 ISSN:2302-8149
e-ISSN:2540-9719

Otomatisasi Ulasan Pengguna: Deteksi Keluhan *Actionable* pada Gojek di *Play Store* menggunakan Metode LSTM

User Review Automation: Detecting Actionable Complaints on Gojek in the Play Store using the LSTM Method

[1]Indira Nailah Ramadhani, [2]Winda Kurnia Sari*, [3]Ken Ditha Tania
[1,2,3]Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Sriwijaya
[1,2,3]Jl. Srijaya Negara, Kota Palembang, Sumatera Selatan, Indonesia
*e-mail: indiranailah@gmail.com

(***received*:** 16 August 2025, ***revised*:** 12 September 2025, ***accepted*:** 14 September 2025)

Abstrak

Penelitian ini bertujuan untuk mengembangkan pendeteksi keluhan secara otomatis pada ulasan aplikasi Gojek dengan menggunakan *Long Short Term Memory* (LSTM). *Dataset* yang digunakan terdiri dari 225.002 ulasan pengguna yang terdapat pada Google Play Store. Tujuan dari penelitian ini sendiri adalah untuk mempermudahkan tim pelayanan dalam memahami kekurangan dari aplikasi yang dikeluhkan oleh para pengguna. Pendeteksi keluhan dengan otomatis akan memudahkan tim pelayanan untuk memberikan tindakan dalam mengatasi permasalahan yang dialami pengguna. Maka dari itu, data ulasan yang diberikan para pengguna diolah dengan baik menggunakan LSTM untuk menciptakan sistem pendeteksi yang efektif dan efisien. Pengolahan dilakukan dengan menggunakan tiga rasio pembagian data yang berbeda, yaitu 90:10, 80:20, dan 70:30 untuk menjamin bahwa sistem yang dibangun sudah stabil dan efektif. Hasil akurasi dari ketiga rasio pembagian data mencapai angka di atas 90%, sehingga membuktikan bahwa sistem mampu mendeteksi keluhan dengan baik. Untuk mempermudah pemantauan hasil klasifikasi, digunakan *dashboard* yang sudah dibangun sebelumnya untuk memvisualisasikan hasil dari sistem yang dibangun menggunakan LSTM. Sistem ini diharapkan dapat mempermudah perusahaan untuk mendeteksi seluruh keluhan dari pengguna dan mencari solusi untuk meningkatkan pelayanan agar memberikan kenyamanan bagi para pengguna.

Kata kunci: gojek, *long short-term memory*, prediksi keluhan, ulasan pengguna

Abstract

This study aims to develop an automatic complaint detector for Gojek app reviews using Long Short Term Memory (LSTM). The dataset consists of 225,002 user reviews on the Google Play Store. The purpose of this study itself is to facilitate the service team in understanding the shortcomings of the application complained by users. Automatic complaint detection will facilitate the service team to take action to resolve the problems experienced by users. Therefore, the review data provided by users is properly processed using LSTM to create an effective and efficient detection system. Processing is carried out using three different data sharing ratios, namely 90:10, 80:20, and 70:30 to ensure that the system is stable and effective. The accuracy results of the three data sharing ratios reached above 90%, thus proving that the system is able to detect complaints well. A pre-built dashboard is used to visualize the results of the system built using LSTM to facilitate monitoring the classification results. This system is expected to facilitate companies in detecting all user complaints and finding solutions to improve services to provide comfort for users.

Keywords: *gojek, long short-term memory, complaint prediction, user reviews*

1 Pendahuluan

Perkembangan teknologi digital sudah merambat ke seluruh bidang kehidupan. Salah satunya adalah bidang industri transportasi yang menyediakan layanan berbasis aplikasi [1]. Inilah yang menyebabkan banyak perusahaan yang berbondong – bondong untuk menciptakan aplikasi pelayanan

