung dengan hasil recall dari referensi lain, sehingga memberikan gambaran yang tidak lengkap tentang kemampuan model secara keseluruhan.

4. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada 3000 ulasan pengguna aplikasi PLN Mobile yang diambil dari *Google Play Store* menggunakan teknik *web scraping*, ditemukan bahwa terdapat 1.965 ulasan positif dan 1.035 ulasan negatif. Hasil pengujian menggunakan beberapa model algoritma dari *machine learning* dan *deep learning* adalah model *logistic regression* memiliki akurasi sebesar 84,47%, *decision tree* memiliki akurasi sebesar 79,30%, dan *random forest* memiliki akurasi sebesar 83,64%. Di sisi lain, model algoritma *deep learning*, yaitu *Neural Network Multi-Layer Perceptron* (MLP), memiliki akurasi sebesar 84,47%, sedangkan pengujian dengan *Long Short - Term Memory* (LSTM) menghasilkan akurasi sebesar 78,83%. Hasil tersebut menunjukkan bahwa dalam konteks sentimen analisis, model *machine learning* menggunakan algoritma *logistic regression* dan model *deep learning* menggunakan algoritma *neural network multi-layer perceptron* (MLP) mendominasi dalam hal akurasi dibandingkan dengan algoritma lainnya. Oleh karena itu, kedua model ini dapat dijadikan referensi unggulan untuk menerapkan sentimen analisis pada topik serupa. Untuk penelitian selanjutnya, disarankan untuk mengimplementasikan teknik *oversampling* dengan metode SMOTE agar jumlah data pada setiap kelas menjadi lebih seimbang. Dengan demikian, pengujian dengan algoritma yang diterapkan dapat memberikan performa yang lebih baik.

Daftar Pustaka

[1] E. Kaban, K. C. Brata, and A. H. Brata, “Evaluasi Usability Menggunakan Metode System Usability Scale (SUS) Dan Discovery Prototyping Pada Aplikasi PLN Mobile,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 3, no. 2, pp. 8952–8958, 2019.

[2] P. A. Lestari, I. Aknuranda, and A. D. Herlambang, “Evaluasi Usability Pada Antarmuka Pengguna Aplikasi PLN Mobile Menggunakan Metode Evaluasi Heuristik,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 3, pp. 2269–2275, 2019.

[3] J. A. Kumar, T. E. Trueman, and E. Cambria, “Gender-Based Multi-Aspect Sentiment Detection Using Multilabel Learning,” *Inf. Sci. (Ny).*, vol. 606, pp. 453–468, 2022, doi: 10.1016/j.ins.2022.05.057.

[4] Y. Pratama, D. T. Murdiansyah, and K. M. Lhaksmana, “Analisis Sentimen Kendaraan Listrik Pada Media Sosial Twitter Menggunakan Algoritma Logistic Regression dan Principal Component Analysis,” *J. Media Inform. Budidarma*, vol. 7, no. 1, pp. 529–535, 2023, doi: 10.30865/mib.v7i1.5575.

[5] A. R. Hidayati, A. S. Fitrani, M. A. Rosid, F. Sains, and U. Muhammadiyah, “Analisa Sentimen Pemilu 2019 Pada Judul Berita Online Menggunakan Metode Logistic Regression,” *J. Penerapan Sist. Inf.*, vol. 4, no. 2, pp. 298–305, 2023.

[6] E. N. Halim, B. Huda, and A. Elanda, “Perbandingan KNN, Decision Tree Dan Naïve Bayes Untuk Analisis Sentimen Marketplace Bukalapak,” *J. Comput. Eng. Syst. Sci.*, vol. 8, no. January, pp. 71–79, 2023.

[7] E. Fitri, “Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine,” *J. Transform.*, vol. 18, no. 1, p. 71, 2020, doi: 10.26623/transformatika.v18i1.2317.

[8] J. Asian, M. Dholah Rosita, and T. Mantoro, “Sentiment Analysis For The Brazilian Anesthesiologist Using Multi-Layer Perceptron Classifier And Random Forest Methods,” *J. Online Inform.*, vol. 7, no. 1, p. 132, 2022, doi: 10.15575/join.v7i1.900.

[9] A. A. Mudding and Arifin A Abd Karim, “Analisis Sentimen Menggunakan Algoritma LSTM Pada Media Sosial,” *J. Publ. Ilmu Komput. dan Multimed.*, vol. 1, no. 3, pp. 181–187, 2022, doi: 10.55606/jupikom.v1i3.517.

[10] L. Farsiah, A. Misbullah, and H. Husaini, “Analisis Sentimen Menggunakan Arsitektur Long Short-Term Memory (LSTM) Terhadap Fenomena Citayam Fashion Week,” *Cybersp. J. Pendidik. Teknol. Inf.*, vol. 6, no. 2, p. 86, 2022, doi: 10.22373/cj.v6i2.14687.

[11] N. S. Fathullah, Y. A. Sari, and P. P. Adikara, “Analisis Sentimen Terhadap Rating dan Ulasan Film Dengan Menggunakan Metode Klasifikasi Naïve Bayes dengan Fitur Lexicon-Based,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 2, pp. 590–593, 2020, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6987

[12] A. Agrani and B. Rikumahu, “Perbandingan Analisis Sentimen Terhadap Digital Payment ‘Go-Pay’ Dan ‘Ovo’ Di Media Sosial Twitter Menggunakan Algoritma Naïve Bayes Dan

Word Cloud," *Agustus*, vol. 7, no. 2, p. 2534, 2020.

