JURNAL FASILKOM
Volume 15 No. 2 | Agustus 2025: 355-367

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Analisis Sentimen dan Prediksi Ulasan Pada Aplikasi Info BMKG

Randi Afif[1], Kristiawan Nugroho[2]
[1,2]Magister Teknologi Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank Semarang
[*1]randiafif0035@mhs.unisbank.ac.id, [2]kristiawan@edu.unisbank.ac.id

Abstract

The Info BMKG application provides weather and climate information for the Indonesian public. User reviews on Google Play Store reflect satisfaction and criticism that can be analyzed to improve services. This study aims to classify sentiments and predict the volume of reviews using Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) algorithms. A total of 3,000 reviews were collected through web scraping, and after preprocessing—including data cleaning, tokenization, stemming, stopword removal, and labeling—the dataset was reduced to 2,645 reviews. The results show that LSTM outperforms RNN in sentiment classification, achieving 90% accuracy and an F1-score of 0.90, while RNN obtained 87% accuracy and an F1-score of 0.82. For predicting the number of negative reviews, RNN performed better (MSE: 104.97; MAE: 7.61; R^2: 0.22), whereas both models performed poorly for the positive category (negative R^2 values). These findings indicate that LSTM is more effective for sentiment classification, while RNN excels in predicting negative review trends.

Keywords: sentiment analysis, prediction, info BMKG, recurrent neural network, long short-term memory, user reviews.

Abstrak

Aplikasi Info BMKG menyediakan informasi cuaca dan iklim bagi masyarakat Indonesia. Ulasan pengguna di *Google Play Store* merefleksikan kepuasan dan kritik yang dapat dianalisis untuk peningkatan layanan. Penelitian ini bertujuan mengklasifikasikan sentimen dan memprediksi volume ulasan menggunakan *Recurrent Neural Network* (*RNN*) dan *Long Short-Term Memory* (*LSTM*). Sebanyak 3000 ulasan diperoleh melalui *web scraping* dan setelah diproses dengan pembersihan data, tokenisasi, *stemming*, penghapusan *stopword*, dan *labelling*, maka jumlahnya menjadi 2645 ulasan. Hasil menunjukkan *LSTM* unggul pada klasifikasi sentimen dengan akurasi 90% dan *F1-score* 0,90, sedangkan *RNN* memperoleh akurasi 87% dan *F1-score* 0,82. Pada prediksi jumlah ulasan negatif, *RNN* lebih baik (*MSE*: 104,97; *MAE*: 7,61; R^2: 0,22), sementara kedua model kurang optimal untuk kategori positif (R^2 negatif). Temuan ini menunjukkan *LSTM* lebih efektif untuk klasifikasi, sedangkan *RNN* lebih unggul dalam prediksi ulasan negatif.

Kata kunci: analisis sentimen, prediksi, info BMKG, *recurrent neural network*, *long short-term memory*, ulasan pengguna.

1. Pendahuluan

Kemajuan teknologi digital telah mempengaruhi cara masyarakat Indonesia dalam mengakses informasi kebencanaan dan cuaca melalui aplikasi mobile, seperti Info BMKG yang dikembangkan oleh Badan Meteorologi, Klimatologi, dan Geofisika. Aplikasi ini menyajikan informasi penting seperti prakiraan cuaca, data gempa bumi, peringatan dini, dan kualitas udara, yang krusial untuk mitigasi risiko bencana dan mendukung aktivitas sehari-hari masyarakat [1,2]. Mempertimbangkan kompleksitas lingkungan yang terus berubah, aplikasi ini berperan penting dalam meningkatkan kesadaran terhadap berbagai ancaman bencana dan perubahan iklim yang dapat mempengaruhi masyarakat secara langsung.

Namun, efektivitas aplikasi tidak semata-mata bergantung pada fitur teknis. Persepsi dan pengalaman pengguna, yang diekspresikan melalui ulasan di platform distribusi aplikasi seperti *Google Play Store*, juga memainkan peran penting. Ulasan ini menyediakan data yang kaya akan opini publik dan penting untuk dianalisis guna menilai tingkat kepuasan pengguna serta mengidentifikasi masalah teknis [3]. Analisis sistematis terhadap ulasan ini dapat memberikan wawasan berharga untuk pengembangan aplikasi, termasuk perbaikan terhadap fitur-fitur yang ada.

Banyak penelitian sebelumnya telah memanfaatkan algoritma *machine learning* dan *deep learning* dalam analisis sentimen ulasan aplikasi. Metode seperti *Naïve Bayes* dan *Support Vector Machine* (*SVM*) telah diterapkan dengan cukup luas. Meskipun demikian, pendekatan berbasis *Recurrent Neural network* (*RNN*) dan *Long Short-Term Memory* (*LSTM*) menunjukkan keunggulan dalam menangani data teks berurutan karena kemampuannya dalam mempertahankan konteks [4,5].

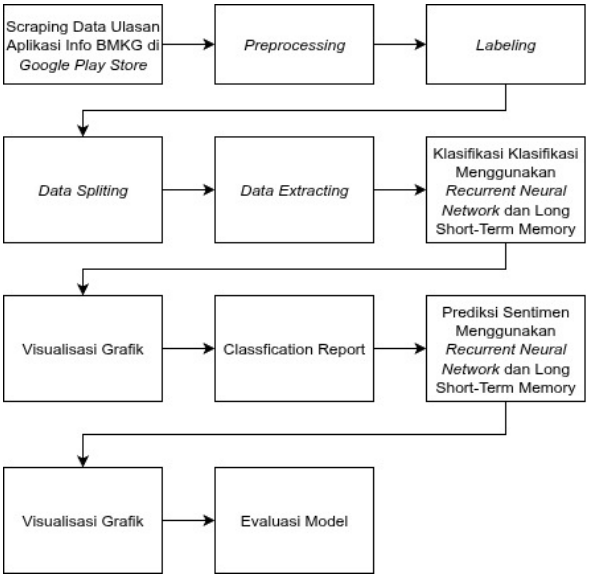
Namun, meskipun penelitian tentang analisis sentimen telah berkembang, masih ada kekurangan dalam kajian yang secara spesifik menganalisis ulasan pengguna aplikasi Info BMKG menggunakan pendekatan *deep learning*, khususnya dalam konteks Bahasa Indonesia. Sebagian besar studi hanya berfokus pada klasifikasi sentimen dan belum memprediksi perilaku pengguna di masa depan, seperti prediksi jumlah ulasan, yang penting untuk menilai tingkat keterlibatan pengguna dan respons terhadap pembaruan layanan [1,6].

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Tujuan dari penelitian ini meliputi menganalisis sentimen aplikasi Info BMKG dengan metode *RNN* dan *LSTM* serta membandingkan akurasi kedua pendekatan tersebut. Selain itu, penelitian ini juga akan memprediksi jumlah ulasan pengguna di masa mendatang berdasarkan data historis. Dengan memberikan wawasan tambahan kepada pengelola aplikasi, hasil penelitian diharapkan bisa meningkatkan kualitas layanan aplikasi publik di Indonesia, serta memberikan kontribusi dalam pengembangan *Natural Language Processing* (*NLP*) untuk Bahasa Indonesia [2,7].

2. Metode Penelitian

2.1 Alur Penelitian



Gambar 1 Alur Penelitian

Alur penelitian ditunjukkan pada Gambar 1. Alur penelitian diawali proses *scraping* yang menggunakan pustaka *google-play-scraper* yang diinstal dalam lingkungan *Python*, dan berhasil mengumpulkan sebanyak 3000 ulasan yang berisi informasi seperti teks ulasan, rating bintang, dan tanggal publikasi [2]. Setelah pengumpulan data, tahapan selanjutnya adalah *preprocessing* dan *labeling*, yang penting untuk menyiapkan data agar siap untuk analisis lebih lanjut. Untuk memastikan model berfungsi dengan baik dengan data baru, data dibagi menjadi dua set: pelatihan dan pengujian [8].

Selanjutnya, proses ekstraksi data dilakukan, yang meliputi ekstraksi fitur relevan dari data ulasan. Penelitian ini memiliki dua tujuan utama: pertama, klasifikasi sentimen menggunakan model *Recurrent Neural network* (*RNN*) dan *Long Short-Term Memory* (*LSTM*); kedua, prediksi jumlah ulasan pengguna aplikasi. Klasifikasi sentimen akan diukur menggunakan *classification report* beserta metrik regresi untuk evaluasi model [5,9]. Semua proses dilakukan menggunakan bahasa pemrograman *Python* versi 3.6.8, yang mendukung berbagai pustaka analisis data dan *machine learning* yang diperlukan.

2.2 Pengumpulan Data

Dalam fase pengumpulan data, sebanyak 3000 ulasan aplikasi Info BMKG dikumpulkan. Ulasan tersebut mengandung informasi penting yang akan digunakan untuk analisis. Data yang dikumpulkan berisi teks ulasan, rating bintang, dan tanggal publikasi. Untuk analisis klasifikasi sentimen, hanya digunakan ulasan yang memiliki rating 1–2 (negatif) dan 4–5 (positif) [2]. Dengan demikian, data yang relevan dipilih untuk representasi data yang lebih akurat dalam konteks analisis sentimen [4]. Sementara itu, untuk tujuan prediksi jumlah ulasan, digunakan data agregat jumlah ulasan harian. Penggunaan data agregat ini penting untuk memahami pola perubahan dalam ulasan dari waktu ke waktu, yang dapat membantu dalam perencanaan pengembangan aplikasi lebih lanjut [10,11].