 ISSN:2302-8149
e-ISSN:2540-9719

transportasi. Adapun salah satunya, seperti Gojek yang menyediakan berbagai macam layanan transportasi secara *online* [2]. Selain Gojek sendiri, masih ada banyak aplikasi industri transportasi lainnya, hal tersebut menyebabkan tingginya tingkat persaingan jumlah pengguna di antara setiap aplikasi. Di dalam Google Play Store sendiri terdapat ulasan mengenai seluruh aplikasi di dalamnya. Ulasan – ulasan tersebut berfungsi sebagai pengukuran yang efektif dan efisien dalam mendapatkan informasi mengenai suatu produk tertentu [3]. Umpan balik yang ada di Google Play Store menjadi alat ukur seberapa maksimalnya kinerja dari suatu aplikasi. Ada banyak sekali ulasan pengguna mengenai fitur dari Gojek, seperti fitur pemesanan, pelacakan pesanan dan transaksi pembayaran [4]. Ulasan tersebut tentunya akan terus bertambah setiap harinya. Tercatat ada sebanyak ribuan hingga jutaan ulasan yang diberikan oleh para pengguna melalui Google Play Store [5]. Dalam jumlah data yang besar, tidak memungkinkan apabila tim layanan pelanggan memproses data tersebut secara manual. Sedangkan, untuk meningkatkan kinerja Gojek, tim layanan pelanggan harus mampu mengambil tindakan untuk mengatasi keluhan – keluhan yang diberikan pelanggan melalui ulasan aplikasi. Akan tetapi, tim layanan pelanggan akan kesulitan dalam memilah dan mengolah data dalam jumlah besar.

Dalam hal ini, tim layanan pelanggan membutuhkan sebuah sistem yang dapat dengan otomatis memisahkan ulasan – ulasan yang diberikan oleh para pengguna. Data ulasan yang terlalu besar dan kompleks merupakan permasalahan yang menjadi landasan utama dilakukannya penelitian ini karena membutuhkan waktu yang lama apabila diolah secara manual. Namun, permasalahan tersebut dapat diatasi dengan penerapan *text mining*. Dengan penerapan *text mining*, sistem otomatisasi pendeteksi keluhan melalui ulasan tentu dapat dibuat. *Text mining* mampu memperoleh data dalam jumlah besar dan memproses teks di dalam data tersebut untuk bisa mendapatkan informasi yang dibutuhkan [6]. Di dalam analisis *text mining*, terdapat beberapa proses yang harus dilalui, seperti *crawling, cleaning* dan pra-pemrosesan. Pra-pemrosesan merupakan tahapan pembentukan sebuah teks menjadi informasi dengan kualitas yang lebih baik dan siap diolah ke dalam proses berikutnya [7]. Setelah pra-pemrosesan, data akan diolah menggunakan metode *machine learning* ataupun *deep learning*, tergantung pada jumlah data yang akan diolah. Data dalam jumlah besar dan kompleks, akan lebih efektif apabila diolah menggunakan *deep learning* [8]. Model *deep learning* mampu memproses dan memahami data dalam jumlah besar walaupun tidak diprogram secara eksplisit. *Deep learning* merupakan sebuah teknologi yang tergolong kuat bahkan memiliki potensi besar dalam meningkatkan penelitian di berbagai bidang [9]**.** Salah satu model *deep learning* adalah LSTM, *Long Short-Term Memory* (LSTM) merupakan salah satu jenis metode dalam jaringan saraf tiruan yang dapat dengan mudah memahami pola dari data yang berurutan dan efektif dalam menemukan solusi yang melibatkan urutan waktu [10]**.**

