JURNAL ILMIAH TEKNOLOGI DAN REKAYASA
Vol. 30, No. 02, August 2025, 119 – 135
p-ISSN: 1410-9093, e-ISSN: 2089-8088
https://ejournal.gunadarma.ac.id/index.php/tekno | tekrek@gunadarma.ac.id

Analisis Sentimen Terhadap Aplikasi M-Paspor Menggunakan Algoritma *Long Short-Term Memory* (LSTM) dan BERT *Embedding*

[1*]Bambang Gunawan Hardianto, [2]Satrio Wibisono

[1,2]Sistem Informasi Bisnis, Magister Manajemen Sistem Informasi, Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat, Indonesia
[1]bambang_gunawan@staff.gunadarma.ac.id, [2]swibisono40@gmail.com

Abstract

The Directorate General of Immigration launched the M-Paspor app for online passport applications. Although it has been downloaded more than 1 million times, its 2.5 rating indicates user dissatisfaction. Review analysis is necessary for developers to understand the issues and improve the app to enhance the overall user experience. Therefore, this study conducts sentiment analysis on M-Paspor app reviews using a combination of Bidirectional Encoder Representations from Transformers (BERT) embedding and LSTM to classify user opinions. The advantage of BERT lies in its ability to understand the deep context of text, while LSTM excels in handling sequential data. LSTM is used as a classification method because it can capture long-term patterns in sequential data through memory management with cell states and three main gates, enabling it to continuously understand sentence context to support sentiment analysis. In this study, labeling consists of three classes: positive, negative, and neutral, using the lexicon method. The LSTM-BERT model shows consistent and higher accuracy values with a smaller proportion of training data. Testing results with a confusion matrix show the highest accuracy of 91.33% on a 70%:30% data split.

Keywords: BERT embedding, LSTM, M-Passport application, sentiment analysis

Abstrak

Direktorat Jenderal Imigrasi meluncurkan aplikasi M-Paspor untuk pengajuan paspor *online*. Meski telah diunduh lebih dari 1 juta kali, rating 2.5 menunjukkan ketidakpuasan pengguna. Analisis ulasan diperlukan agar pengembang memahami masalah dan memperbaiki aplikasi demi meningkatkan pengalaman pengguna secara keseluruhan. Oleh karena itu, pada penelitian ini dilakukan analisis sentimen terhadap ulasan aplikasi M-Paspor dengan menggunakan kombinasi Bidirectional Encoder Representations from Transformers (BERT) *embedding* dan LSTM untuk mengklasifikasikan opini pengguna. Keunggulan BERT mampu memahami konteks mendalam dari teks dan kekuatan LSTM dalam menangani urutan data. Penggunaan LSTM sebagai metode klasifikasi dikarenakan mampu menangkap pola jangka panjang dalam data sekuensial melalui pengelolaan memori dengan *cell state* dan 3 (tiga) gerbang utama, sehingga dapat memahami konteks kalimat secara berkelanjutan untuk mendukung analisis sentimen. Adapun pada penelitian ini pelabelan terdiri dari 3 kelas yaitu positif, negatif, dan netral menggunakan metode lexicon. Model LSTM-BERT menunjukkan nilai akurasi yang konsisten dan lebih tinggi pada proporsi data latih yang lebih sedikit. Hasil pengujian dengan *confusion matrix* diperoleh akurasi tertinggi 91,33% pada split data 70%:30%.

Kata Kunci: analisis sentimen, aplikasi M-Paspor, BERT *embedding*, LSTM

Article Info: Received October 08, 2024; Accepted July 25, 2025; Published August 23, 2025.
* Corresponding Author

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi dimanfaatkan oleh sektor pemerintahan untuk meningkatkan kinerja yang lebih efektif dan efisien, salah satunya melalui kebijakan implementasi TIK oleh pemerintah Indonesia guna mewujudkan tata kelola pemerintahan yang transparan dan akuntabel [1]. Salah satu inovasi dari Direktorat Jenderal Imigrasi adalah peluncuran aplikasi M-Paspor, yang memungkinkan masyarakat mengajukan permohonan paspor baru atau pergantian secara *online* [2]. Aplikasi ini bertujuan mempermudah proses pendaftaran dan pengumpulan berkas tanpa harus datang ke kantor imigrasi, serta mendapat sambutan positif dari masyarakat [2]. Hal tersebut ditunjukkan dengan jumlah pengunduh aplikasi M-Paspor di Google Play Store yang melebihi angka dari 1 juta pengguna. Akan tetapi, jika dilihat berdasarkan ulasan dari 28 ribu pengguna pada tanggal 2 bulan Februari 2024, rata-rata *rating* yang didapatkan aplikasi M-Paspor yaitu 2,5, yang berarti aplikasi tersebut memiliki ulasan yang "buruk" dari penggunanya, sehingga diperlukan analisis fitur yang dinilai buruk oleh seluruh pengguna agar dapat diperbaiki [3]. Masalah yang dikeluhkan oleh pengguna aplikasi antara lain seperti *server* yang tidak stabil dan juga gangguan ketika pengambilan data [2]. Adanya ulasan dari setiap pengguna dapat berpengaruh pada calon pengguna sebagai bahan pertimbangan penggunaan suatu aplikasi [4]. Dengan banyaknya data ulasan yang ada pada aplikasi M-Paspor, terdapat rating yang tidak sesuai dengan ulasan yang diberikan oleh pengguna, sehingga perlu dilakukan pengolahan untuk klasifikasi opini yang dapat dilakukan oleh pengembang aplikasi M-Paspor sebagai bahan untuk perbaikan aplikasi. Dengan memahami sentimen negatif yang ada, pengembang dapat lebih fokus dalam memperbaiki area tertentu dalam aplikasi yang menjadi sumber ketidakpuasan, sehingga meningkatkan pengalaman pengguna secara keseluruhan.

Analisis sentimen merupakan proses otomatis dalam memahami, mengekstrak, dan mengolah data tekstual untuk memperoleh informasi sentimen dalam kalimat opini. Teknik ini menjadi penting dalam pembelajaran pemrosesan bahasa alami guna menarik faktor emosional dan banyak diterapkan pada *crowd opinion monitoring, business intelligence*, serta kecerdasan buatan [5, 6]. Analisis sentimen terhadap ulasan pengguna aplikasi M-Paspor menjadi salah satu proses yang digunakan untuk memahami respons masyarakat dalam penggunaan layanan digital tersebut. Dalam analisis sentimen, digunakan kombinasi BERT *embedding* dan LSTM (*Long Short-Term Memory*) sebagai metode klasifikasi karena LSTM mampu memodelkan hubungan jangka panjang dalam data serta mengatasi kompleksitas dimensi tinggi [7]. LSTM sebagai jenis *Recurrent Neural Network* (RNN), dirancang untuk mengingat informasi jangka panjang dan pendek secara bersamaan, sehingga cocok untuk memproses data berurutan seperti teks. Keunggulannya dalam mempertahankan informasi penting tanpa mengalami vanishing gradient menjadikannya efektif untuk klasifikasi teks yang memerlukan pemahaman konteks penuh, seperti ulasan pengguna [7].

Penggunaan BERT pada penelitian ini dikarenakan model tersebut telah menyediakan *output* berupa model *pre-trained* yang bisa diadopsi untuk berbagai tugas NLP (*Natural Language Processing*), seperti klasifikasi teks [8]. Kombinasi BERT *embedding* dengan LSTM dinilai menguntungkan karena dapat meningkatkan performa model [9]. Proses pelabelan pada penelitian ini akan menggunakan metode Lexicon InSet yang dikembangkan pada tahun 2018 menggunakan data dari X, mencakup 3.609 kosakata bermuatan positif dan 6.609 kosakata bermuatan *negative* [10]. Penelitian ini dilakukan dengan tujuan mengetahui ulasan yang diberikan pengguna aplikasi M-Paspor pada Google Play Store dan melakukan pengelompokkan opini menjadi 3 kelas yaitu positif, negatif, dan netral dalam analisis

sentimen menggunakan teknik seperti *lexicon*. BERT *embedding* dan LSTM memungkinkan peneliti untuk mengidentifikasi pola dalam ulasan teks, menemukan topik utama yang dikeluhkan pengguna, seperti proses pendaftaran, pembayaran, atau masalah teknis lainnya. Analisis sentimen dapat digunakan untuk mengukur perubahan persepsi pengguna setelah pembaruan aplikasi. Rai et al. menunjukkan bahwa penerapan LSTM pada model BERT meningkatkan akurasi sebesar 2.50% dan 1,10% pada *dataset* PolitiFact dan GossipCop dibandingkan dengan model BERT Vanilla [11].