Agar data siap dianalisis, fase pemrosesan mencakup langkah pra-perlakuan. Langkah ini penting untuk memastikan keakuratan model yang dihasilkan dan memberikan informasi yang relevan dan berharga kepada pengembang aplikasi guna meningkatkan kualitas layanan [5,12].

2.3 *Data Preprocessing*

Preprocessing data merupakan langkah krusial dalam pengolahan teks sebelum digunakan dalam proses klasifikasi sentimen maupun prediksi. Untuk keperluan studi ini, kami membersihkan dan mengorganisir evaluasi pengguna aplikasi BMKG Info yang kami ambil dari *Google Play Store*. Beberapa tahapan penting dalam *preprocessing* yang diterapkan meliputi *casefolding*, *filtering*, *tokenizing*, dan *stemming*.

Casefolding sebagai langkah pertama, teks diubah menjadi huruf kecil. Tujuannya adalah untuk menetapkan aturan kapitalisasi yang seragam untuk istilah-istilah dengan makna yang identik. Misalnya, "Cuaca" dan "cuaca" disamakan menjadi "cuaca" agar diperlakukan sebagai satu entitas dalam analisis [2].

Sebaliknya, penyaringan bertujuan untuk mengecualikan kata-kata yang sering muncul (*stopword*s), tanda baca, simbol, dan angka yang tidak memberikan kontribusi signifikan terhadap analisis dari data.. *Stopwords* dalam Bahasa Indonesia yang umum dihapus mencakup kata seperti "yang", "dan", "dengan", serta "atau" [13]. Langkah ini sangat penting guna memastikan kualitas data yang akan digunakan dalam pelatihan model menjadi lebih optimal [14].

Tokenizing adalah proses selanjutnya yang melibatkan pemecahan kalimat menjadi kata-kata tunggal atau token. Sebagai contoh, ulasan "aplikasi sangat membantu" akan dipecah menjadi daftar token: "aplikasi", "sangat", "membantu". Token ini kemudian digunakan sebagai unit dasar dalam tahap representasi data dengan *word embedding* [2].

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Stemming merupakan suatu prosedur dengan tujuan akhir untuk menyederhanakan semua token ke bentuk paling sederhana, atau kata dasar. Contoh dari tahap ini adalah penggantian "menggunakan" dengan "guna", dan "berjalan" dengan "jalan". Tujuannya adalah untuk menstandardisasi dan mempersempit penyajian fakta dengan mengurangi perbedaan dalam istilah-istilah yang terdengar berbeda tetapi menyiratkan hal yang sama. [15].

Penggunaan pustaka *Python*, seperti *NLTK* dan Sastrawi, dalam tahap ini memungkinkan pemrosesan yang lebih efisien dan terstruktur. Tahap *preprocessing* ini merupakan landasan penting sebelum data digunakan untuk pelatihan model *RNN* dan *LSTM*, mengingat bahwa kualitas data input sangat mempengaruhi performa model klasifikasi dan prediksi [13].

2.4 *Labeling*

Pelabelan data merupakan tahapan penting dalam proses persiapan sebelum pelatihan model klasifikasi, di mana setiap ulasan pengguna diberi kategori sentimen tertentu. Dalam studi ini, ulasan terhadap aplikasi Info BMKG dikelompokkan menjadi dua jenis sentimen, yakni positif dan negatif. Peringkat 1 dan 2 menunjukkan sentimen negatif dalam ulasan, sementara peringkat 4 dan 5 menunjukkan sentimen positif. Ulasan dengan peringkat 3 tidak disertakan dalam penelitian ini karena dianggap netral dan tidak memihak [16].

Proses pemberian label ini dilakukan secara otomatis dengan bantuan skrip *Python*, yang memanfaatkan data rating hasil scraping dari *Google Play Store*. Otomatisasi ini bertujuan meningkatkan efisiensi dan mengurangi potensi bias yang dapat timbul jika dilakukan secara manual [13]. Selain itu, perhatian juga diberikan pada keseimbangan distribusi kelas untuk menghindari masalah ketimpangan data (*class imbalance*), yang dapat mempengaruhi performa model. Dengan pelabelan yang akurat dan konsisten, model diharapkan mampu belajar secara efektif dan memberikan hasil klasifikasi maupun prediksi sentimen yang optimal [13].

2.5 *Data splitting*

Data splitting merupakan tahap penting dalam proses pembangunan model pembelajaran mesin, karena bertujuan untuk memisahkan data ke dalam dua bagian utama: data latih (*training set*) dan data uji (*testing set*). Pada penelitian ini, data yang telah diproses dan diberi label dibagi dengan rasio 80:20; 80% digunakan untuk melatih model, sementara 20% digunakan untuk menguji kinerjanya [2,13].

Tujuan utama bagian ini adalah menguji generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya. Selagi model mempelajari pola dalam data pelatihan, model diuji pada data uji untuk melihat seberapa baik model dapat memprediksi label pada data yang tidak ada dalam set pelatihan [15,16]. Dengan melakukan pemisahan data yang sistematis dan metodologis, peneliti dapat memberikan evaluasi yang lebih objektif terkait performa model yang dibangun. Selain itu, ini juga memberikan kesempatan untuk menerapkan teknik tuning model yang lebih mendalam sesuai dengan hasil evaluasi dari dataset yang terpisah [15,17].

2.7 *Data Extracting*

Pada tahap *Data Extracting*, proses yang dilakukan bertujuan untuk mengubah teks ulasan yang telah melalui tahapan *preprocessing* menjadi bentuk numerik yang dapat dipahami oleh model *machine learning*. Penggunaan *Tokenizer* sangat vital dalam langkah ini, karena berfungsi untuk membangun kamus kata (*word index*) berdasarkan semua kata unik yang terdapat dalam kumpulan data ulasan [9]. Setiap kata unik yang ditemukan akan diberikan indeks berupa bilangan bulat yang merepresentasikan posisi kata tersebut dalam kamus. Proses ini memungkinkan model untuk mengenali kata-kata dalam bentuk yang lebih terstruktur.

Setelah itu, setiap ulasan akan dikonversi menjadi deretan bilangan bulat yang merepresentasikan kata-kata dalam ulasan tersebut. Proses konversi ini diikuti dengan penggunaan teknik *Padding*. *Padding* diperlukan agar seluruh data memiliki dimensi yang sama, sehingga bisa diproses dengan baik oleh arsitektur *neural network* [9]. *Padding* membantu dalam memastikan bahwa setiap masukan ke model memiliki panjang yang seragam, yang merupakan syarat penting dalam kebanyakan model pembelajaran mesin.

Implementasi *Tokenizer* pada penelitian ini menggunakan pustaka *Keras*, di mana parameter yang digunakan adalah *num_words* sebanyak 1000 kata paling sering muncul. Dengan memberikan batasan ini, pemodelan dapat difokuskan pada kata-kata yang paling relevan dalam konteks analisis sentimen, sambil mengabaikan kata-kata yang mungkin kurang signifikan [9]. Pada akhirnya, data teks telah sepenuhnya dikonversi menjadi data numerik yang siap digunakan dalam pelatihan model klasifikasi sentimen.

2.8 Klasifikasi Sentimen dengan *RNN* dan *LSTM*

Proses klasifikasi sentimen dilakukan dengan membangun dua model *neural network* yang berbeda, yaitu *Recurrent Neural network* (*RNN*) dan *Long Short-Term Memory* (*LSTM*). Model pertama, *RNN*, memanfaatkan arsitektur sederhana yang mampu memproses data sekuensial dengan mempertahankan informasi dari input sebelumnya. Namun, *RNN* cenderung lemah dalam mengingat konteks jangka panjang, yang membuatnya tidak selalu ideal untuk analisis sentimen yang kompleks [18].

Sebagai solusi, *LSTM* digunakan karena dikenal lebih unggul dalam menangani dependensi kata yang panjang serta mampu mengatasi permasalahan *vanishing gradient* yang sering terjadi pada *RNN*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

konvensional [6,19]. Pada kedua model ini, proses pelatihan dilakukan dengan memanfaatkan *layer embedding* sebagai *input layer* untuk mengubah indeks kata menjadi vektor berdimensi rendah. Selanjutnya, data diproses melalui *hidden Layer* yang terdiri dari lapisan *RNN* atau *LSTM* yang diatur untuk mengoptimalkan pembelajaran pola data [20].

2.9 Visualisasi Grafik

Untuk lebih memahami kinerja model dalam kategorisasi sentimen, penting untuk merepresentasikan hasil penilaian dan analisis model secara visual. Dengan melakukan pengukuran ini, kita dapat lebih memahami cara menginterpretasikan prediksi model *RNN* dan *LSTM*, terutama dalam mengidentifikasi sentimen positif dan negatif. Diagram batang adalah salah satu jenis visualisasi yang digunakan., yang menggambarkan distribusi jumlah data yang telah diklasifikasikan ke dalam masing-masing kategori sentimen. Melalui grafik ini, peneliti dapat mengamati secara langsung apakah prediksi model bersifat seimbang atau menunjukkan kecenderungan tertentu dalam mengklasifikasikan ke salah satu kelas sentimen [4].