Dalam penelitian ini, data yang akan dianalisis berupa data umpan balik pengguna aplikasi Gojek yang bersifat teks dan memiliki karakteristik sekuensial. Metode LSTM mampu untuk memodelkan dan mempelajari hubungan jangka panjang di antara setiap data yang membutuhkan konteks di dalam pemahamannya [11]. Data yang sudah disaring nantinya akan diklasifikasikan menjadi dua, yaitu Keluhan dan Bukan Keluhan. Penelitian ini bertujuan untuk membangun model deteksi keluhan yang dilaporkan pengguna guna untuk bisa memilah tindakan solusi apa yang harus diprioritaskan. Dengan adanya tindakan pemberian solusi atas ulasan berupa keluhan, tentunya akan menambah kenyamanan pengguna saat menggunakan Gojek. Pengguna yang merasa puas akan kinerja Gojek tentunya akan memberikan umpan balik yang baik pula dan memperngaruhi penilaian Gojek melalui Google Play Store. Maka dari itu, sistem tersebut diharapkan dapat membantu memprediksi keluhan yang paling banyak dilaporkan oleh para pengguna. Dengan adanya sistem tersebut, pihak tim pelayanan dapat bekerja dengan lebih efektif dan efisien dalam menganalisis data ulasan dalam jumlah banyak dan kompleks. Sistem otomatis membantu perusahaan mengenali ulasan yang berupa keluhan dan bukan keluhan dalam waktu yang singkat.

2 Tinjauan Literatur

Proses klasifikasi dengan LSTM juga cukup sering digunakan pada penelitian yang sudah ada sebelumnya. Adapun pada penelitian [12] yang mengolah data aplikasi Maxim pada Google Play Store menghasilkan nilai akurasi sebesar 84% yang menggunakan metode *naive bayes.* Menurut [13] metode LSTM mampu menghasilkan klasifikasi teks berita sebanyak 4000 data yang memiliki

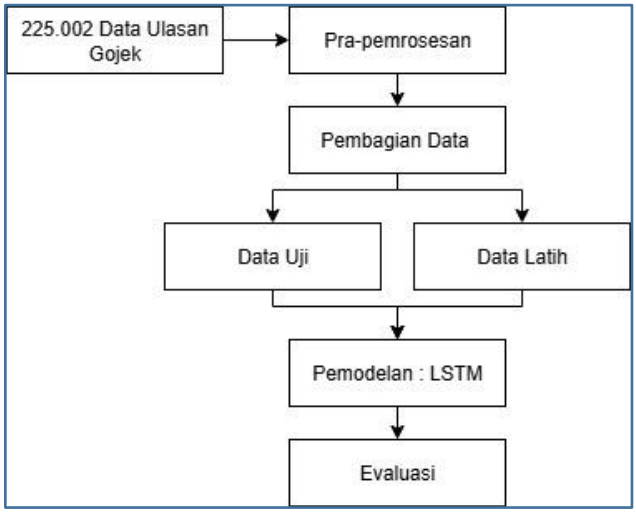
 ISSN:2302-8149
e-ISSN:2540-9719

kategori 1000 di setiap *dataset* dengan efektif dan menghasilkan *accuracy* sebesar 88.75%, *precision* 88.97%, *recall* 88.75%, dan *F1-Score* 88.76%. Penelitian tersebut membuktikan bahwa LSTM memiliki kemampuan yang kuat dalam mengklasifikasikan teks dalam jumlah besar. Adapun dalam penelitian [14], LSTM mampu mengotomatisasi data ulasan pengguna aplikasi Alfagift ke dalam beberapa kategori dan membuktikan bahwa LSTM berhasil memberikan label dengan tepat sehingga mencapai nilai *accuracy* 95%, *precision* 94%, *recall* 93%, dan *F1-Score* 94%. Penelitian relevan lainnya dilakukan oleh [15], LSTM mampu mengklasifikasikan tingkat kepuasan para pengguna terhadap aplikasi Sirekap ke dalam beberapa label.

Tentunya sudah banyak penelitian terdahulu yang menggunakan metode LSTM dalam mengukur ulasan para pengguna terhadap suatu aplikasi di dalam *Play Store*. Akan tetapi, masih terdapat celah (*gap*) penelitian yang belum dibahas di dalam penelitian sebelumnya. Salah satunya adalah pada proses *training* dan *testing*. Pada umumnya, beberapa penelitian terdahulu tersebut hanya menggunakan satu rasio pembagian data pada saat pengujian model. Padahal, variasi rasio pembagian data dapat memengaruhi performa model secara signifikan. Selain itu, berdasarkan sisi arsitektur modelnya, sebagian besar penelitian terdahulu masih menggunakan *layer embedding* yang sederhana dan jumlahnya terbatas. Arsitektur model yang lebih dalam berpontensi untuk meningkatkan referensi fitur teks sehingga menghasilkan analisis prediksi yang lebih akurat. Oleh karena itu, penelitian ini menggunakan tiga variasi rasio pembagian data yang berbeda untuk meningkatkan performa model dan menerapkan arsitektur model dengan tiga *layers* untuk menguji kualitas model dalam klasifikasi.