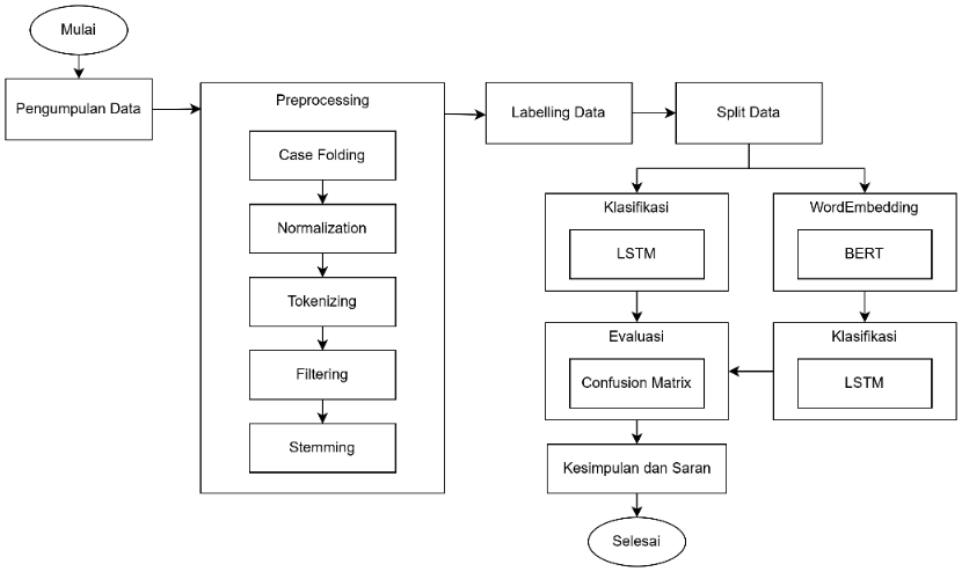
Penelitian terkait dengan analisis sentimen pernah dilakukan oleh Muhammad et al. [12]. Pada penelitian tersebut hal yang diulas adalah mengenai persepsi publik terhadap hotel Indonesia. Pada penelitian tersebut dilakukan analisis sentimen menggunakan LSTM. Hasil dari penelitian tersebut adalah diperoleh akurasi terbaik sebesar 85.96%. Penelitian [13] membahas klasifikasi *toxicity* di media sosial dengan LSTM dan *word embedding*, yang menunjukkan hasil akurasi 93% untuk kombinasi LSTM-Glove dan 94% untuk LSTM-BERT. Sementara itu, Divate [14] melakukan analisis sentimen dari berita Marathi menggunakan LSTM dan Word2Vec, yang menghasilkan nilai *recall* 100%, dan *F1-score* 72% menunjukkan efektivitas metode ini dalam mengelompokkan opini publik. Terdapat penelitian lain yang juga menerapkan LSTM seperti penelitian [15] yang membahas *review e-commerce* dengan hasil akurasi 87%. Penelitian lain dilakukan oleh Khan et al. [16] menghasilkan akurasi sebesar 99% menggunakan LSTM dalam melakukan analisis sentimen dalam bahasa Bangla. Penelitian [17] membahas pengenalan emosi menggunakan LSTM yang menghasilkan akurasi 88.47%. Penelitian yang dilakukan oleh Chakraborty et al. [18] dalam ulasan terkait COVID-19 menggunakan LSTM menghasilkan akurasi sebesar 84.46%. Penelitian lain dilakukan oleh Abdelwahab et al. [19] dalam analisis sentimen bahasa Arab menggunakan LSTM menghasilkan akurasi sebesar 79%.

Berdasarkan uraian di atas, pemilihan BERT dan LSTM didasarkan pada keunggulan masing-masing yaitu BERT dalam merepresentasikan makna semantik dan LSTM dapat memproses data berurutan [7]. Akan tetapi, sebagian besar penelitian sebelumnya hanya menggunakan salah satu dari keduanya, atau belum mengevaluasi kombinasi metode tersebut dalam konteks aplikasi layanan publik di Indonesia. Oleh karena itu, penelitian ini memiliki kebaruan dengan menganalsis ulasan aplikasi M-Paspor menggunakan kombinasi BERT dan LSTM.

2. Metode Penelitian

Gambar 1 menggambarkan tahapan dari penelitian yang dilakukan. Setiap proses dalam tahapan tersebut menunjukkan langkah sistematis mulai dari pengumpulan data hingga evaluasi hasil analisis sentimen.

Tahap pertama dari Gambar 1 diawali dengan pengumpulan data. Dalam penelitian ini, data dikumpulkan melalui teknik *web scraping*. *Web scraping* adalah proses pengambilan informasi dari suatu situs web yang kemudian digunakan untuk melakukan analisis sentimen. Proses *web scraping* dilakukan dengan mengambil ulasan yang ditinggalkan oleh pengguna M-Paspor pada situs Google Play Store, menggunakan bantuan sebuah pustaka Python yang disebut Google Play Scraper. *Dataset* yang dimiliki terdiri dari 2000 ulasan aplikasi M-Paspor. *Dataset* ini diambil secara *scraping* pada ulasan Google Play Store aplikasi M-Paspor. Proses *scraping* pada penelitian ini dibatasi dengan data yang diambil yaitu 2000 ulasan, dan periode yang digunakan yaitu dari 2022-01-05 sampai 2024-06-25. Data yang didapatkan berupa data *reviewId, userName, userImage, content, score, thumbsUpCount, reviewCreateVersion,* dan *at*. Data yang digunakan hanya *content* dan *score.*



Gambar 1. Tahapan Penelitian

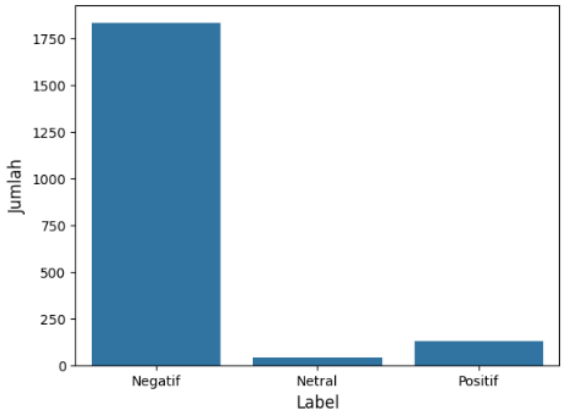
	review	rating	casefolding	normalization	tokenizing	filtering	stemming	label
0	Kenapa sampai sekarang aplikasinya masih saja ...	1	kenapa sampai sekarang aplikasinya masih saja ...	kenapa sampai sekarang aplikasinya masih saja ...	[kenapa, sampai, sekarang, aplikasinya, masih,...	aplikasinya tolong diperbaiki secepatnya penga...	aplikasi tolong baik cepat aju passport	Negatif
1	Bisa bisanya sekelas apk nasional maintenance....	1	bisa bisanya sekelas apk nasional maintenance ...	bisa bisanya sekelas aplikasi nasional mainten...	[bisa, bisanya, sekelas, aplikasi, nasional, m...	bisanya sekelas aplikasi nasional maintenance ...	bisa kelas aplikasi nasional maintenance bgt l...	Negatif
2	Aplikasi bukanya mempermudah tapi malah memper...	1	aplikasi bukanya mempermudah tapi malah memper...	aplikasi bukanya mempermudah tapi malah memper...	[aplikasi, bukanya, mempermudah, tapi, malah, ...	aplikasi bukanya mempermudah mempersulit drama...	aplikasi buka mudah sulit drama masak kuota pe...	Negatif
3	Walaupun sudah upload2 dokumen untuk persyarat...	1	walaupun sudah upload dokumen untuk persyarat...	walaupun sudah upload dokumen untuk persyarata...	[walaupun, sudah, upload, dokumen, untuk, pers...	upload dokumen persyaratan perpanjang passport...	upload dokumen syarat panjang passport potokop...	Netral

Gambar 2. *Dataframe* Hasil *Preprocessing*

Tahap kedua dilakukan *preprocessing*. *Preprocessing* merupakan tahapan pembersihan dan pengelolaan data teks yang nantinya akan dilakukan pembobotan dan analitis. Data yang terkumpul kemudian diproses melalui tahap *preprocessing*, yang terdiri dari beberapa langkah, seperti *case folding*, *normalization*, *tokenization*, *filtering*, dan *stemming*. Gambar 2 merupakan hasil *data preprocessing.*

Kemudian tahap ketiga dilakukan *labeling data* dengan menggunakan metode Lexicon InSet. Pada tahap ini dilakukan pelabelan dengan tujuan untuk mengkategorikan kalimat termasuk kategori positif, negatif, atau netral. Proses pelabelannya setiap kata diberikan skor secara manual dalam rentang -5 hingga 5, kemudian klasifikasi sentimen ditentukan berdasarkan jumlah total skor. Jika total skor lebih dari 0 maka termasuk sentimen positif, kurang dari 0 menjadi sentimen negatif, dan jika bernilai 0 maka digolongkan sebagai sentimen netral [10]. Gambar 3 merupakan hasil pelabelan yang terbentuk dengan menggunakan metode Lexicon InSet dengan detail kategori negatif sebanyak 1834, positif 127, dan netral 39.