Selain grafik batang, *confusion matrix* juga digunakan untuk memvisualisasikan performa klasifikasi. *Confusion matrix* menampilkan jumlah prediksi yang benar dan salah dari setiap kelas, serta memberikan gambaran tentang pola kesalahan yang terjadi dalam proses klasifikasi. Melalui *confusion matrix*, dapat diketahui jumlah *true positive*, *false positive*, *true negative*, dan *false negative* secara lebih terstruktur. Visualisasi ini sangat berguna untuk mengungkapkan kelemahan model dalam membedakan sentimen positif dan negatif dan memungkinkan perbandingan akurasi masing-masing model secara lebih komprehensif. Seluruh visualisasi ini dibuat menggunakan pustaka *Matplotlib* dan seaborn, yang telah tersedia di *Python*. Dengan memanfaatkan kedua pustaka ini, hasil visualisasi dapat disajikan secara informatif dan lebih mudah dipahami sebagai bagian dari analisis performa model klasifikasi sentimen pada penelitian ini [4,5].

2.10 *Classification report*

Evaluasi performa model klasifikasi dilakukan dengan memanfaatkan *classification report* yang menyajikan beberapa metrik penting, yaitu *accuracy*, *precision*, *Recall*, dan *F1-score* [9]. *Accuracy* Mengevaluasi akurasi prediksi relatif terhadap keseluruhan set data yang digunakan untuk pengujian. Accuracy digunakan untuk mengevaluasi ketepatan prediksi model terhadap keseluruhan data uji, sebagaimana pada Persamaan (1), yaitu perbandingan antara jumlah *True Positive* (*TP*) dan *True Negative* (*TN*) terhadap total data yang mencakup *TP*, *False Positive* (*FP*), *TN*, dan *False Negative* (*FN*). *TP* adalah jumlah data positif yang diprediksi positif oleh model, *TN* adalah jumlah data negatif yang diprediksi negatif, *FP* adalah jumlah data negatif yang salah diprediksi positif, sedangkan *FN* adalah jumlah data positif yang salah diprediksi negatif. Meskipun akurasi bermanfaat untuk melihat ketepatan prediksi, nilainya dapat kurang sesuai jika jumlah data pada tiap kelas tidak seimbang. Untuk mengetahui kekuatan prediktif model, menghitung *accuracy* adalah solusi untuk mencerminkan kinerja yang optimal jika terjadi ketidakseimbangan jumlah data antar kelas. Adapun rumusnya sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

Precision, sebagaimana pada Persamaan (2), merupakan perbandingan antara TP dengan jumlah TP dan FP. Nilai ini menunjukkan seberapa tepat model dalam memprediksi kelas positif, di mana precision yang tinggi berarti model jarang memberikan prediksi positif yang keliru, berikut rumus perhitungannya:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

Recall, sebagaimana pada Persamaan (3), merupakan perbandingan antara *TP* dengan jumlah *TP* dan *FN*. Metrik ini mengukur kemampuan model dalam mendeteksi semua data positif yang ada, di mana *recall* yang tinggi menandakan sebagian besar data positif berhasil teridentifikasi, berikut rumusnya:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

F1-score, sebagaimana pada Persamaan (4), dihitung sebagai rata-rata harmonik antara *precision* dan *recall*. Nilai *F1-score* yang tinggi menunjukkan model memiliki performa seimbang antara ketepatan prediksi positif dan kemampuan mendeteksi seluruh data positif, berikut rumusnya:

$$F1 - Score = 2\ \mathrm{X} \frac{Precision\ \mathrm{x}\ Recall}{Precision + Recall} \quad (4)$$

Hasil dari *classification report* ini diproses menggunakan fungsi *classification_report* dari pustaka *sklearn.metrics*, yang menyajikan hasil metrik untuk masing-masing kelas sentimen, baik positif maupun negatif, serta nilai rata-rata keseluruhan model [21].

2.11 Prediksi Sentimen dengan *RNN* dan *LSTM*

Pada tahap prediksi ini, digunakan dua algoritma deep learning, yaitu *Recurrent Neural Network* (*RNN*) dan *Long Short-Term Memory* (*LSTM*), untuk memprediksi jumlah ulasan dengan sentimen positif dan negatif pada aplikasi Info BMKG berdasarkan data historis bulanan yang telah melalui proses pengolahan sebelumnya. Model *RNN* dibangun dengan arsitektur berlapis yang terdiri dari kombinasi lapisan *SimpleRNN*, *Dropout*, dan *Dense*. Data yang digunakan diformat sebagai deret waktu (*time series*) dengan jendela input sepanjang 12 bulan untuk memprediksi nilai pada bulan berikutnya [22].

Sementara itu, model *LSTM* dirancang menggunakan struktur jaringan yang lebih kompleks karena memiliki kemampuan yang lebih baik dalam menangkap pola

P-ISSN : 2089-3353
E-ISSN : 2808-9162

temporal jangka panjang. Arsitektur *LSTM* disusun dari beberapa lapisan *LSTM* yang diikuti oleh *Dropout* dan *Dense*, sehingga memungkinkan pembelajaran fitur yang lebih dalam. Proses pelatihan kedua model dilakukan menggunakan fungsi *loss Mean Squared Error* (*MSE*) dan optimisasi *RMSprop*. Untuk mengurangi pengaruh *outlier* pada data, digunakan teknik normalisasi *RobustScaler*, sedangkan pembagian data dilakukan dengan skema 80% data latih dan 20% data uji [23].

Setelah proses pelatihan selesai, kedua model digunakan untuk memproyeksikan jumlah ulasan hingga akhir tahun 2026, dengan tujuan mengidentifikasi tren sentimen dari waktu ke waktu. Evaluasi terhadap performa model dilakukan menggunakan tiga metrik utama, yakni *Mean Squared Error* (*MSE*), *Mean Absolute Error* (*MAE*), dan koefisien determinasi (R^2 *Score*). *MSE* menghitung rata-rata dari kuadrat selisih antara nilai prakiraan dan aktual, sedangkan kesalahan absolut rata-rata (*MAE*) mengkuantifikasi rata-rata semua kesalahan. Tingkat kesalahan prediksi yang menurun ditunjukkan oleh nilai *MSE* dan *MAE* yang lebih rendah. Selain itu, kapasitas prediktif yang kuat ditunjukkan oleh nilai R^2 yang mendekati 1, yang menunjukkan bahwa model tersebut dapat menjelaskan sebagian fluktuasi data. [24].

2.12 Visualisasi Grafik Prediksi

Hasil prediksi dari kedua model divisualisasikan dalam bentuk grafik untuk memudahkan pemahaman dan analisis tren data. Visualisasi dilakukan menggunakan *library Matplotlib* dalam format *linechart*. Data yang divisualisasikan meliputi gabungan antara data aktual dan hasil prediksi dari model *RNN* dan *LSTM*, baik untuk sentimen positif maupun negatif. Visualisasi ini menyediakan gambaran perbandingan tren antara data asli dan data hasil prediksi serta menunjukkan kecenderungan jumlah ulasan positif dan negatif dari waktu ke waktu [25].

Selain itu, dibuat pula visualisasi prediksi jangka panjang dari *LSTM* hingga akhir tahun 2026. Penyajian visual ini mempermudah interpretasi hasil dan memberikan gambaran lebih jelas mengenai fluktuasi data ulasan pengguna aplikasi Info BMKG, sehingga pemangku keputusan dapat menindaklanjuti tren yang teridentifikasi.

3. Hasil dan Pembahasan

3.1 Pengumpulan Data

Evaluasi pengguna terhadap aplikasi BMKG Info yang terdapat di *Google Play Store* menjadi dasar penelitian ini. Data dikumpulkan menggunakan paket *google-play-scraper*, yang ditulis dalam bahasa *Python*. Untuk memastikan data yang diperoleh bersih dan relevan, proses pengumpulan dilakukan secara bertahap hingga terkumpul 3.000 ulasan pelanggan. Teks lengkap ulasan pengguna yang terdapat pada kolom "Full Text" merupakan kolom data yang diperoleh dan diproses lebih lanjut. Tabel 1 menampilkan hasil dari proses pengumpulan data.

Tabel 1 Hasil Pengumpulan Data

Full Text
sangat membantu sekali
Oke
90% Akurat dibanding aplikasi cuaca lain, dan juga bebas iklan. akan lebih menarik jika dengan adanya widget di layar depan hp. itu saja
Ok
kurang akurat

3.2 *Preprocessing* Data

Tujuan dari langkah prapemrosesan adalah untuk membersihkan dan mengorganisasikan data teks agar dapat dianalisis lebih baik nantinya.

3.2.1 *Case folding*

Pada tahap casefolding, data teks diubah menjadi huruf kecil. Tujuannya adalah untuk memastikan bahwa pemrosesan data tidak terpengaruh oleh variasi kapitalisasi. Dengan demikian, kata seperti “Akurat” dan “akurat” akan diperlakukan sama oleh sistem. Proses ini bertujuan untuk menjaga konsistensi penulisan kata dalam keseluruhan data teks yang akan dianalisis.

3.2.2 *Filtering*

Pada tahap ini, data disaring untuk menghilangkan informasi yang tidak relevan. Bagian dari prosedur ini meliputi penghapusan istilah-istilah yang sering muncul dan tidak berkontribusi apa pun terhadap penelitian, serta konjungsi dan kata penghubung. Selain itu, *filtering* juga menghilangkan karakter-karakter seperti angka, tanda baca, simbol khusus, hingga spasi ganda. Dengan demikian, data yang tersisa hanyalah kata-kata inti yang dianggap penting bagi pemodelan.