3 Metode Penelitian

Metode penelitian yang digunakan adalah *text mining*, dengan model *Long Short-Term Memory* (LSTM). Pada bagian ini menjelaskan alur penelitian yang dilakukan dari awal hingga akhir setelah mengolah data sehingga mendapatkan solusi yang diinginkan. Adapun proses yang harus dilalui dalam penelitian ini, yaitu *scraping*, pra-pemrosesan, pembagian data, dan *evaluation*. Sebelum melakukan pengolahan data, telah dilakukan *labelling* dengan menggunakan *label encoder* secara otomatis. *Label encoder* pada setiap ulasan ditetapkan berdasarkan bintang yang diberikan oleh para pengguna saat menuliskan ulasan. Saat menulis ulasan, pengguna juga memberikan penilaian dengan memberikan bintang. Bintang tersebutlah yang digunakan untuk melabelkan ulasan menjadi Keluhan atau Bukan Keluhan. Adapun metodologi yang diusulkan digambarkan pada diagram alir, seperti pada Gambar 1.



Gambar 1 Metodologi yang diusulkan

3.1 *Dataset*

Dalam melakukan olah data, pemilihan *dataset* tentunya merupakan bagian yang penting. Dalam penelitian kali ini, *dataset* yang digunakan adalah data ulasan aplikasi Gojek pada Google Play Store yang berjumlah 225.002 data. Dalam model *deep learning*, sangat dibutuhkan *dataset* yang secara spesifik mewakili suatu bahasa tertentu [16]. *D2ataset* yang akan diolah pada penelitian kali ini adalah *dataset* yang menggunakan Bahasa Indonesia. *Dataset* yang digunakan berisi ulasan berupa

 ISSN:2302-8149
e-ISSN:2540-9719

teks dan juga penilaian berupa *rating* yang digunakan untuk memberikan umpan balik pada aplikasi Gojek. Data ini dapat dimanfaatkan untuk pengembangan sistem dan mengetahui kekurangan dari aplikasi yang harus segara diperbaiki agar menambah kenyamanan dari para pengguna. Adapun *dataset* tersebut memiliki isi yang digambarkan pada Tabel 1.

Tabel 1 Data ulasan gojek

Kolom	Deksripsi	Tipe Data
username	Nama pengguna yang memberikan ulasan	string
content	Isi atau teks ulasan yang diberikan oleh pengguna	string
score	Penilaian bintang (*rating*) yang diberikan	int
at	Tanggal dan waktu pada saat ulasan diberikan	datetime
appVersion	Versi aplikasi yang digunakan saat ulasan diberikan	string

Berdasarkan *dataset* yang ada, terdapat 5 label awal yang tersedia. Label tersebut dijadikan penilaian kinerja dari aplikasi Gojek. Dalam penelitian kali ini, terdapat transformasi label, yaitu label 5 yang ditransformasi ke dalam 2 label. Label diklasifikasikan menjadi 'Keluhan' dan 'Bukan Keluhan' berdasarkan dari bintang dan ulasan berupa teks yang diberikan oleh para pengguna Berdasarkan *dataset* di atas, terdapat distribusi jumlah pemberian bintang dari pengguna pada Tabel 2.