Setelah proses pelabelan seperti yang ditunjukkan pada Gambar 3, dilakukan visualisasi 20 kata yang sering muncul. Kata-kata tersebut kemudian divisualisasi menggunakan *word cloud*, yang menggambarkan kata-kata yang sering muncul dalam teks. Ukuran kata dalam *word cloud* mencerminkan frekuensi kemunculannya, kata yang lebih sering muncul akan terlihat lebih besar. Ini mempermudah dalam mengidentifikasi kata kunci utama dengan cepat tanpa memerlukan analisis yang mendalam. Adapun hasil visualisasi *word cloud* dapat dilihat pada Gambar 4.



Gambar 3. Hasil Pelabelan

Gambar 4. Visualisasi *Word Cloud*

Berdasarkan Gambar 4, kata "aplikasi" sebagai yang paling sering muncul, diikuti oleh kata-kata seperti "daftar," "bayar," "udah," dan "paspor". Setelah itu dilanjutkan dengan proses *split* data yang merupakan proses pembagian data yang digunakan dalam penelitian. Pada penelitian ini data dibagi menjadi dua bagian yaitu data latih dan data uji. Dalam pembagian data tidak ada ketentuan rasionya, namun dari beberapa penelitian sebelumnya, rasio pembagian data yang sering digunakan adalah 70%:30%, 80%:20%, dan 90%:10%. Namun pembagian data ini penting karena diperlukan keseimbangan data. Apabila ketidakseimbangan data terjadi jika jumlah data antar kelas berbeda signifikan dan dapat menyebabkan akurasi model buruk bahkan *overfitting*, di mana model menghafal data latih termasuk *noise*, alih-alih mempelajari pola datanya [20].

Selanjutnya, tahapan penelitian dilanjutkan dengan dua langkah bercabang, yakni langsung melakukan klasifikasi menggunakan LSTM dan melakukan tahap *word embedding.* Pada penelitian ini *word embedding* menggunakan BERT. BERT memanfaatkan *encoder*, prinsip *attention*, dan membaca teks secara keseluruhan sebagai *input*, bukan urutan sekuensial. Dengan karakteristik tersebut, BERT dapat mengerti hubungan kontekstual setiap token dengan baik [21].

LSTM, sebagai pengembangan RNN dengan interaksi tambahan pada modulnya, menggunakan *blok memory-cell* (*input*, *forget*, dan *output gate*) untuk mengatasi *vanishing gradient* pada RNN. Metode ini berpotensi kuat dalam menghadapi data dimensi tinggi dan besar [7], sehingga mampu mengenali pola sentimen untuk prediksi [22].

Pembangunan model LSTM dimulai dengan tiga lapisan LSTM, masing-masing dengan 128 unit dan fungsi aktivasi *tanh*, yang menangani urutan data. Lapisan LSTM pertama dan kedua mengembalikan urutan penuh, sedangkan lapisan ketiga mengembalikan *output* akhir. Model kemudian dilengkapi dengan dua lapisan *dense*, satu dengan 128 unit dan aktivasi ReLU, dan yang terakhir dengan unit sebanyak jumlah kelas dan aktivasi *softmax* untuk klasifikasi. Model dilatih menggunakan *optimizer* Adam dan metrik *accuracy*. Setelah model

dibangun, selanjutnya dilakukan evaluasi menggunakan *confusion matrikx.* Dalam *confusion matrix* terdapat empat metrik evaluasi performa yang digunakan sebagai berikut [23]:

1. Akurasi
   Akurasi merupakan total prediksi benar yang dilakukan oleh model dibandingkan dengan seluruh data yang diuji. Perhitungan akurasi dilakukan menggunakan Persamaan (1).
$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$
2. *Precision*
   Precision merupakan proporsi prediksi positif yang benar dari seluruh prediksi positif yang dilakukan oleh model. Rumus perhitungan *precision* disajikan dalam Persamaan (2).
$$Precision = \frac{TP}{TP + FP} \quad (2)$$
3. *Recall*
   Recall merupakan proporsi prediksi positif yang benar dari seluruh data yang benar-benar berkelas positif. Formulasi matematis *recall* tercantum pada Persamaan (3).
$$Recall = \frac{TP}{TP + FN} \quad (3)$$
4. *F1-Score*
   F1-Score merupakan rata-rata harmonik dari nilai *precision* dan *recall*. Nilai *F1-score* dapat dihitung sebagaimana ditunjukkan pada Persamaan (4).
$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (4)$$

Keterangan:
TP (*True Positive*): Data aktual yang berada di kelas positif dan model memprediksi positif.
TN (*True Negative*): Data aktual yang berada di kelas negatif dan model memprediksi negatif.
FP (*False Positive*): Data aktual yang berada di kelas negatif atau netral, namun model memprediksi positif.
FN (*False Negative*): Data aktual yang berada di kelas positif atau netral, namun model memprediksi negatif.

3. Hasil dan Pembahasan

Dalam proses analisis data klasifikasi dilakukan dengan algoritma LSTM. Hasil *scraping data* akan digunakan untuk analisis. Setelah data siap, model BERT dan LSTM dibangun. Proses BERT melibatkan penentuan panjang maksimum kalimat, tokenisasi menggunakan *tokenizer* IndoBERT dari pustaka Transformers dengan model *indobenchmark/indobert-base-p1*, dan menghasilkan *input* ID, *attention mask*, serta *token type* ID. Representasi *embedding* dari lapisan tersembunyi terakhir BERT, yang mencerminkan makna semantik, kemudian digunakan sebagai fitur *input* untuk pemodelan LSTM.

Selanjutnya, dilakukan pembuatan dan penentuan model LSTM. Ketidakseimbangan kelas pada data pelatihan ditangani menggunakan *class weighting* dari *sklearn*. Kemudian, dilakukan eksplorasi *hyperparameter* meliputi jumlah unit LSTM (64, 128, 256), nilai *dropout* (0.3, 0.5), dan jenis *optimizer* (Adam, RMSprop, SGD) melalui *grid search* manual untuk mencari konfigurasi dengan akurasi validasi tertinggi. *Random seed* diatur untuk menjaga konsistensi eksperimen. Model LSTM dibangun secara *sequential* dengan dua lapisan LSTM (aktivasi *tanh*, *dropout*, *recurrent_dropout*), diikuti lapisan *dense 12*8 (aktivasi *ReLU*), dan lapisan *output dense* (aktivasi *softmax* untuk klasifikasi multi-kelas).

Kemudian model dilatih menggunakan data latih dan validasi selama hingga 100 *epoch*. Pemilihan *epoch* ini berdasarkan jika jumlah *epoch* yang digunakan terlalu besar model mudah mengalami *overfitting*, sebaliknya jika *epoch* yang digunakan terlalu kecil kemampuan generalisasi model berkurang [24]. Penggunaan *epoch* 100 juga didukung pada penelitian [25]. Penerapan *epoch* 100 pada penelitian tersebut berhasil menghasilkan akurasi sebesar 80.34%.