3.2.3 *Tokenizing*

Setelah data bersih dari kata-kata yang tidak relevan, proses selanjutnya adalah *tokenizing*. Pada tahap ini, kalimat dalam ulasan dipecah menjadi unit kata-kata kecil (token) agar dapat diproses secara lebih spesifik oleh algoritma *machine learning*. Perhatikan kalimat berikut: "Aplikasi ini sangat bagus." Token seperti ["aplikasi", "ini", "sangat", "bagus"] akan digunakan untuk mempartisinya. Data sudah dipreteli menjadi unit kata sederhana selama tokenisasi, yang memfasilitasi pemrosesan teks.

3.2.4 *Stemming*

Tahap terakhir adalah *stemming*, yang melibatkan reduksi kata menjadi bentuk paling fundamentalnya, atau kata dasar. Sebagai contoh, istilah "jalan," "lari," dan "utama" akan diubah menjadi versi paling dasar dari "berjalan," "berlari," dan "bermain," masing-masing. Langkah ini penting agar sistem secara konsisten mengenali makna kata yang mendasarinya dengan

P-ISSN : 2089-3353
E-ISSN : 2808-9162

mengurangi varian kata. Tabel 2 menampilkan hasil prapemrosesan.

Tabel 2 Hasil *Preprocessing*

Full Text	*Text Clean*	*Text Filtering*	*Text Tokenizing*	*Text Stemming*
sangat membantu sekali	sangat membantu sekali	membantu	['membantu']	bantu
oke	oke	oke	['oke']	oke
90% Akurat dibanding aplikasi cuaca lain, dan juga bebas iklan. akan lebih menarik jika dengan adanya widget di layar depan hp. itu saja	akurat dibanding aplikasi cuaca lain dan juga bebas iklan akan lebih menarik jika dengan adanya widget di layar depan hp itu saja	akurat dibanding aplikasi cuaca bebas iklan menarik widget layar hp	['akurat', 'dibanding', 'aplikasi', 'cuaca', 'bebas', 'iklan', 'menarik', 'widget', 'layar', 'hp']	akurat banding aplikasi cuaca bebas iklan tarik widget layar hp
ok kurang akurat	ok kurang akurat	ok akurat	['ok'] ['akurat']	ok akurat

3.3 *Labeling* Data

Tahap *labeling* dilakukan untuk memberikan penanda (label) pada setiap data ulasan yang telah dikumpulkan. Kategori label ditentukan berdasarkan rating bintang yang diberikan oleh pengguna pada aplikasi. Secara umum, ulasan dengan empat atau lima bintang menunjukkan sentimen positif, sedangkan ulasan dengan satu atau dua bintang menunjukkan sentimen negatif. Demi objektivitas dan minimnya bias dalam penelitian sentimen positif/negatif, studi ini tidak menyertakan ulasan dengan peringkat 3 bintang.

Skrip Python digunakan untuk mengotomatiskan proses pelabelan, yang mengurangi kemungkinan kesalahan manusia dan memastikan temuan yang konsisten dan akurat. Pemodelan klasifikasi berikut akan menggunakan label yang diberikan sebagai titik awal. Tabel 3 menampilkan hasil pelabelan.

Tabel 3. Hasil *Labeling*

Text Stemming	*Labeling*
bantu	Positif
oke	positif
akurat banding aplikasi cuaca bebas iklan tarik widget layar hp	positif
ok	positif
akurat	negatif

Dari total 3.000 data ulasan awal, proses labeling menghasilkan 2.645 data terlabel yang memenuhi kriteria penelitian, terdiri dari data sentimen positif dan negatif, sedangkan 355 data dengan rating 3 bintang dikeluarkan dari analisis.

3.4 Klasifikasi Sentimen

Pada tahap ini dilakukan proses pemodelan untuk klasifikasi sentimen menggunakan dua algoritma, yaitu *Recurrent Neural network* (*RNN*) dan *Long Short-Term Memory* (*LSTM*). Setelah melalui proses labelling, dataset yang digunakan menjadi berjumlah 2645 data, yang kemudian dibagi menjadi data latih (*train_set*) sebanyak 2116 data dan data uji (*test_set*) sebanyak 529 data. Label yang digunakan yaitu *y_train* dan *y_test* dengan jumlah yang sama sesuai pembagian dataset.
Pemodelan dilakukan menggunakan *library Keras* pada *Python*, dengan arsitektur dan parameter yang disesuaikan dengan karakteristik masing-masing algoritma. Proses klasifikasi bertujuan untuk memetakan data ulasan ke dalam dua kategori sentimen, yaitu positif dan negatif.

3.4.1 Klasifikasi Sentimen Model *RNN*
Pada penelitian ini, proses pemodelan pertama dilakukan menggunakan algoritma *Recurrent Neural network* (*RNN*). Data yang digunakan berupa hasil *preprocessing* dan *labeling* ulasan yang telah disimpan dalam format *CSV*. Data tersebut dipisahkan menjadi fitur berupa hasil *tokenizing* dan *stemming* dari teks ulasan, serta label sentimen positif dan negatif. Proses tokenisasi dilakukan dengan *library Keras Tokenizer* yang membatasi jumlah kata maksimal sebanyak 1000 kata paling sering muncul. Selanjutnya, teks diubah menjadi sekuens angka melalui metode *texts_to_sequences*(), lalu dirapikan panjangnya dengan proses *Padding* sehingga seluruh data memiliki panjang yang seragam, yaitu 100 token.

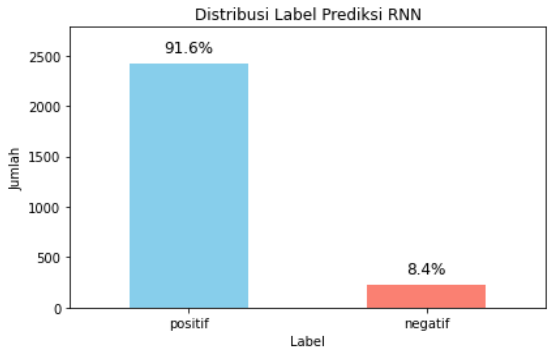
Selanjutnya, LabelEncoder dari sklearn digunakan untuk mengonversi label data ke dalam format numerik. Dengan menggunakan fungsi train_test_split, data kemudian dipartisi menjadi data latih (80%) dan data uji (20%) untuk mencapai distribusi data yang seimbang.
Salah satu dari banyak lapisan yang membentuk model *RNN* adalah Lapisan Embedding, yang dapat menampung hingga seribu kata., dimensi output 64, dan panjang input 100. Lapisan berikutnya adalah *SimpleRNN Layer* dengan 128 unit neuron serta menggunakan fungsi aktivasi *ReLU*. Setelah itu, model dilengkapi dengan dua lapisan *Dense Layer* berturut-turut yang masing-masing memiliki 64 neuron dan 32 neuron, dengan aktivasi *ReLU* pada keduanya. Karena kategorisasi bersifat biner (positif atau negatif), lapisan terakhir menggunakan Lapisan Keluaran Padat dengan fungsi aktivasi sigmoid dan satu neuron.

Ukuran evaluasi yang disebut Akurasi, fungsi kerugian yang disebut binary_crossentropy, dan pengoptimal RMSprop digunakan untuk membangun model ini. Proses pelatihan model disempurnakan dengan ModelCheckpoint, yang secara otomatis menyimpan bobot model optimal berdasarkan akurasi validasi tertinggi, memastikan model terbaik. Setelah 20 iterasi pelatihan dengan ukuran batch 32, model divalidasi menggunakan dataset X_test dan y_test.

Untuk lebih memahami distribusi data, tahap selanjutnya dari penelitian ini adalah membuat

P-ISSN : 2089-3353
E-ISSN : 2808-9162

representasi visual dari hasil kategorisasi. Visualisasi data ditampilkan sebagai diagram batang, yang membandingkan jumlah ulasan yang diklasifikasikan sebagai positif dan negatif. Sentimen pengguna tentang aplikasi yang dianalisis direpresentasikan secara visual, sehingga memudahkan pembaca untuk memahami trennya. Pada Gambar 2 terdapat diagram batang distribusi label prediksi *RNN*.



Gambar 2. *Barplot* Data Hasil Klasifikasi *RNN*

Dalam proses analisis sentimen, visualisasi data sangat penting untuk memberikan gambaran umum mengenai kata-kata yang sering muncul pada masing-masing kategori sentimen. Salah satu metode visualisasi yang umum digunakan adalah *Wordcloud*, yang menampilkan frekuensi kata berdasarkan ukuran huruf. *Wordcloud* ini dihasilkan dari hasil klasifikasi menggunakan metode *Recurrent Neural network* (*RNN*). Worcloud bisa dilihat pada gambar 3 dan gambar 4.

Gambar 3 *Wordcloud* Positif *RNN*

Gambar 3 merupakan *Wordcloud* untuk sentimen positif. Kata-kata yang sering muncul adalah "bantu", "bagus", "mantap", dan "informasi", yang menggambarkan apresiasi pengguna terhadap manfaat aplikasi, terutama terkait informasi bencana dan cuaca.

Gambar 4. *Wordcloud* Negatif *RNN*

Awan kata yang mewakili kategori sentimen negatif ditunjukkan pada Gambar 4. Kata-kata seperti "baik", "pembaruan", "aplikasi", dan "lokasi" mendominasi visualisasi ini, menunjukkan bahwa pengguna memiliki kekhawatiran tentang fungsionalitas dan kinerja aplikasi.