Tabel 2 Distribusi penilaian

Penilaian (Bintang)	Jumlah Ulasan	Label (*Biner*)
1 Bintang	45,229	0 (Keluhan)
2 Bintang	8,942	0 (Keluhan)
3 Bintang	9,460	0 (Keluhan)
4 Bintang	14,316	1 (Bukan Keluhan)
5 Bintang	147,055	1 (Bukan Keluhan)

Pada Tabel 2 digambarkan bahwa aplikasi Gojek ini meraih banyak penilaian bagus dari para pengguna, yaitu mendapatkan bintang 5 sebanyak 147,055 penilaian. Dalam transportasi label ini digunakan klasifikasi nilai *biner*, yaitu menggunakan nilai 0 untuk menginterpretasikan label 'Keluhan' dan nilai 1 untuk menginterpretasikan label 'Bukan Keluhan'. Berdasarkan klasifikasi ini, nantinya akan dapat dideteksi berapa banyak keluhan yang diberikan oleh para pengguna yang harus segera diatasi. Akan tetapi, penilaian hanya berdasarkan bintang saja dinilai masih kurang efektif dan membutuhkan analisis lebih lanjut terhadap ulasan – ulasan yang diberikan oleh para pengguna dalam bentuk teks.

3.2 Pra-pemrosesan

Setelah menentukan *dataset* yang ingin diolah, tahapan berikutnya dalam pengolahan data adalah pra-pemrosesan. Pra-pemrosesan merupakan tahapan yang paling awal dalam klasifikasi teks yang memiliki tujuan untuk bisa mempersiapkan data berupa teks sebelum diproses lebih lanjut [17]. Dengan adanya tahapan ini, *dataset* yang akan diolah tentunya menjadi lebih sesuai dengan model yang ingin digunakan nantinya. Dalam tahapan pra-pemrosesan ini sendiri terdapat beberapa proses lainnya, seperti *cleaning*, *stopwords*, *tokenization*, *padding* [18]. *Cleaning* dilakukan pada proses awal pra-pemrosesan. Pada proses ini, data yang akan digunakan tentunya harus dibersihkan terlebih dahulu, seperti menghapus *missing values*, dan mengambil kolom yang akan digunakan, yaitu kolom ulasan teks dan bintang.

Data yang berhasil melalui proses pembersihan berikutnya akan melalui proses tokenisasi. Tokenisasi membantu untuk mengatasi kata – kata yang tidak umum atau baru yang belum ada pada kosakata awal agar menjadi bagian yang lebih sederhana [19]. Adanya ulasan teks pada data yang akan diolah tentunya akan sulit jika diolah secara langsung. Dalam model LSTM, tokenisasi berfungsi untuk mengubah data teks yang tidak terstruktur ke dalam bentuk yang dapat diproses oleh model pembelajaran mesin dalam wujud bilangan numerik [20]. Teks yang terdapat di dalam ulasan Gojek

 ISSN:2302-8149
e-ISSN:2540-9719

nantinya akan diubah ke dalam bentuk angka berdasarkan urutan yang ada di kamus/ *tokenizer*. Adapun pembatasan kata pada saat tokenisasi juga sangatlah penting, mengingat ada banyaknya data yang diolah tentunya memiliki banyak kata – kata unik dan dapat menganggu efisiensi komputasi.

Proses pra-pemrosesan berikutnya yang digunakan di dalam penelitian ini adalah proses *padding*. Setelah melalui tokenisasi, tentunya panjang teks akan menjadi berbeda. Sebelum angka *tokenizer* diolah menggunakan model LSTM, angka tersebut harus melalui proses *padding*. *Padding* merupakan penyesuaian panjang sebuah teks dengan teks lainnya agar berukuran sama [21]. Proses *padding* dapat membantu model lebih mudah memproses data.