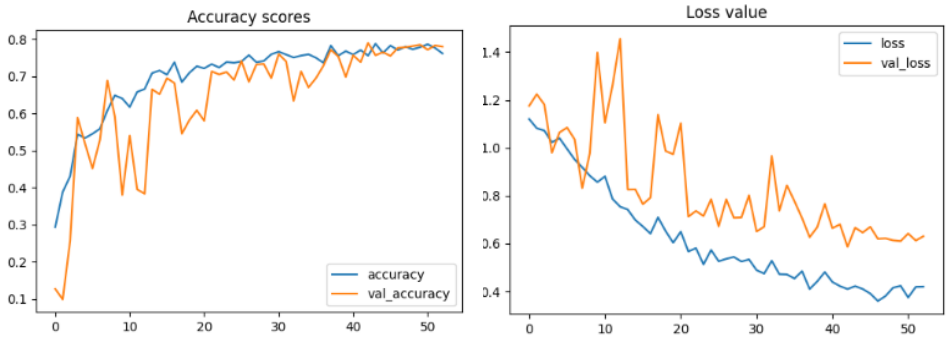
Pada penelitian ini, penentuan model dilakukan pada setiap proses analisis berdasarkan rasio pembagian *dataset* serta penggunaan BERT untuk *word embedding* atau tanpa menggunakannya. Oleh karena itu, klasifikasi dilakukan dengan dua jenis yakni LSTM-BERT ataupun LSTM saja. Setiap jenis model tersebut dianalisis berdasarkan tiga rasio pembagian *dataset*. Rasio pembagian *dataset* yang digunakan terdiri dari 70%:30%, dengan 1400 data latih dan 600 data uji; 80%:20%, dengan 1600 data latih dan 400 data uji; dan 90%:10%, dengan 1800 data latih dan 200 data uji. Berikut ini adalah klasifikasi LSTM-BERT dan LSTM.

3.1 Klasifikasi LSTM-BERT

Berikut ini adalah hasil klasifikasi dengan menggunakan LSTM-BERT dengan 3 macam pembagian *dataset*.

a. Klasifikasi LSTM-BERT *Dataset* 70%:30%

Klasifikasi LSTM-BERT rasio pembagian *dataset* 70%:30%, model terbaiknya memiliki hasil akurasi validasi terbaik 0.9133 menggunakan parameter 256 LSTM unit, 0.3 *dropout rate* dan RMSprop *optimizer*. Grafik *accuracy score* dan *loss value* model terbaik dengan parameter tersebut dan jumlah *epoch* 43 dapat dilihat pada Gambar 5.



Gambar 5. *Accuracy Score* dan *Loss Value* LSTM-BERT 70%:30%

Berdasarkan Gambar 5, terlihat bahwa model mampu memberikan performa yang baik selama proses pelatihan hingga mencapai 43 *epoch*. Pada grafik *accuracy*, akurasi data latih (ditunjukkan oleh garis biru) dan akurasi data validasi (garis oranye) sama-sama mengalami peningkatan signifikan pada awal pelatihan, khususnya hingga sekitar *epoch* ke-15. Setelah itu, kedua nilai tersebut cenderung stabil dan konsisten mengalami kenaikan perlahan hingga akhir pelatihan. Akurasi validasi tertinggi yang dicapai adalah sebesar 0.9133, menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data yang tidak dilatih sebelumnya. Sementara itu, pada grafik *loss*, terlihat bahwa nilai *loss* pada data pelatihan terus menurun secara konsisten, menandakan bahwa kesalahan prediksi semakin kecil seiring bertambahnya *epoch*. Meskipun nilai *loss* pada data validasi mengalami fluktuasi, secara keseluruhan garisnya juga menurun hingga akhir pelatihan. Kecenderungan *loss* yang menurun serta akurasi yang meningkat pada kedua data menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan. Dengan menggunakan parameter 256-unit LSTM, *dropout*

rate sebesar 0.3, dan *optimizer* RMSprop, model LSTM-BERT menunjukkan performa terbaik pada *epoch* ke-43.

Hasil evaluasi model LSTM-BERT pembagian dataset 70%:30% terbaik dengan parameter 256-unit LSTM, *dropout rate* sebesar 0.3, dan *optimizer* RMSprop menunjukkan performa yang sangat baik berdasarkan *confusion matrix*. Model ini mencapai akurasi 91.33%, yang berarti 91.33% dari semua prediksi yang dilakukan oleh model adalah benar. *Precision* model tercatat sebesar 89.55%, menunjukkan bahwa 89.55% dari semua prediksi positif adalah tepat. Model ini juga memiliki *recall* sebesar 91.33%, menandakan kemampuannya dalam mengidentifikasi 91.33% dari semua kasus positif. *F1-Score*, yang merupakan rata-rata harmonis dari presisi dan *recall*, tercatat sebesar 89.18%, menunjukkan keseimbangan yang kuat antara ketepatan prediksi dan kemampuan identifikasi kasus positif.

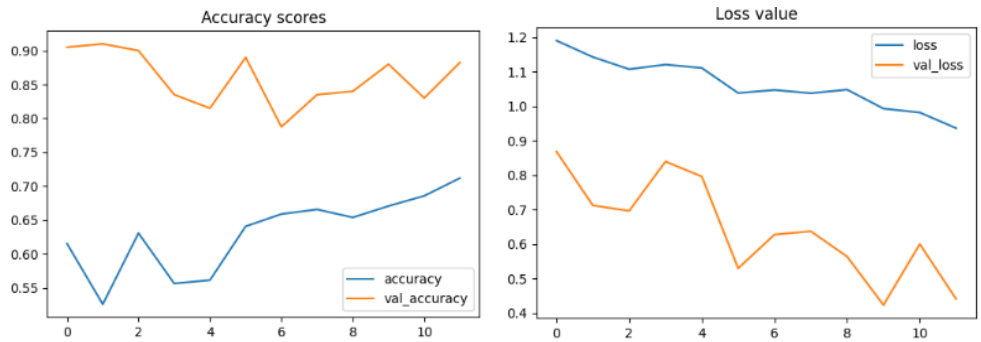
Tabel 1. Performa Per Kelas LSTM-BERT 70%:30%

	Precision	*Recall*	*F1-score*	*Support*
Negatif	0.9299	0.9855	0.9569	552
Netral	0.0000	0.0000	0.0000	12
Positif	0.6667	0.1111	0.1905	36
Accuracy			0.9133	600
Macro avg	0.5322	0.3655	0.3825	600
Weighted avg	0.8955	0.9133	0.8918	600

Model LSTM-BERT dengan rasio dataset 70%:30% pada Tabel 1 menunjukkan performa sangat baik pada kelas Negatif (*F1-score* 0.9569 dari 552 data), namun gagal mengenali kelas Netral (*F1-score* 0.0000 dari 12 data) dan kurang optimal pada kelas Positif (*F1-score* 0.1905 dari 36 data), dengan *recall* yang sangat rendah meskipun *precision* cukup tinggi. Akurasi keseluruhan sebesar 0.9133 dipengaruhi dominasi data Negatif. Nilai *macro average* yang rendah mencerminkan ketimpangan performa antar kelas, sementara *weighted average* yang lebih tinggi menunjukkan pengaruh besar kelas mayoritas. Model masih perlu ditingkatkan untuk menangani kelas minoritas.

b. Klasifikasi LSTM-BERT *Dataset* 80%:20%

Klasifikasi LSTM-BERT rasio pembagian *dataset* 80%:20%, model terbaiknya memiliki hasil akurasi validasi terbaik 0.9100 menggunakan parameter 128 LSTM unit, 0.5 *dropout rate* dan RMSprop *optimizer*. Grafik *accuracy score* dan *loss value* model terbaik dengan parameter tersebut dan jumlah *epoch* 2 dapat dilihat pada **Error! Reference source not found.**6.



Gambar 6. Grafik *Accuracy Score* dan *Loss Value* LSTM-BERT 80%:20%

Seperti yang ditunjukkan pada Gambar 6, grafik *accuracy scores* dan *loss value* menggambarkan peningkatan akurasi secara bertahap dan penurunan nilai *loss*, baik

pada data latih maupun validasi. Grafik *accuracy score* menunjukkan bahwa model secara konsisten meningkatkan performa klasifikasinya pada *epoch* awal, khususnya pada *epoch* ke-2 yang menjadi titik terbaik model. Sementara itu, grafik *loss value* menunjukkan tren penurunan yang cukup stabil pada *val_loss*, mengindikasikan bahwa model mengalami *generalization* yang baik terhadap data validasi dan tidak menunjukkan gejala *overfitting* yang signifikan. Dengan demikian, konfigurasi parameter tersebut dapat dianggap optimal untuk klasifikasi sentimen menggunakan pendekatan LSTM-BERT dalam analisis rasio pembagian *dataset* 80%:20%.