Matriks kebingungan adalah fase selanjutnya. Matriks kebingungan menggunakan 2.645 titik data individual. Pada Tabel 4, Anda dapat mengamati hasil matriks kebingungan klasifikasi *RNN*.

Tabel 4 *Confusion matrix* Klasifikasi *RNN*

Actual/Predicted	*Predicted Negative*	*Predicted Positive*
Actual Negative	214 (*TN*)	389 (FP)
Actual Positive	8 (*FN*)	2034 (TP)

Metrik penilaian dapat diturunkan dari matriks kebingungan yang diberikan di atas. Perhitungan akurasi menggunakan Persamaan (1), yaitu:

$$Accuracy = \frac{2034 + 214}{2034 + 214 + 389 + 8} \approx 85\%$$

Perhitungan tersebut menunjukkan bahwa 85% dari seluruh prediksi yang dilakukan model adalah benar. Selanjutnya untuk label positif, presisi dihitung menggunakan Persamaan (2), adalah sebagai berikut:

$$Precision = \frac{2034}{2034 + 389} \approx 84\%$$

Nilai ini berarti bahwa dari semua prediksi positif yang dihasilkan model, 84% di antaranya benar-benar positif. Selanjutnya, nilai *recall* diperoleh menggunakan Persamaan (3), adalah sebagai berikut:

$$Recall = \frac{2034}{2034 + 8} \approx 100\%$$

Hasil perhitungan *recall* tersebut menunjukkan bahwa hampir semua data positif berhasil terdeteksi oleh model. Terakhir, skor *F1* dihitung menggunakan Persamaan (4), adalah sebagai berikut:

$$F1 - Score = 2\,\mathrm{X}\frac{0.84 \text{ x } 1.00}{0.84 + 1.00} \approx 91\%$$

Penilaian model menggunakan laporan klasifikasi mengikuti penemuan matriks kebingungan dan metrik penilaian. Tabel 5 menunjukkan laporan kategorisasi hasil uji *RNN*.

Tabel 5. *Classification report* Klasifikasi *RNN*

	Precision	*Recall*	*F1-score*	Support
Negatif	0.96	0.35	0.52	603
Positif	0.84	1.00	0.91	2042
Accuracy			0.85	2645
Macro avg	0.90	0.68	0.71	2645

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Weighted avg	0.87	0.85	0.82	2645

Sentimen ulasan dievaluasi menggunakan model Jaringan *RNN*, dan temuannya ditunjukkan pada Tabel 5. Sebagian besar evaluasi diklasifikasikan secara akurat oleh model, karena mencapai akurasi keseluruhan sebesar 85%.

Meskipun model menunjukkan akurasi yang hampir sempurna dalam memprediksi ulasan yang tidak menguntungkan (Presisi = 0,96), model tersebut masih kurang dalam menangkap semua ulasan yang sebenarnya negatif (*Recall* = 0,35). Terdapat ruang untuk peningkatan kinerja model pada emosi negatif, sebagaimana ditunjukkan oleh skor *F1* sebesar 0,52 untuk area ini.

Di sisi lain, akurasi model (0,84) dan *Recall* (1,000) sangat baik untuk evaluasi bersentimen positif. Oleh karena itu, dapat disimpulkan bahwa model ini tidak hanya memprediksi secara akurat tetapi juga menangkap hampir semua evaluasi positif. Kualitas prediksi yang sangat baik untuk emosi positif ditunjukkan oleh skor *F1* sebesar 0,91 untuk kategori ini.

3.4.2 Klasifikasi Sentimen Model *LSTM*

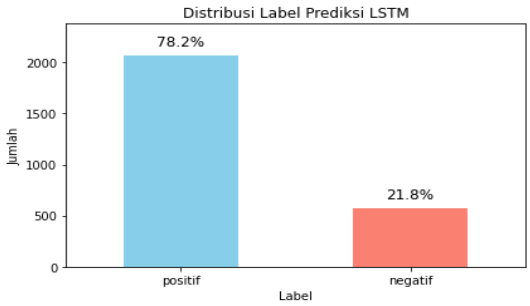
Pada pemodelan berikutnya, algoritma yang digunakan adalah *Long Short-Term Memory* (*LSTM*). Data yang digunakan sama seperti sebelumnya, yaitu hasil *preprocessing* dan *labeling* ulasan, yang disimpan dalam file *CSV*. Fitur yang digunakan adalah hasil *tokenizing* dan *stemming* dari teks ulasan, sedangkan label terdiri dari dua kelas, yaitu positif dan negatif. Tokenisasi teks dilakukan menggunakan *Keras Tokenizer* dengan batas maksimal 1000 kata paling sering muncul. Teks ulasan diubah menjadi sekuens angka melalui *texts_to_sequences*(), kemudian diproses menggunakan *Padding* agar seluruh data memiliki panjang seragam, yaitu 100 token.

Label sentimen dikonversikan menjadi angka menggunakan LabelEncoder dari *library sklearn*. Selanjutnya, data dibagi menggunakan train_test_split menjadi data latih 80% dan data uji 20% agar distribusi data lebih proporsional.

Arsitektur model *LSTM* yang dibangun terdiri dari beberapa lapisan, dimulai dari Embedding *Layer* dengan input maksimal 1000 kata, dimensi output 128, dan panjang input 100 token. Setelah itu, model dilanjutkan dengan *LSTM Layer* yang memiliki 128 unit neuron dengan fungsi aktivasi tanh. Kemudian, ditambahkan dua *Dense Layer* berturut-turut dengan jumlah neuron 64 dan 32 yang menggunakan fungsi aktivasi *ReLU*. Pada lapisan output, digunakan *Dense Layer* dengan 1 neuron yang memakai aktivasi sigmoid, karena klasifikasi ini bersifat biner antara positif dan negatif.

Model dikompilasi menggunakan optimizer *RMSprop*, *binary_crossentropy* sebagai *loss function*, dan metrik *Accuracy*. Untuk memastikan bobot terbaik dari model, digunakan *ModelCheckpoint* yang secara otomatis akan menyimpan model dengan akurasi validasi tertinggi. Proses pelatihan dilakukan selama 20 *epoch* dengan *batch size* 32, dan validasi dilakukan pada data *X_test* dan *y_test*.

Untuk lebih memahami distribusi data sentimen, temuan klasifikasi akan divisualisasikan sebagai tahap selanjutnya dalam proses analisis. Visualisasi data ini membandingkan jumlah ulasan untuk kategori positif dan negatif dan ditampilkan sebagai diagram batang. Diagram batang yang dihasilkan terdapat pada Gambar 5.



Gambar 5 *Barplot* Data Hasil Klasifikasi *LSTM*

Langkah selanjutnya adalah memvisualisasikan kata-kata yang paling sering muncul dalam data ulasan. Visualisasi ini dilakukan menggunakan *Wordcloud*, yang berguna untuk memberikan gambaran secara visual mengenai kata yang dominan dalam masing-masing kategori sentimen. Worcloud bisa dilihat pada gambar 6 dan gambar 7.

Gambar 6 *Wordcloud* Positif *LSTM*

Gambar 6 menunjukkan *Wordcloud* untuk kategori sentimen negatif. Berdasarkan visualisasi tersebut, kata-kata yang paling sering muncul antara lain “tolong”, “gak”, “update”, “lokasi”, “gempa”, dan “buka”. Kata-kata ini menunjukkan keluhan yang sering muncul dari pengguna aplikasi Info BMKG.

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Gambar 7 *Wordcloud* Negatif *LSTM*

Gambar 7 menampilkan *Wordcloud* untuk kategori sentimen positif. Kata-kata dominan yang muncul meliputi "bantu", "bagus", "mantap", "informasi", "aplikasi", dan "cuaca". Kata-kata tersebut mencerminkan apresiasi positif dari pengguna terhadap aplikasi yang memberikan manfaat dalam informasi kebencanaan dan cuaca.

Matriks kebingungan adalah fase selanjutnya. Matriks kebingungan menggunakan total 2.645 data. Tabel 6 menunjukkan temuan matriks kebingungan klasifikasi *LSTM*.

Tabel 6. *Confusion matrix* Klasifikasi *LSTM*

Actual/Predicted	*Predicted Negative*	*Predicted Positive*
Actual Negative	458 (*TN*)	145 (FP)
Actual Positive	119 (*FN*)	1923 (TP)

Metrik penilaian pada penelitian ini diturunkan dari *confusion matrix* yang telah disajikan sebelumnya. Perhitungan akurasi menggunakan Persamaan (1) adalah sebagai berikut:

$$Accuracy = \frac{1923 + 458}{1923 + 458 + 145 + 119} \approx 90\%$$

Hasil ini menunjukkan bahwa 90% dari seluruh prediksi yang dilakukan model adalah benar. Selanjutnya untuk label positif, presisi dihitung menggunakan Persamaan (2) adalah sebagai berikut:

$$Precision = \frac{1923}{1923 + 145} \approx 93\%$$

Nilai presisi menunjukkan bahwa dari semua prediksi positif yang dihasilkan model, 93% di antaranya benar-benar positif. Selanjutnya, nilai recall diperoleh menggunakan Persamaan (3) adalah sebagai berikut:

$$Recall = \frac{1923}{1923 + 119} \approx 94\%$$

Hasil *recall* model berhasil mendeteksi 94% dari seluruh data positif yang ada. Terakhir, skor *F1* dihitung menggunakan Persamaan (4) adalah sebagai berikut:

$$F1 - Score = 2\,\mathrm{X}\frac{0.93\ \mathrm{x}\ 0.94}{0.94 + 0.94} \approx 94\%$$

Penilaian model menggunakan laporan klasifikasi dilakukan setelah penemuan matriks kebingungan dan metrik penilaian. Tabel 7 menampilkan laporan kategorisasi yang diperoleh dari hasil uji *LSTM*.