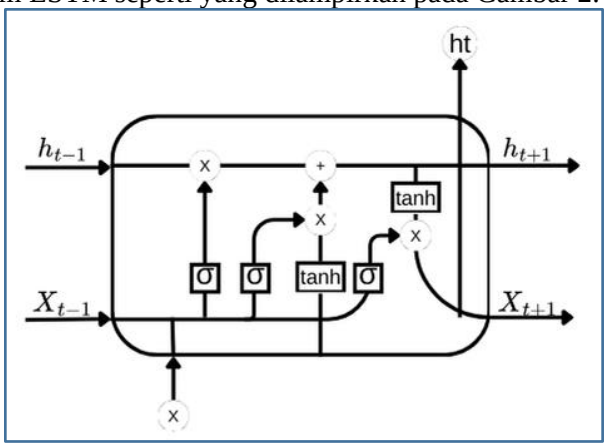
3.3 Pembagian Data

Data yang berhasil melalui proses pra-pemrosesan akan diolah ke dalam tahapan berikutnya, yaitu pembagian data. Dalam pengolahan data terdapat pembagian data, yaitu data uji dan data latih. Data latih digunakan untuk melatih model mesin dan membangun informasi baru bagi model, sedangkan model uji berfungsi sebagai alat evaluasi akurasi model dalam klasifikasi suatu ulasan [22]. Dengan adanya pembagian data, performa model akan dapat diukur dengan data latih dan nantinya bisa diterapkan pada data uji. Pembagian dilakukan dengan memanggil fungsi *train_test_split* dengan parameter *test_size*. Dalam pembagian data, terdapat beberapa proporsi, data dapat dibagi
menjadi 90% data latih dan 10% data uji dengan parameter *test_size=0.1* atau 80% data latih dan 20% data uji dengan parameter *test_size=0.2* ataupun pembagian 70% data latih dan 30% data uji dengan parameter *test_size=0.3*.

Dalam penelitian ini, digunakan tiga rasio berbeda secara langsung, untuk mengetahui apakah terdapat pengaruh dalam perbandingan data latih yang berbeda terhadap performa model dalam klasifikasi ulasan pengguna. Perbedaan tiga rasio dalam pembagian data ini dijadikan acuan untuk melihat apakah model mampu untuk tetap bekerja secara optimal dan efektif untuk mengolah data meskipun jumlah data latih dan data uji berubah. Oleh karena itu, pengujian dilakukan untuk mengetahui efektivitas sistem. Hal itu dikarenakan penelitian ini berfokus agar sistem yang dibangun nantinya dapat dengan optimal mengidentifikasi dan mendeteksi keluhan dari pengguna dengan tepat.

3.4 *Long Short-Term Memory* (LSTM)

Tahapan penelitian yang dilakukan berikutnya adalah pemodelan. Dalam penelitian kali ini, digunakan model LSTM dalam mengolah data. Meskipun fokus penelitian bukan pada arsitektur model secara mendalam, LSTM tetap dipilih karena keunggulannya dalam memproses data. LSTM sendiri memiliki komponen yang berfungsi membawa informasi selama masa *training*. Terdapat tiga gerbang yang dilalui dalam LSTM seperti yang dilampirkan pada Gambar 2.



Gambar 2 Arsitektur LSTM
Sumber: [23]

Arsitektur LSTM tersebut terdiri dari beberapa gerbang yang pertama, yaitu *forget gate* yang memiliki peran sebagai pemilah informasi yang harus dibuang dan harus digunakan [23].

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t] + b_f\right) \quad (1)$$

 ISSN:2302-8149
e-ISSN:2540-9719

Pada persamaan (1), f_t mempresentasikan *forget gate* yang digunakan untuk menenrukan proporsi informasi dari *cell state* yang akan dipertahankan sebelumya. Nilai f_t ini diperoleh dari fungsi *sigmoid* terhadap kombinasi *hidden state* sebelumnya, yaitu h_{t-1} dan *input* yang ada pada saat ini dilambangkan dengan X_t, serta dengan bobot W_f dan ditambang dengan nilai bias b_f.

Keterangan:
f_t= *forget gate*
σ = fungsi *sigmoid*
W_f = bobot *forget gate*
h_{t-1} = *output timestep* 1
X_t = *timestep* terkini
b_f = bobot bias *forget gate*

Setelah melalui *forget gate*, yang berikutnya akan dilalui adalah *input gate* [23]. Setiap *gate* tentunya memiliki peran masing – masing. *Input gate* yang berfungsi untuk mengelola banyaknya informasi yang masuk ke dalam memori (*cell state)* [24].