Hasil evaluasi model LSTM-BERT pembagian *dataset* 80%:20% terbaik dengan akurasi 91%, yang berarti 91% dari semua prediksi yang dilakukan oleh model adalah benar. *Precision* model tercatat sebesar 83.77%, menunjukkan bahwa 83.77% dari semua prediksi positif adalah tepat. Model ini juga memiliki *recall* sebesar 91%, menandakan kemampuannya dalam mengidentifikasi 91% dari semua kasus positif. *F1-Score* tercatat sebesar 86.98%, menunjukkan keseimbangan yang kuat antara ketepatan prediksi dan kemampuan identifikasi kasus positif.

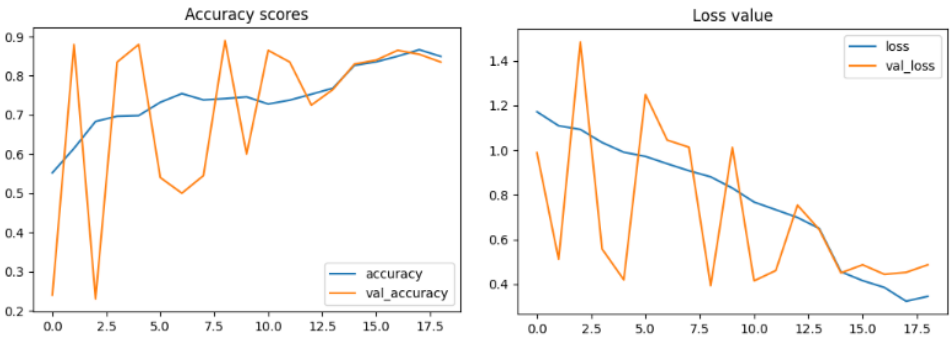
Tabel 2. Performa Per Kelas LSTM-BERT 80%:20%

	Precision	*Recall*	*F1-Score*	*Support*
Negatif	0.9121	1.000	0.9540	363
Netral	0.5000	0.1250	0.2000	8
Positif	0.0000	0.0000	0.0000	29
Accuracy			0.9100	400
Macro avg	0.4707	0.3750	0.3750	400
Weighted avg	0.8377	0.9100	0.8698	400

Model LSTM-BERT dengan rasio dataset 80%:20% pada Tabel 2 menunjukkan performa sangat baik pada kelas Negatif (*F1-score* 0.9540 dari 363 data), namun masih lemah pada kelas Netral (*F1-score* 0.2000 dari 8 data) dan gagal total pada kelas Positif (*F1-score* 0.0000 dari 29 data). Akurasi keseluruhan tetap tinggi (0.9100) karena dominasi data Negatif. Nilai *macro average* yang rendah tetap mencerminkan ketidakseimbangan performa antar kelas, sedangkan *weighted average* yang tinggi menunjukkan bobot besar dari kelas mayoritas (negatif). Secara keseluruhan, model masih perlu ditingkatkan untuk mengenali kelas minoritas (netral dan positif).

c. Klasifikasi LSTM-BERT *Dataset* 90%:10%

Klasifikasi LSTM-BERT rasio pembagian *dataset* 90%:10%, model terbaiknya memiliki hasil akurasi validasi terbaik 0.8900 menggunakan parameter 128 LSTM unit , 0.3 *dropout rate* dan *RMSprop optimizer*. Grafik *accuracy score* dan *loss value* model terbaik dengan parameter tersebut dan jumlah *epoch* 9 dapat dilihat pada ***Error! Reference source not found.***7.



Gambar 7. *Accuracy Score* dan *Loss Value* LSTM-BERT 90%:10%

Berdasarkan Gambar 7, klasifikasi menggunakan model LSTM-BERT dengan rasio pembagian *dataset* 90%:10% menunjukkan performa yang cukup baik. Model terbaik pada skenario ini berhasil mencapai akurasi validasi sebesar 0.8900 dengan menggunakan parameter 128-unit LSTM, *dropout rate* sebesar 0.3, dan *optimizer* RMSprop. Grafik *accuracy score* dan *loss value* menunjukkan bahwa seiring dengan bertambahnya jumlah *epoch*, model mengalami peningkatan akurasi secara konsisten baik pada data pelatihan maupun validasi. Hal ini menandakan bahwa model belajar dengan baik dari data pelatihan. Meskipun grafik *val_accuracy* terlihat fluktuatif di awal, performanya menjadi lebih stabil dan meningkat seiring waktu, hingga akhirnya menyamai atau mendekati *train accuracy* pada *epoch* ke-9. Sementara itu, grafik *loss value* juga menunjukkan penurunan yang signifikan pada kedua kurva (*loss* dan *val_loss*), yang menandakan bahwa model semakin baik dalam meminimalkan kesalahan prediksi.

Hasil evaluasi model LSTM-BERT pembagian *dataset* 90%:10% terbaik dengan akurasi 89%, yang berarti 89% dari semua prediksi yang dilakukan oleh model adalah benar. *Precision* model tercatat sebesar 85.21%, menunjukkan bahwa 85.21% dari semua prediksi positif adalah tepat. Model ini juga memiliki *recall* sebesar 89%, menandakan kemampuannya dalam mengidentifikasi 89% dari semua kasus positif. *F1-Score* tercatat sebesar 86.47%, menunjukkan keseimbangan yang kuat antara ketepatan prediksi dan kemampuan identifikasi kasus positif.

Tabel 3. Performa Per Kelas LSTM-BERT 90%:10%

	Precision	*Recall*	*F1-Score*	*Support*
Negatif	0.9053	0.9829	0.9425	175
Netral	0.0000	0.0000	0.0000	5
Positif	0.6000	0.3000	0.4000	20
Accuracy			0.8900	200
Macro avg	0.5018	0.4276	0.4475	200
Weighted avg	0.8521	0.8900	0.8647	200

Berdasarkan Tabel 3, model LSTM-BERT dengan rasio *dataset* 90%:10% memiliki performa sangat baik pada kelas Negatif dengan *F1-score* 0.9425. Namun, performa pada kelas Netral sangat buruk (*F1-score* 0.0000) akibat jumlah data yang minim. Kelas Positif menunjukkan hasil kurang optimal dengan *F1-score* 0.4000. Meskipun akurasi keseluruhan mencapai 89%, nilai *macro average* yang rendah menandakan ketidakseimbangan performa antar kelas, sedangkan *weighted average* yang cukup tinggi menunjukkan dominasi akurasi pada kelas mayoritas (negatif).

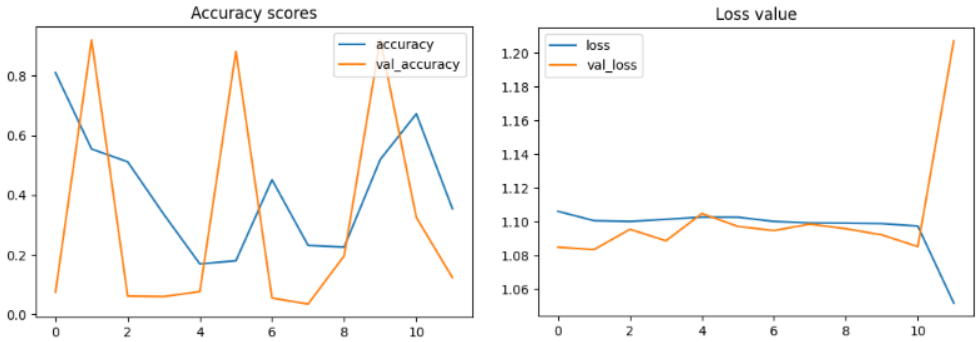
3.2 Klasifikasi LSTM

Berikut ini adalah klasifikasi dengan menggunakan LSTM saja dengan 3 macam pembagian *dataset*.

a. Hasil Klasifikasi LSTM *Dataset* 70%:30%

Klasifikasi LSTM rasio pembagian *dataset* 70%:30%, model terbaiknya memiliki hasil akurasi validasi terbaik 0.9200 menggunakan parameter 64 LSTM unit, 0.3 *dropout rate* dan Adam *optimizer*. Grafik *accuracy score* dan *loss value* model terbaik dengan parameter tersebut dan jumlah *epoch* 2 dapat dilihat pada Gambar 88.

Berdasarkan Gambar 88, grafik menunjukkan hasil pelatihan model LSTM dengan rasio pembagian data 70%:30%. Grafik *accuracy score* memperlihatkan bahwa nilai *accuracy* dan *validation accuracy* mengalami fluktuasi yang cukup signifikan, namun secara umum menunjukkan peningkatan pada awal pelatihan dan kemudian stabil di akhir.