Kinerja model *LSTM* dalam mengidentifikasi sentimen ulasan positif dan negatif terdapat pada Tabel 7. Secara umum, tingkat akurasi model sebesar 90% menunjukkan bahwa model tersebut secara akurat mengklasifikasikan sebagian besar evaluasi. Mayoritas prediksi negatif untuk ulasan dengan sentimen negatif sesuai dengan kategori sebenarnya, tetapi tidak persis, dengan Presisi 0,79. Model ini berhasil menangkap sebagian besar ulasan negatif, dengan *Recall* 0,76. Terdapat kompromi yang sehat antara Presisi dan *Recall* dengan skor *F1* 0,78 dalam kelompok ini.

Tabel 7. *Classification report* Klasifikasi *LSTM*

	Precision	*Recall*	*F1-score*	Support
Negatif	0.79	0.76	0.78	603
Positif	0.93	0.94	0.94	2042
Accuracy			0.90	2645
Macro avg	0.86	0.85	0.86	2645
Weighted avg	0.90	0.90	0.90	2645

Di sisi lain, akurasi dan *Recall* model sama-sama 0,94 untuk evaluasi bernada positif. Algoritme ini dapat menangkap hampir semua evaluasi positif dan cukup andal dalam memprediksinya. Dengan skor *F1* 0,94, prediksi kategori ini untuk emosi positif secara konsisten benar.

3.5 Prediksi Sentimen Menggunakan *RNN* dan *LSTM*

Pada penelitian ini, selain melakukan klasifikasi sentimen, peneliti juga melakukan prediksi jumlah ulasan sentimen positif dan negatif pada aplikasi Info BMKG dari Mei 2025 sampai Desember 2026. Prediksi ini bertujuan untuk melihat kecenderungan jumlah ulasan berdasarkan sentimen secara periodik dari waktu ke waktu. Prediksi dilakukan menggunakan dua metode yang berbeda, yaitu *Recurrent Neural network* (*RNN*) dan *Long Short-Term Memory* (*LSTM*). Data yang digunakan merupakan hasil agregasi jumlah ulasan per bulan yang telah dipisahkan berdasarkan kategori sentimen positif dan negatif dari bulan Mei 2012 hingga bulan Mei 2025. Dengan demikian, model prediksi dapat mempelajari pola *time series* dari jumlah ulasan secara lebih spesifik.

3.5.1 Prediksi Sentimen Model *RNN*

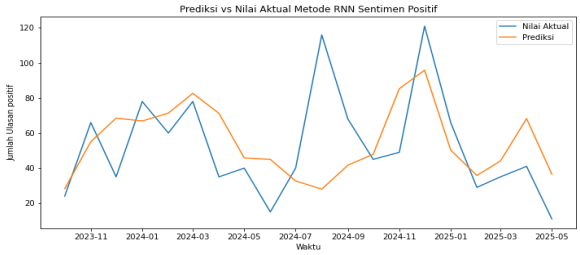
Model *RNN* mengantisipasi jumlah ulasan dengan dua cara berbeda: positif dan negatif. Data dibagi menjadi dua bagian: 80% untuk pelatihan dan 20% untuk

P-ISSN : 2089-3353
E-ISSN : 2808-9162

pengujian. Data dinormalisasi menggunakan MinMaxScaler sebelum pemodelan untuk menjaga pembelajaran model tetap stabil dan mencegah ketidakseimbangan skala. Model *RNN* yang digunakan memiliki arsitektur dengan tiga lapisan *SimpleRNN* berurutan, masing-masing terdiri dari 128, 64, dan 32 neuron dengan aktivasi *ReLU*. Setelahnya, model dilengkapi dengan lapisan *Dense* berjumlah neuron 32 dan 16, serta satu lapisan output untuk memprediksi satu nilai numerik. Model dikompilasi dengan fungsi *loss Mean Squared Error* (*MSE*) dan optimizer *RMSprop*. Selama proses pelatihan, digunakan teknik *ModelCheckpoint* untuk menyimpan bobot terbaik berdasarkan nilai *loss* terkecil pada data validasi. Pelatihan dilakukan selama 100 *epoch* dengan *batch size* sebesar 16.

Hasil evaluasi model *RNN* untuk prediksi ulasan sentimen positif menghasilkan nilai *Mean Squared Error* (*MSE*) sebesar 1181.3669, *Mean Absolute Error* (*MAE*) sebesar 27.0639, dan *R-Squared* (R^2) sebesar -0.4224. Sementara itu, hasil evaluasi pada ulasan sentimen negatif menunjukkan *MSE* sebesar 104.9717, *MAE* sebesar 7.6132, dan *R-Squared* (R^2) sebesar 0.2198. Meskipun kinerja prediksi keseluruhan masih relatif fluktuatif, temuan ini menunjukkan bahwa model *RNN* unggul dalam prediksi kategori negatif dibandingkan dengan prediksi kategori positif.

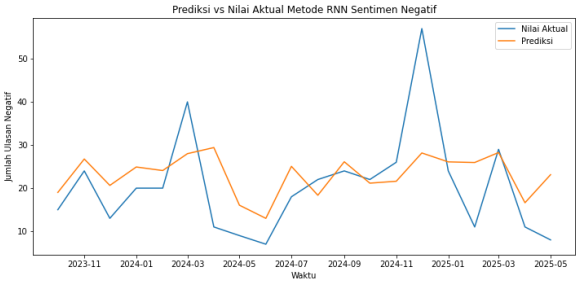
Untuk mengetahui sejauh mana model *RNN* mampu memprediksi jumlah ulasan sentimen positif dan negatif, dilakukan visualisasi antara nilai aktual dengan nilai prediksi. Visualisasi ini bertujuan untuk memberikan gambaran secara langsung terkait performa model dalam mengikuti pola data historis.



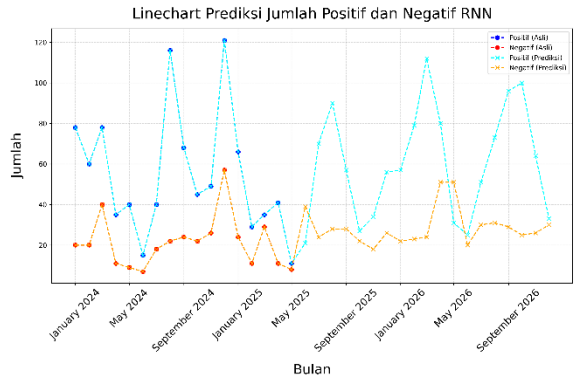
Gambar 8. Visualisasi Prediksi vs Nilai Aktual Metode *RNN* Sentimen Positif

Pada gambar 8 terlihat perbandingan antara nilai aktual dan hasil prediksi jumlah ulasan sentimen positif menggunakan model *RNN*. Garis biru mewakili data aktual, sementara garis oranye mewakili hasil prediksi model. Secara umum, model *RNN* mampu mengikuti tren naik-turun data, namun tidak mampu menangkap fluktuasi yang lebih ekstrem. Misalnya, pada puncak data aktual yang sangat tinggi, prediksi model *RNN* cenderung lebih stabil dan tidak mencapai puncak yang sama. Hal ini menunjukkan bahwa *RNN* memiliki keterbatasan dalam memodelkan pola data *time series* yang sangat fluktuatif atau mengandung *outlier*. Namun, untuk pola naik-turun yang lebih ringan, prediksi *RNN* masih cukup mendekati pola data sebenarnya.

Gambar 9 merupakan perbandingan antara nilai aktual dan hasil prediksi jumlah ulasan sentimen negatif menggunakan model *RNN*. Sama seperti grafik sebelumnya, garis biru menunjukkan data aktual sedangkan garis oranye menunjukkan hasil prediksi. Dari grafik ini terlihat bahwa model *RNN* kesulitan mengikuti lonjakan-lonjakan data negatif yang tajam. Data aktual cenderung mengalami fluktuasi yang lebih ekstrem, sedangkan prediksi model *RNN* lebih datar dan cenderung rata mengikuti rata-rata pola sebelumnya. Hal ini mengindikasikan bahwa model *RNN* cenderung memberikan prediksi konservatif dan kesulitan mengikuti pola data *time series* negatif yang volatil.