[13] A. Amalia, W. Oktinas, I. Aulia, and R. F. Rahmat, "Determination Of Quality Television Programmes Based On Sentiment Analysis On Twitter," *J. Phys. Conf. Ser.*, vol. 978, no. 1, 2018, doi: 10.1088/1742-6596/978/1/012117.

[14] E. Y. Sari, A. D. Wierfi, and A. Setyanto, "Sentiment Analysis Of Customer Satisfaction On Transportation Network Company Using Naive Bayes Classifier," *2019 Int. Conf. Comput. Eng. Network, Intell. Multimedia, CENIM 2019 - Proceeding*, vol. 2019-Novem, 2019, doi: 10.1109/CENIM48368.2019.8973262.

[15] H. Juwiantho, E. I. Setiawan, J. Santoso, and M. H. Purnomo, "Sentiment Analysis Twitter Bahasa Indonesia Berbasis WORD2VEC Menggunakan Deep Convolutional Neural Network," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 1, pp. 181–188, 2020, doi: 10.25126/jtiik.202071758.

[16] M. U. Albab, Y. Karuniawati P, and M. N. Fawaiq, "Optimization Of The Stemming Technique On Text preprocessing President 3 Periods Topic," *J. Transform.*, vol. 20, no. 2, pp. 1–10, 2023, [Online]. Available: https://journals.usm.ac.id/index.php/transformatika/■page1

[17] A. Bimantara and T. A. Dina, "Klasifikasi Web Berbahaya Menggunakan Metode Logistic Regression," *Annu. Res. Semin.*, vol. 4, no. 1, pp. 173–177, 2019, [Online]. Available: https://seminar.ilkom.unsri.ac.id/index.php/ars/article/view/1932

[18] A. De Caigny, K. Coussement, and K. W. De Bock, "A New Hybrid Classification Algorithm For Customer Churn Prediction Based On Logistic Regression And Decision Trees," *Eur. J. Oper. Res.*, vol. 269, no. 2, pp. 760–772, 2018, doi: 10.1016/j.ejor.2018.02.009.

[19] P. Lauren, G. Qu, J. Yang, P. Watta, G. Bin Huang, and A. Lendasse, "Generating Word Embeddings from An Extreme Learning Machine For Sentiment Analysis And Sequence Labeling Tasks," *Cognit. Comput.*, vol. 10, no. 4, pp. 625–638, 2018, doi: 10.1007/s12559-018-9548-y.

[20] F. R. Suprihati, "Analisis Klasifikasi SMS Spam Menggunakan Logistic Regression," *J. Sist. Cerdas*, vol. 4, no. 3, pp. 155–160, 2021, doi: 10.37396/jsc.v4i3.166.

[21] N. Nurajijah, D. A. Ningtyas, and M. Wahyudi, "Klasifikasi Siswa SMK Berpotensi Putus Sekolah Menggunakan Algoritma Decision Tree, Support Vector Machine Dan Naive Bayes," *J. Khatulistiwa Inform.*, vol. 7, no. 2, pp. 85–90, 2019, doi: 10.31294/jki.v7i2.6839.

[22] I. Afdhal, R. Kurniawan, I. Iskandar, R. Salambue, E. Budianita, and F. Syafria, "Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 1, pp. 122–130, 2022, [Online]. Available: http://ojs.serambimekkah.ac.id/jnkti/article/view/4004/pdf

[23] D. A. Alboaneen, H. Tianfield, and Y. Zhang, "Sentiment Analysis Via Multi-Layer Perceptron Trained By Meta-Heuristic Optimisation," *Proc. - 2017 IEEE Int. Conf. Big Data, Big Data 2017*, vol. 2018-Janua, pp. 4630–4635, 2017, doi: 10.1109/BigData.2017.8258507.

[24] M. F. Muzakki, R. F. Umbara, F. Informatika, and U. Telkom, "Analisis Sentimen Mahasiswa Terhadap Fasilitas Universitas Telkom Menggunakan Metode Jaringan Saraf Tiruan Dan TF-IDF," *e-Prodeceeding Eng.*, vol. 6, no. 2, pp. 8608–8616, 2019.

[25] M. Kim and K.-H. Kang, "Comparison of Neural Network Techniques for Text Data Analysis," *Int. J. Adv. Cult. Technol.*, vol. 8, no. 2, pp. 231–238, 2020, doi: https://doi.org/10.17703/IJACT.2020.8.2.231.

[26] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM Network: A Deep Learning Approach For Short-Term Traffic Forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, 2017, doi: 10.1049/iet-its.2016.0208.

[27] K. Goseva-Popstojanova and J. Tyo, "Identification Of Security Related Bug Reports Via Text Mining Using Supervised And Unsupervised Classification," *Proc. - 2018 IEEE 18th Int. Conf. Softw. Qual. Reliab. Secur. QRS 2018*, pp. 344–355, 2018, doi: 10.1109/QRS.2018.00047.

[28] S. Chotirat and P. Meesad, "Part-Of-Speech Tagging Enhancement To Natural Language Processing For Thai Wh-Question Classification With Deep Learning," *Heliyon*, vol. 7, no. 10, p. e08216, 2021, doi: 10.1016/j.heliyon.2021.e08216.

[29] S. Syafrizal, M. Afdal, and R. Novita, "Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 10–19, 2023, doi: 10.57152/malcom.v4i1.983.

[30] T. Informatika and F. Teknik, "Analisis Sentimen Terkait Ulasan Pada Aplikasi PLN Mobile Menggunakan Metode Support Vector Machine," *KESATRIA - J. Penerapan Sist. Inf. (Komputer Manajemen)*, vol. 5, no. 1, pp. 303–312, 2024.