Gambar 8. *Accuracy Score* dan *Loss Value* LSTM 70%:30%

Model mencapai akurasi validasi terbaik sebesar 0.9200 pada parameter terbaik yaitu 64 LSTM unit, *dropout rate* 0.3, dan *optimizer Adam* dengan jumlah *epoch*-2. Sementara itu, grafik *loss value* menunjukkan tren penurunan baik pada *loss* maupun *validation loss*, yang mengindikasikan bahwa model mengalami proses pembelajaran yang baik dengan penurunan kesalahan selama pelatihan dan validasi.

Hasil evaluasi model LSTM pembagian *dataset* 70%:30% terbaik dengan akurasi 92%, yang berarti 92% dari semua prediksi yang dilakukan oleh model adalah benar. Presisi model tercatat sebesar 84.64%, menunjukkan bahwa 84.64% dari semua prediksi positif adalah tepat. Model ini juga memiliki *recall* sebesar 92%, menandakan kemampuannya dalam mengidentifikasi 92% dari semua kasus positif. *F1-Score* sebesar 88.17% menunjukkan keseimbangan yang kuat antara ketepatan prediksi dan kemampuan identifikasi kasus positif.

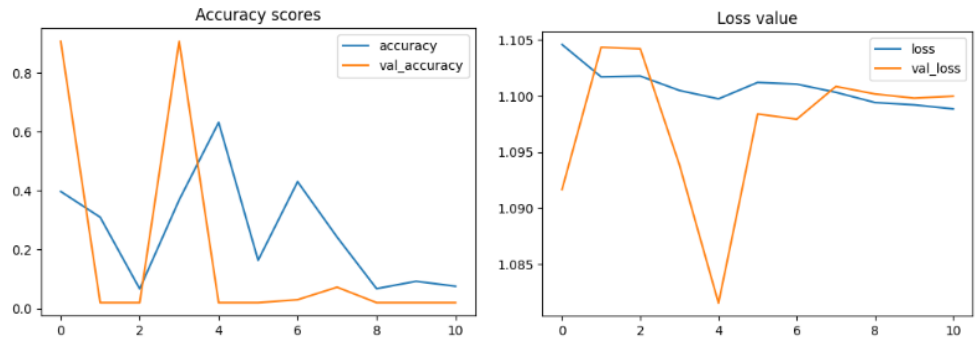
Tabel 4. Performa Per Kelas LSTM 70%:30%

	Precision	*Recall*	*F1-Score*	*Support*
Negatif	0.9200	1.0000	0.9583	552
Netral	0.0000	0.0000	0.0000	12
Positif	0.0000	0.0000	0.0000	36
Accuracy			0.9200	600
Macro avg	0.3067	0.3333	0.3194	600
Weighted avg	0.8464	0.9200	0.8817	600

Berdasarkan Tabel 4, model LSTM dengan rasio data 70%:30% menunjukkan performa sangat baik pada kelas Negatif, dengan *precision* 0.9200, *recall* sempurna 1.0000, dan *F1-score* 0.9583 dari 552 data. Namun, model gagal mengenali kelas Netral dan Positif, ditunjukkan oleh nilai *precision*, *recall*, dan *F1-score* sebesar 0.0000 pada kedua kelas tersebut. Akurasi keseluruhan tetap tinggi di angka 0.9200 karena dominasi data kelas Negatif. Nilai *macro average* yang rendah (*precision* 0.3067, *recall* 0.3333, *F1-score* 0.3194) mencerminkan ketidakseimbangan performa antar kelas, sementara *weighted average* yang lebih tinggi (*precision* 0.8464, *recall* 0.9200, *F1-score* 0.8817) menunjukkan model lebih baik dalam menangani kelas mayoritas (negatif).

b. Hasil Klasifikasi LSTM *Dataset* 80%:20%

Klasifikasi LSTM rasio pembagian *dataset* 80%:20%, model terbaiknya memiliki hasil akurasi validasi terbaik 0.9075 menggunakan parameter 64 LSTM unit, 0.3 *dropout rate* dan Adam *optimizer*. Grafik *accuracy score* dan *loss value* model terbaik dengan parameter tersebut dan jumlah *epoch* 1 atau 3 dapat dilihat pada Gambar 9.



Gambar 9. *Accuracy Score* dan *Loss Value* LSTM 80%:20%

Berdasarkan Gambar 9, model terbaik menggunakan 64-unit LSTM, *dropout rate* 0.3, dan Adam *optimizer*. Dari grafik, terlihat bahwa *validation accuracy* (garis oranye) sempat fluktuatif namun mencapai nilai tertinggi 0.9075, sedangkan *training accuracy* (biru) menunjukkan tren menurun. Pada grafik *loss*, *validation loss* awalnya tinggi namun kemudian menurun drastis di *epoch* ke-2, lalu stabil. Ini menunjukkan model mengalami proses pembelajaran awal yang kurang stabil namun mulai terkonsolidasi setelah beberapa *epoch*, meskipun dengan hanya 1 *epoch* terbaik dihasilkan model dengan performa optimal.

Hasil evaluasi model LSTM pembagian *dataset* 80%:20% terbaik dengan akurasi 90%, yang berarti 90% dari semua prediksi yang dilakukan oleh model adalah benar. *Precision* model tercatat sebesar 82.36%, menunjukkan bahwa 82.36% dari semua prediksi positif adalah tepat. Model ini juga memiliki *recall* sebesar 90.75%, menandakan kemampuannya dalam mengidentifikasi 90.75% dari semua kasus positif. *F1-Score* tercatat sebesar 86.35%, menunjukkan keseimbangan yang kuat antara ketepatan prediksi dan kemampuan identifikasi kasus positif.

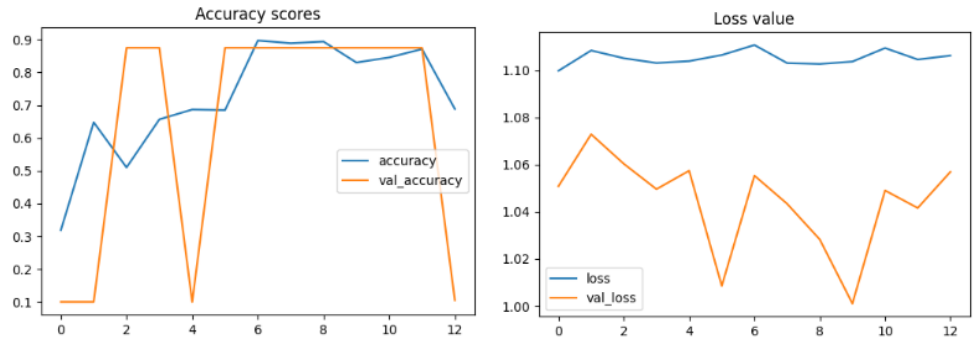
Tabel 5. Performa Per Kelas LSTM 80%:20%

	Precision	*Recall*	*F1-Score*	*Support*
Negatif	0.9075	1.0000	0.9515	363
Netral	0.0000	0.0000	0.0000	8
Positif	0.0000	0.0000	0.0000	29
Accuracy			0.9075	400
Macro avg	0.3025	0.3333	0.3172	400
Weighted avg	0.8236	0.9075	0.8635	400

Tabel 5 menunjukkan performa model LSTM pada data uji dengan rasio pelatihan 80%:20%. Model menunjukkan kinerja sangat baik dalam mengklasifikasikan kelas Negatif, dengan *precision* sebesar 0.9075, *recall* sempurna (1.0000), dan *F1-score* sebesar 0.9515, dari total 363 data dalam kelas ini. Namun, model gagal sepenuhnya dalam mengenali kelas Netral dan Positif, yang ditunjukkan dengan nilai *precision*, *recall*, dan *F1-score* sebesar 0.0000 pada kedua kelas tersebut. Jumlah data dalam kelas Netral dan Positif masing-masing hanya 8 dan 29, yang kemungkinan menyebabkan model kurang belajar dari kelas-kelas ini (masalah ketidakseimbangan data). Meskipun demikian, model tetap mencatat akurasi keseluruhan yang tinggi, yaitu 90.75%. Namun, nilai *macro average* menunjukkan performa rendah (*F1-score* 0.3172), sedangkan *weighted average* menunjukkan nilai lebih tinggi (*F1-score* 0.8635), mencerminkan dominasi kelas Negatif dalam hasil klasifikasi.

c. Hasil Klasifikasi LSTM *Dataset* 90%:10%

Klasifikasi LSTM rasio pembagian *dataset* 90%:10%, model terbaiknya memiliki hasil akurasi validasi 0.8750 menggunakan parameter 64 LSTM unit, 0.3 *dropout rate* dan RMSprop *optimizer*. Grafik *accuracy score* dan *loss value* model terbaik dengan parameter tersebut dan jumlah *epoch* 3 dapat dilihat pada Gambar 60.