Gambar 9. Visualisasi Prediksi vs Nilai Aktual Metode *RNN* Sentimen Negatif



Gambar 10 Visualisasi *Linechart* Prediksi Jumlah Positif dan Negatif *RNN*

Gambar 10 menunjukkan visualisasi jumlah data sentimen positif dan negatif berdasarkan hasil prediksi model *Recurrent Neural network* (*RNN*) dalam bentuk *linechart*. Sumbu horizontal merepresentasikan rentang waktu dari bulan Januari 2024 hingga bulan Desember 2026, sedangkan sumbu vertikal menunjukkan jumlah data pada masing-masing waktu. Data yang ditampilkan terdiri dari data asli dan hasil prediksi, dengan legenda yang membedakan masing-masing jenis data: positif asli (garis biru), negatif asli (garis merah), positif prediksi (garis *cyan*), dan negatif prediksi (garis oranye). Secara keseluruhan, grafik ini memberikan gambaran mengenai performa model *RNN* dalam memprediksi tren jumlah sentimen dari waktu ke

P-ISSN : 2089-3353
E-ISSN : 2808-9162

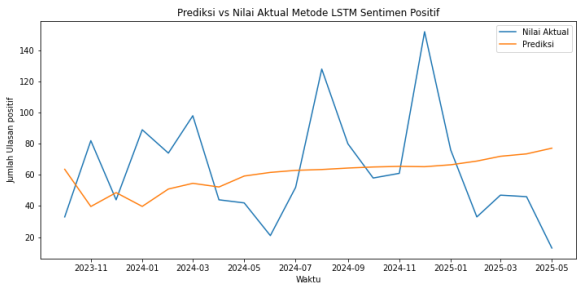
waktu serta menunjukkan adanya kesesuaian pola antara data asli dan prediksi dalam beberapa periode, meskipun terdapat pula beberapa perbedaan signifikan. Grafik ini juga bermanfaat untuk melihat potensi tren sentimen di masa mendatang dan mengevaluasi efektivitas model dalam memetakan data historis dan proyeksi.

3.5.2 Prediksi Sentimen Model *LSTM*

Data dibagi menjadi data latih sebesar 80% dan data uji sebesar 20%, dengan proses windowing selama 12 bulan untuk memprediksi bulan selanjutnya. Data kemudian diubah bentuknya menjadi tiga dimensi sesuai kebutuhan arsitektur *LSTM*. Model *LSTM* dirancang dengan tiga lapisan *LSTM* berurutan, yaitu *LSTM* 64 neuron dengan return sequences, *LSTM* 32 neuron dengan return sequences, dan *LSTM* 32 neuron tanpa return sequences. Setiap lapisan *LSTM* dilengkapi dengan *Dropout* 0.2 untuk mengurangi risiko overfitting. Setelahnya, model memiliki lapisan *Dense* sebanyak 64 neuron, 16 neuron, dan satu neuron output. Model dikompilasi dengan fungsi *loss Mean Squared Error* (*MSE*) dan optimizer *RMSprop*. Pada proses pelatihan, *ModelCheckpoint* digunakan untuk menyimpan bobot model terbaik berdasarkan *loss* terkecil di data validasi. Pelatihan dilakukan selama 100 *epoch* dengan *batch size* sebesar 16.

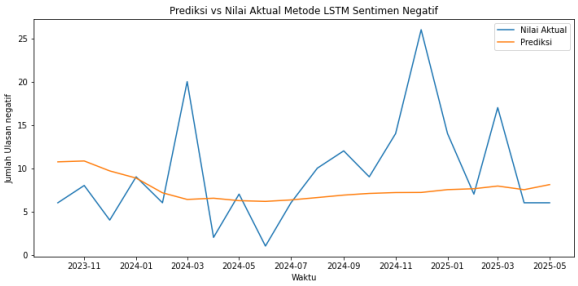
Prediksi jumlah ulasan sentimen positif dan negatif juga dilakukan menggunakan algoritma *LSTM* dengan data historis bulanan yang telah dinormalisasi dan dibentuk sebagai data *time series*. Evaluasi model menggunakan *MSE*, *MAE*, dan R^2. Hasil untuk sentimen positif menunjukkan *MSE* sebesar 1354.9937, *MAE* sebesar 26.9058, dan R^2 sebesar -0.1834. Sementara untuk sentimen negatif, diperoleh *MSE* sebesar 43.2527, *MAE* sebesar 5.2271, dan R^2 sebesar -0.1998. Nilai R^2 negatif menandakan model kurang mampu menjelaskan variansi data.

Untuk mengetahui sejauh mana model *LSTM* mampu memprediksi jumlah ulasan dengan sentimen positif dan negatif, dilakukan visualisasi antara nilai aktual dan nilai prediksi. Visualisasi ini bertujuan untuk memberikan gambaran secara langsung mengenai performa model dalam mengikuti pola data historis. Hasil visualisasi perbandingan antara nilai prediksi dan nilai aktual menggunakan metode *LSTM* untuk sentimen positif dan negatif dapat dilihat pada Gambar 12 dan 13.



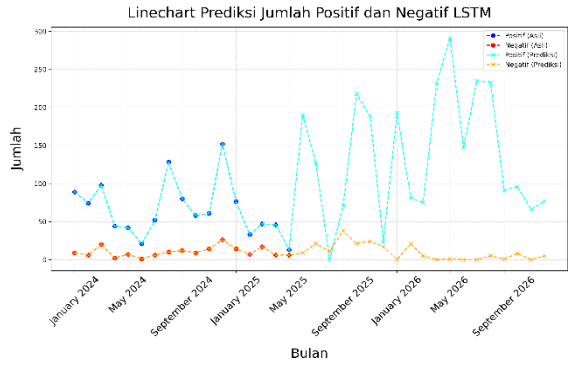
Gambar 12. Visualisasi Prediksi vs Nilai Aktual Metode *LSTM* Sentimen Positif

Gambar 12 menampilkan hasil prediksi jumlah ulasan dengan sentimen positif menggunakan model *LSTM* yang dibandingkan dengan nilai aktual. Pada grafik tersebut, terlihat bahwa nilai aktual memiliki pola fluktuatif dengan kenaikan dan penurunan yang tajam di beberapa titik, sedangkan garis prediksi dari model *LSTM* cenderung halus dan membentuk tren yang lebih stabil. Model tampak mengikuti pola umum dari data, namun tidak mampu menangkap lonjakan-lonjakan ekstrem yang terjadi pada nilai aktual. Hal ini menunjukkan bahwa model *LSTM* cukup baik dalam menggambarkan tren keseluruhan untuk sentimen positif, tetapi kurang responsif terhadap perubahan mendadak.



Gambar 13. Visualisasi Prediksi vs Nilai Aktual Metode *LSTM* Sentimen Negatif

Sementara itu, Gambar 13 menunjukkan hasil prediksi jumlah ulasan dengan sentimen negatif. Sama seperti grafik sebelumnya, nilai aktual menunjukkan fluktuasi yang tinggi, namun perbedaannya terletak pada bagaimana model memprediksi tren. Pada sentimen negatif, garis prediksi terlihat lebih datar dan kurang mengikuti perubahan arah dari data aktual. Model *LSTM* terlihat lebih konservatif dalam merespons variasi data negatif dibandingkan dengan data positif, sehingga menghasilkan prediksi yang lebih rata. Perbedaan ini menunjukkan bahwa model *LSTM* sedikit lebih baik dalam menangkap tren sentimen positif dibandingkan sentimen negatif, meskipun pada keduanya model tetap menunjukkan kecenderungan meratakan prediksi dan belum mampu merepresentasikan dinamika data dengan akurat.



Gambar 14. Visualisasi *Linechart* Prediksi Jumlah Positif dan Negatif *LSTM*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Gambar 14 memperlihatkan visualisasi hasil prediksi jumlah ulasan dengan sentimen positif dan negatif menggunakan model *LSTM* yang dibandingkan dengan data aktual dalam rentang waktu dari bulan Januari tahun 2024 hingga bulan Desember 2026. Pada grafik ini, garis biru merepresentasikan jumlah ulasan aktual dengan sentimen positif, sedangkan garis merah menunjukkan jumlah ulasan aktual dengan sentimen negatif. Adapun hasil prediksi model *LSTM* ditampilkan dalam garis biru muda untuk sentimen positif dan garis oranye untuk sentimen negatif. Berdasarkan visualisasi tersebut, dapat dilihat bahwa jumlah ulasan dengan sentimen positif umumnya lebih tinggi dibandingkan dengan sentimen negatif. Sementara itu, ulasan dengan sentimen negatif cenderung lebih stabil dengan variasi yang tidak terlalu besar, dan prediksi model untuk kategori ini juga menunjukkan pola yang lebih datar.

4. Kesimpulan

Long Short-Term Memory (*LSTM*) mengungguli jaringan *Recurrent Neural Network* (*RNN*) dalam kategorisasi sentimen. Akurasi, *Recall*, dan skor *F1* untuk model *LSTM* adalah 0,90, yang menghasilkan tingkat keberhasilan 90%. Di sisi lain, model *RNN* mencapai akurasi 85%, presisi 0,87, *Recall* 0,85, dan skor *F1* 0,82. Meskipun akurasi *RNN* sangat baik (0,96) untuk kategori sentimen negatif, nilai *Recall*-nya (0,35) sangat rendah, menunjukkan bahwa model tersebut kesulitan menangkap seluruh spektrum evaluasi negatif secara akurat. Namun, dalam kategorisasi dua kelas, model *LSTM* secara keseluruhan berkinerja lebih baik, dengan *Recall* 0,76 dan presisi 0,79 untuk kategori negatif.

Dalam hal prediksi jumlah ulasan berdasarkan sentimen, model *RNN* menunjukkan performa yang lebih baik untuk prediksi kategori sentimen negatif dengan nilai *Mean Squared Error* (*MSE*) sebesar 104.97, *Mean Absolute Error* (*MAE*) sebesar 7.61, dan *R-Squared* (R^2) sebesar 0.22. Sebaliknya, model *LSTM* menghasilkan *MSE* sebesar 43.25, *MAE* sebesar 5.23, namun memiliki nilai R^2 negatif (-0.20), menandakan rendahnya kemampuan model dalam menjelaskan variabilitas data. Untuk prediksi ulasan positif, kedua model menunjukkan performa yang kurang memadai, ditandai dengan nilai R^2 yang rendah atau negatif. Secara umum, dapat disimpulkan bahwa model *LSTM* lebih unggul dalam hal klasifikasi sentimen karena kemampuannya menjaga keseimbangan performa antar kelas, sedangkan model *RNN* menunjukkan kinerja yang sedikit lebih stabil dalam memprediksi jumlah ulasan negatif, meskipun keduanya masih menghadapi tantangan dalam menangani data historis dengan pola yang fluktuatif.