Gambar 60. *Accuracy Score* dan *Loss Value* LSTM 90%:10%

Grafik sebelah kiri (*Accuracy scores*) pada Gambar 60, terlihat bahwa akurasi pelatihan (*train accuracy*) relatif tinggi dan stabil, sementara akurasi validasi (*val accuracy*) bervariasi namun mencapai nilai tertinggi sebesar 0.875. Grafik kanan (*Loss value*) menunjukkan bahwa nilai *loss* pada data pelatihan cenderung menurun, menandakan proses pembelajaran berjalan baik, sedangkan *loss* validasi berfluktuasi, yang mengindikasikan kemungkinan *overfitting* ringan setelah beberapa *epoch*. Model ini menunjukkan performa terbaiknya pada *epoch* ke-3.

Hasil evaluasi model LSTM pembagian *dataset* 90%:10% terbaik dengan akurasi 87.50%, yang berarti 87.50% dari semua prediksi yang dilakukan oleh model adalah benar. *Precision* model tercatat sebesar 76.56%, menunjukkan bahwa 76.56% dari semua prediksi positif adalah tepat. Model ini juga memiliki *recall* sebesar 87.5%, menandakan kemampuannya dalam mengidentifikasi 87.5% dari semua kasus positif. *F1-Score*, yang merupakan rata-rata harmonis dari presisi dan *recall*, tercatat sebesar 81.67%, menunjukkan keseimbangan yang kuat antara ketepatan prediksi dan kemampuan identifikasi kasus positif.

Tabel 6. Performa Per Kelas LSTM 90%:10%

	Precision	*Recall*	*F1-Score*	*Support*
Negatif	0.8750	1.0000	0.9333	175
Netral	0.0000	0.0000	0.0000	5
Positif	0.0000	0.0000	0.0000	20
Accuracy			0.8750	200
Macro avg	0.3025	0.3333	0.3111	200
Weighted avg	0.7656	0.8750	0.8167	200

Tabel 6 menampilkan performa model LSTM pada data uji dengan rasio 90%:10%. Model mampu mengklasifikasikan kelas Negatif dengan sangat baik, ditunjukkan oleh *precision* sebesar 0.8750, *recall* sempurna (1.0000), dan *F1-score* 0.9333 dari total 175 data. Namun, performa model terhadap kelas Netral dan Positif sangat rendah, dengan *precision*, *recall*, dan *F1-score* sebesar 0.0000, yang berarti model sama sekali tidak mengenali data dalam kedua kelas tersebut (masalah ketidakseimbangan data). Akurasi keseluruhan mencapai 87.50%, namun ini terutama karena dominasi kelas Negatif. Nilai *macro average* yang rendah (*F1-score* 0.3111) menunjukkan performa buruk terhadap kelas minoritas, sementara *weighted average*

yang lebih tinggi (*F1-score* 0.8167) mencerminkan kontribusi besar dari kelas mayoritas.

Berdasarkan hasil analisis model LSTM, baik yang menggunakan BERT maupun yang tidak, dengan parameter dan rasio pembagian *dataset* yang berbeda, hasil keseluruhannya dirangkum pada Tabel 7. Tabel tersebut menyajikan ringkasan performa model berdasarkan variasi parameter dan pembagian data.

Tabel 7. Hasil Evaluasi Keseluruhan

Split Data	LSTM Unit	*Dropout Rate*	*Optimizer*	Akurasi	*Precision*	*Recall*	*F1-Score*
LSTM-BERT							
70%:30%	256	0.3	RMSprop	91.33%	89.55%	91.33%	89.18%
80%:20%	128	0.5	RMSprop	91.00%	83.77%	91.00%	86.98%
90%:10%	128	0.3	RMSprop	89.00%	85.21%	89.00%	86.47%
LSTM							
70%:30%	64	0.3	Adam	92.00%	84.64%	92.00%	88.17%
80%:20%	64	0.3	Adam	90.75%	82.36%	90.75%	86.35%
90%:10%	64	0.3	RMSprop	87.50%	76.56%	87.50%	81.67%

Tabel 7 menyajikan hasil evaluasi yang membandingkan performa antara model LSTM-BERT dan LSTM pada berbagai rasio pembagian data (70%:30%, 80%:20%, dan 90%:10%) dengan konfigurasi parameter yang berbeda. Secara umum, model LSTM-BERT menunjukkan performa yang lebih stabil dan kompetitif di semua rasio. Pada rasio 70%:30%, model LSTM-BERT dengan 256 unit LSTM, *dropout rate* 0.3, dan *optimizer* RMSprop mencatat akurasi sebesar 91.33%, *precision* 89.55%, dan *F1-score* 89.18%, menunjukkan bahwa model ini sangat baik dalam mengenali data dengan keseimbangan antara *recall* dan *precision*. Meski terjadi sedikit penurunan pada rasio 80%:20% dan 90%:10%, LSTM-BERT tetap menjaga performa tinggi dengan *F1-score* masing-masing 86.98% dan 86.47%.

Sementara itu, model LSTM standar menunjukkan performa terbaiknya pada rasio 70%:30% dengan konfigurasi 64-unit LSTM, *dropout* 0.3, dan *optimizer Adam*. Pada konfigurasi ini, LSTM berhasil mencapai akurasi tertinggi sebesar 92.00%, dengan *precision* 84.64% dan *F1-score* 88.17%, bahkan mengungguli LSTM-BERT dalam akurasi. Namun, performa LSTM cenderung menurun seiring dengan berkurangnya porsi data latih. Pada rasio 80%:20% dan terutama 90%:10%, *precision* menurun signifikan hingga hanya 76.56%, yang berdampak pada turunnya *F1-score* menjadi 81.67%. Hal ini mengindikasikan bahwa LSTM cenderung lebih sensitif terhadap jumlah data pelatihan dibandingkan LSTM-BERT.

Dari sisi parameter, terlihat bahwa jumlah unit LSTM dan *dropout rate* memiliki pengaruh langsung terhadap hasil akhir. LSTM-BERT dengan unit lebih banyak (256) pada rasio 70%:30% menunjukkan performa lebih baik dibanding saat menggunakan 128 unit. Selain itu, *dropout rate* yang lebih rendah (0.3) memberikan hasil yang lebih stabil dibandingkan 0.5. Penggunaan *optimizer* juga berperan penting, di mana LSTM menunjukkan hasil terbaik dengan Adam, sedangkan LSTM-BERT konsisten bekerja optimal dengan *RMSprop*.

Selain itu, pada *dataset* yang digunakan terdapat ketidakseimbangan jumlah label yaitu negatif sebanyak 1834, positif 127, dan netral 39. Untuk mengatasi hal tersebut, penelitian ini mengimpor “class_weight” dari “sklearn.utils”. Fungsi “compute_class_weight” kemudian dipanggil dengan parameter “class_weight='balanced'”, “classes=np.unique(y_train_le)”, dan “y=y_train_le”, sehingga menghasilkan *array* bobot di mana kelas minoritas diberi bobot lebih besar, sedangkan kelas mayoritas bobot lebih kecil. Masing-masing label dengan hasil evaluasinya telah dijabarkan sebelumnya pada Tabel 1 hingga Tabel 6.

Secara keseluruhan, LSTM lebih unggul pada data latih yang besar, namun LSTM-BERT lebih konsisten dan tangguh dalam berbagai skenario, terutama ketika data latih lebih

sedikit. Hasil ini memperlihatkan bahwa pemilihan model dan konfigurasi parameter yang tepat sangat menentukan performa akhir sistem klasifikasi teks berbasis LSTM. Kombinasi LSTM dan BERT menunjukkan hasil yang lebih baik, tidak hanya karena konfigurasi parameter yang digunakan, tetapi dapat dimungkinkan karena kemampuan BERT dalam memahami semantik bahasa secara kontekstual yang memperkuat performa model [26].