Daftar Rujukan

[1] B. T. Ariyanto, Y. Ruldeviyani, G. S. B. Dharmawan, and M. I. F. Bahar, "Optimizing User Satisfaction: A Comprehensive Evaluation of the Info BMKG App Using UEQ+ and IPA Methods," *International Journal of Economics (IJEC),* vol. 3, no. 1, pp. 139–156, 2024, doi: 10.58291/ijec.v3i1.242.

[2] I. M. Karo Karo, J. A. Karo Karo, Y. Yunianto, H. Hariyanto, M. Falah, and M. Ginting, "Analisis Sentimen Ulasan Aplikasi Info BMKG Di Google Play Menggunakan TF-IDF Dan *Support Vector Machine*," *Multitek Indonesia,* vol. 4, no. 4, pp. 1423–1430, 2023, doi: 10.47065/josh.v4i4.3943.

[3] D. Pratmanto, F. F. D. Imaniawan, and V. Maarif, "Analisis Sentimen Pada Ulasan Pengguna Aplikasi Identitas Kependudukan Digital Dengan Metode Naive Bayes Dan K-Nearest," *Computatio: Journal of Computer Science and Information Systems,* vol. 7, no. 2, pp. 155–166, 2023, doi: 10.24912/computatio.v7i2.26322.

[4] T. A. Zuraiyah, M. M. Mulyati, and G. H. F. Harahap, "Perbandingan Metode *Naïve Bayes*, *Support Vector Machine* Dan *Recurrent Neural network* Pada Analisis Sentimen Ulasan Produk E-Commerce," *Multitek Indonesia,* vol. 17, no. 1, pp. 28–44, 2023, doi: 10.24269/mtkind.v17i1.7092.

[5] M. V. Shyahrin, Y. Sibaroni, and D. Puspandari, "Penerapan Metode *Long Short-Term Memory* Dan Word2Vec Dalam Analisis Sentimen Ulasan Pada Aplikasi Ferizy," *Techno.Com,* vol. 22, no. 4, pp. 833–842, 2023, doi: 10.33633/tc.v22i4.9205.

[6] A. Ligthart, C. Catal, and B. Tekinerdogan, "Systematic Reviews in Sentiment Analysis: A Tertiary Study," *Artificial Intelligence Review,* vol. 54, no. 7, pp. 4997–5053, 2021, doi: 10.1007/s10462-021-09973-3.

[7] A. R. Harungguan, H. Napitupulu, and F. Firdaniza, "Analisis Sentimen Dengan Metode Klasifikasi *Naïve Bayes* Dan Seleksi Fitur Chi-Square," *In Search (Informatic, Science, Entrepreneur, Applied Art, Research, Humanism),* vol. 22, no. 2, pp. 332–339, 2023, doi: 10.37278/insearch.v22i2.762.

[8] I. Gede Bintang Arya Budaya, L. Putu Safitri Pratiwi, and D. Panji Agustino, "Klasifikasi Sentimen Untuk Analisis Kepuasan Pelayanan Puskesmas Berbasis Arsitektur *LSTM*," *Smart Comp: Jurnalnya Orang Pintar Komputer,* vol. 12, no. 4, 2023, doi: 10.30591/smartcomp.v12i4.5361.

[9] S. Jurnalis Pipin and H. Kurniawan, "Analisis Sentimen Kebijakan MBKM Berdasarkan Opini Masyarakat Di Twitter Menggunakan *LSTM*," *Jurnal SIFO Mikroskil,* vol. 23, no. 2, pp. 197–208, 2022, doi: 10.55601/jsm.v23i2.900.

[10] A. Nurian, M. S. Ma'arif, I. N. Amalia, and C. Rozikin, "Analisis Sentimen Pengguna Aplikasi Shopee Pada Situs Google Play Menggunakan Naive Bayes Classifier," *Jurnal Informatika dan Teknik Elektro Terapan (JITET),* vol. 12, no. 1, 2024, doi: 10.23960/jitet.v12i1.3631.

[11] S. Sisnawati, R. Astuti, and F. Muhamad basysyar, "Analisis Sentimen Pada Aplikasi Kfcku Di Google Playstore Menggunakan *Naïve Bayes*," *JATI (Jurnal Mahasiswa Teknik Informatika),* vol. 8, no. 3, pp. 3010–3016, 2024, doi: 10.36040/jati.v8i3.8382.

[12] D. P. Prabowo, R. A. Pramunendar, and R. A. Megantara, "Prediksi Sentimen Masyarakat Terhadap Penggunaan Vaksin Covid 19 Menggunakan *RNN*," *Jurnal Informatika UPGRIS,* vol. 8, no. 1, pp. 42–45, 2022, doi: 10.26877/jiu.v8i1.11599.

[13] A. R. Isnain, H. Sulistiani, B. M. Hurohman, A. Nurkholis, and S. Styawati, "Analisis Perbandingan Algoritma *LSTM* Dan Naive Bayes Untuk Analisis Sentimen," *Jurnal Edukasi dan Penelitian Informatika (JEPIN),* vol. 8, no. 2, p. 299, 2022, doi: 10.26418/jp.v8i2.54704.

[14] A. Fuad and M. Al-Yahya, "Analysis and Classification of Mobile Apps Using Topic Modeling: A Case Study on Google Play Arabic Apps," *Complexity,* vol. 2021, no. 1, 2021, doi: 10.1155/2021/6677413.

[15] S. Wang, J. Ren, and R. Bai, "A Regularized Attribute Weighting Framework for Naive Bayes," *IEEE Access*, vol. 8, pp. 225639–225649, 2020, doi: 10.1109/ACCESS.2020.3044946.

[16] A. S. Berliana and M. Mustikasari, "Analisis Sentimen Pada Ulasan Aplikasi Jakartanotebook Di Google Play Menggunakan Metode *Recurrent Neural network* (*RNN*)," *Jurnal Informatika dan Teknik Elektro Terapan (JITET),* vol. 12, no. 3, 2024, doi: 10.23960/jitet.v12i3.5067.

[17] C. K. Aridas, S. Karlos, V. G. Kanas, N. Fazakis, and S. B. Kotsiantis, "Uncertainty Based Under-Sampling for Learning Naive Bayes Classifiers under Imbalanced Data Sets," *IEEE*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Access, vol. 8, pp. 2122–2133, 2020, doi: 10.1109/ACCESS.2019.2961784.

[18] H.-C. Kuo *et al.*, "Penggunaan Software Orange Data Mining Pada Implementasi Text Mining Dalam Analisis Sentimen Netizen Di Twitter Terhadap Kelangkaan Minyak Goreng," *Jurnal Ilmiah Informatika (JIF),* vol. 4, no. 1, p. 1, 2023, doi: 10.33884/jif.v11i01.6611.

[19] M. S. Islam *et al.*, "Challenges and Future in *Deep learning* for Sentiment Analysis: A Comprehensive Review and a Proposed Novel Hybrid Approach," *Artificial Intelligence Review,* vol. 57, no. 3, 2024, doi: 10.1007/s10462-023-10651-9.

[20] V. Mironov, A. Gusarenko, and G. Tuguzbaev, "Graphic Documents Parametric Personalization for Information Support of Educational Design Using Situation-Oriented Databases," 2020, doi: 10.2991/aisr.k.201029.050.

[21] V. Balakrishnan, Z. Shi, C. L. Law, R. Lim, L. L. Teh, and Y. Fan, "A *Deep learning* Approach in Predicting Products' Sentiment Ratings: A Comparative Analysis," *Journal of Supercomputing,* vol. 78, no. 5, pp. 7206–7226, 2021, doi: 10.1007/s11227-021-04169-6.

[22] S. J. Pipin, R. Purba, and H. Kurniawan, "Prediksi Saham Menggunakan *Recurrent Neural network* (*RNN-LSTM*) Dengan Optimasi Adaptive Moment Estimation," *J. Comput. Syst. Informatics*, vol. 4, no. 4, pp. 806–815, 2023, doi: 10.47065/josyc.v4i4.4014.

[23] N. Moch Farryz Rizkilloh and N. Sri Widiyanesti, "Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (*LSTM*)," *J. RESTI (Rekayasa Sist. Dan Teknol. Informasi)*, vol. 6, no. 1, pp. 25–31, 2022, doi: 10.29207/resti.v6i1.3630.

[24] Y. Kryvenchuk, N. Horishna, and N. None, "Creation of Salary Prediction System," *Herald of Khmelnytskyi National University. Technical Sciences,* vol. 317, no. 1, pp. 276–279, 2023, doi: 10.31891/2307-5732-2023-317-1-276-279.

[25] N. Selle, N. Yudistira, and C. Dewi, "Perbandingan Prediksi Penggunaan Listrik Dengan Menggunakan Metode Long Short Term Memory (*LSTM*) Dan *Recurrent Neural network* (*RNN*)," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 9, no. 1, pp. 155–162, 2022, doi: 10.25126/jtiik.2022915585.