4. Kesimpulan

Berdasarkan hasil analisis dan perhitungan, dapat disimpulkan hasil pengelompokan data menggunakan metode Lexicon Inset dan visualisasi *word could* menunjukkan bahwa ulasan aplikasi M-Paspor di Google Play Store cenderung negatif dengan jumlah kata negatif 1834 positif 127, dan netral 39. Hal tersebut mencerminkan ketidakpuasan pengguna. Visualisasi *word cloud* menyoroti kata-kata yang paling sering muncul seperti "aplikasi," "daftar," "bayar," dan "paspor," serta masalah yang dilaporkan seperti "*error*," "sulit," dan "email". Penggunaan metode BERT *embedding* dan LSTM terbukti lebih konsisten dalam melakukan klasifikasi dalam berbagai skenario terutama ketika jumlah data latih lebih sedikit dengan akurasi klasifikasi tertinggi sebesar 91,33% pada pembagian data 70%:30%. Meskipun nilai akurasi tertinggi terdapat LSTM yakni 92%, namun performa LSTM cenderung menurun seiring dengan berkurangnya porsi data latih.

Peneliti selanjutnya disarankan untuk membandingkan performa BERT dengan model transformer lain seperti RoBERTa atau DistilBERT untuk meningkatkan analisis sentimen. Hal ini karena apabila dilakukan perbandingan, pembagian data latih lebih banyak dibandingkan dengan data uji juga dapat memungkinkan terjadinya *overfitting*.

Daftar Referensi

[1] Hasmawaty, Y. T. Utami, and D. Antoni, “Building green smart city capabilities in South Sumatra, Indonesia,” *Sustain.*, vol. 14, no. 13, pp. 1–16, 2022, doi: 10.3390/su14137695.

[2] B. S. Helpiastuti, I. Syaifana, and H. Rohman, “Kualitas pelayanan m-paspor di Kantor Imigrasi Kelas I TPI Jember,” *J. Ilm. Manaj. Publik dan Kebijak. Sos.*, vol. 7, no. 1, pp. 15–30, 2023, [Online]. Available: https://ejournal.unitomo.ac.id/index.php/negara/article/view/5464.

[3] J. Simmich, M. H. Ross, N. E. Andrews, A. Vaezipour, and T. G. Russell, “Content and quality of mobile apps for the monitoring of musculoskeletal or neuropathic pain in Australia: Systematic evaluation,” *JMIR mHealth uHealth*, vol. 11, no. 1, pp. 1–11, 2023, doi: 10.2196/46881.

[4] Y. H. Tsai, C. C. Lin, and M. H. Lee, “Analysis of application data mining to capture consumer review data on booking websites,” *Mobile Inf. Syst.*, vol. 2022, 2022, doi: 10.1155/2022/3062953.

[5] A. Abayomi-Alli, O. Abayomi-Alli, S. Misra, and L. Fernandez-Sanz, “Study of the Yahoo-Yahoo hash-tag tweets using sentiment analysis and opinion mining algorithms,” *Information*, vol. 13, no. 3, 2022, doi: 10.3390/info13030152.

[6] M. H. Widianto and Y. Cornelius, “Sentiment analysis towards cryptocurrency and NFT in Bahasa Indonesia for Twitter large amount data using BERT,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 1, pp. 303–309, 2023.

[7] A. S. Raihan and I. Ahmed, “A Bi-LSTM autoencoder framework for anomaly detection: A case study of a wind power dataset,” in *Proc. IEEE 19th Int. Conf. Autom. Sci. Eng.*, 2023, doi: 10.1109/CASE56687.2023.10260331.

[8] S. Mohammadi and M. Chapon, “Investigating the performance of fine-tuned text classification models based on BERT,” in *Proc. IEEE 22nd Int. Conf. High Perform.*

Comput. Commun.; IEEE 18th Int. Conf. Smart City; IEEE 6th Int. Conf. Data Sci. Syst. (HPCC-SmartCity-DSS), 2020, pp. 1252–1257, doi: 10.1109/HPCC-SmartCity-DSS50907.2020.00162.

[9] K. Jain and R. Jindal, "NLP-enabled recommendation of gashtags for COVID-based tweets using hybrid BERT-LSTM model," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, 2024, doi: 10.1145/3640812.

[10] R. Firdaus, I. Asror, and A. Herdiani, "Lexicon-based sentiment analysis of Indonesian language student feedback evaluation," *Indones. J. Comput.*, vol. 6, no. 1, pp. 1–12, 2021, doi: 10.34818/indojc.2021.6.1.408.

[11] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, "Fake news classification using transformer-based enhanced LSTM and BERT," *Int. J. Cogn. Comput. Eng.*, vol. 3, pp. 98–105, Mar. 2022, doi: 10.1016/j.ijcce.2022.03.003.

[12] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment analysis using Word2vec and long short-term memory (LSTM) for Indonesian hotel reviews," *Procedia Comput. Sci.*, vol. 179, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.

[13] A. Alsharef, K. Aggarwal, Sonia, D. Koundal, H. Alyami, and D. Ameyed, "An automated toxicity classification on social media using LSTM and word embedding," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/8467349.

[14] M. S. Divate, "Sentiment analysis of Marathi news using LSTM," *Int. J. Inf. Technol.*, vol. 13, no. 5, pp. 2069–2074, 2021, doi: 10.1007/s41870-021-00702-1.

[15] V. Yadav, P. Verma, and V. Katiyar, "Long short-term memory (LSTM) model for sentiment analysis in social data for e-commerce product reviews in Hindi language," *Int. J. Inf. Technol.*, vol. 15, no. 2, pp. 759–772, 2023, doi: 10.1007/s41870-022-01010-y.

[16] S. S. Khan, P. K. Mondal, S. Shaqib, N. Ahmed, N. N. I. Prova, and A. Sattar, "Performance analysis of LSTM and Bi-LSTM model with different optimizers in Bangla sentiment analysis," in *Proc. 15th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2024, doi: 10.1109/ICCCNT61001.2024.10726142.

[17] P. S. Shilpa, S. Jacob, S. Rissa, and P. Vinod, "Context-aware gender and age recognition from smartphone sensors," in *Proc. Int. Conf. Comput. Commun. Secur. Intell. Syst. (IC3SIS)*, 2021, doi: 10.1109/IC3SIS54991.2022.9885610.

[18] A. K. Chakraborty, S. Das, and A. K. Kolya, "Sentiment analysis of COVID-19 tweets using evolutionary classification-based LSTM model," in *Proc. Res. Appl. Artif. Intell.*, Aug. 2020, pp. 75–86, doi: 10.1007/978-981-16-1543-6_7.

[19] Y. Abdelwahab, M. Kholief, and A. A. H. Sedky, "Justifying Arabic text sentiment analysis using explainable AI (XAI): LASIK surgeries case study," *Information*, vol. 13, no. 11, 2022, doi: 10.3390/info13110536.

[20] M. Mujahid et al., "Data oversampling and imbalanced datasets: An investigation of performance for machine learning and feature engineering," *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00943-4.

[21] F. A. Pratama and A. Romadhony, "Identifikasi komentar toksik dengan BERT," *e-Proceeding Eng.*, vol. 7, no. 2, pp. 7941–7949, 2020.

[22] A. Rolangon, A. Weku, and G. A. Sandag, "Perbandingan algoritma LSTM untuk analisis sentimen pengguna Twitter terhadap layanan rumah sakit saat pandemi COVID-19," *TeIKa*, vol. 13, no. 1, pp. 31–40, 2023, doi: 10.36342/teika.v13i01.3063.

[23] K. Wabang, O. D. Nurhayati, and Farikhin, "Application of the Naïve Bayes classifier algorithm to classify community complaints," *RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 1, no. 1, pp. 19–25, 2022.

[24] R. A. Pramunendar, D. P. Prabowo, and R. A. Megantara, "Metode recurrent neural network (RNN) dengan arsitektur LSTM untuk analisis sentimen opini publik terkait

vaksin COVID-19," *J. Inform. Upgris*, vol. 8, no. 1, pp. 44–48, 2022.
[25] C. A. Maharani, B. Warsito, and R. Santoso, "Analisis sentimen vaksin COVID-19 pada Twitter menggunakan recurrent neural network (RNN) dengan algoritma long short-term memory (LSTM)," *J. Gaussian*, vol. 12, no. 3, pp. 403–413, 2024, doi: 10.14710/j.gauss.12.3.403-413.
[26] R. Belaroussi, S. C. Noufe, F. Dupin, and P. O. Vandanjon, "Polarity of Yelp reviews: A BERT–LSTM comparative study," *Big Data Cogn. Comput.*, vol. 9, no. 5, pp. 1–25, 2025, doi: 10.3390/bdcc9050140.