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (2)$$

Untuk persamaan (2), i_t mempresentasikan *input gate* yang berfungsi untuk memanajemen seberapa besar informasi baru yang masuk melalui *input* x_t dan akan disimpan di dalam *cell state*. Nilai i_t dihitung menggunakan fungsi *sigmoid* yang berasal dari gabungan *hidden state* sebelumnya, yaitu h_{t-1}. *Input* terkini ditandai dengan W_i dan bias b_i.

$$\tilde{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C) \quad (3)$$

Sementara itu, persamaan (3) menghasilkan $\tilde{C}_t$, yaitu *candidate cell state* yang berisi informasi terbaru yang akan diteruskan ke dalam *cell state*. *Candidate cell state* ini dihitung menggunakan fungsi tanh agar menghasilkan nilai *output* pada rentang $-1,1$.

Keterangan:
i_t= *input gate*
σ = fungsi *sigmoid*
W_i = bobot *input gate*
h_{t-1}= *output timestep* 1
x_t = *timestep* terkini
b_i = bobot bias *input gate*
$\tilde{C}_t$ = kandidat baru *cell state*
$tanh$ = fungsi *tanh*
W_C = bobot *cell state*
b_C = bobot bias *cell gate*

Setelah berhasil melalui perhitungan pada *input gate*, hasilnya akan dikirim ke dalam *gate* berikutnya. *Gate* yang terakhir adalah *output gate*. *Output gate* bekerja untuk memberikan hasil dari perhitungan yang sudah dijalankan [23].

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (4)$$

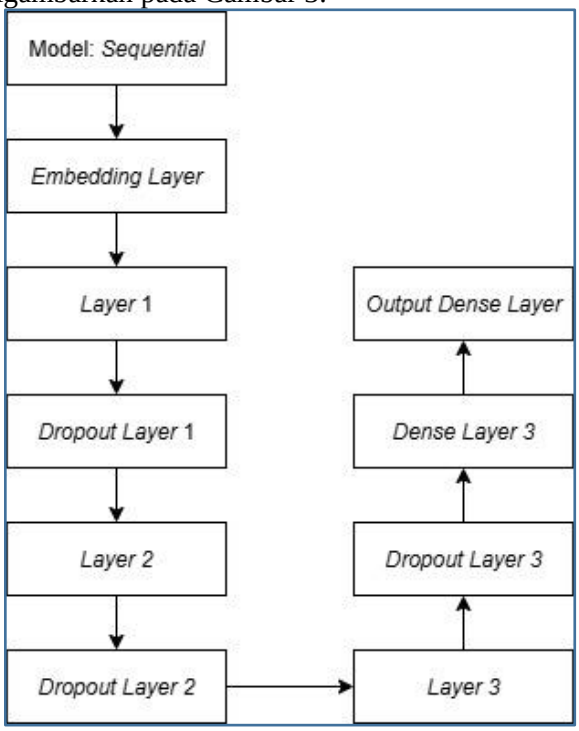
Di dalam persamaan (4), o_t mempresentasikan *output date* yang menjadi gerbang terakhir dalam arsitektur LSTM. W_o mempresentasi bobot dari *output gate* dan b_o sebagai bobot bias. Dengan adanya *output gate*, LSTM dapat mengatur ukuran besarnya informasi yang akan dikeluarkan.

Keterangan:
o_t= *output gate*
σ = fungsi *sigmoid*
W_o = bobot *output gate*
h_{t-1} = *output timestep* 1
x_t = *timestep* terkini
b_o = bobot bias *output gate*

Penelitian ini menggunakan pendekatan *sequential model* yang didampingi oleh tiga lapisan bertingkat mulai dari proses *input* hingga *output*. Adapun parameter model LSTM dibangun dengan beberapa tahapan. Tahap awal pembangunan model dimulai dengan menggunakan *embedding layer*

 ISSN:2302-8149
e-ISSN:2540-9719

berdimensi 128 dengan tiga lapisan sekaligus dengan masing – masing 64 unit *neuron* dengan tujuan agar LSTM mampu menangkap dan memahami pola kompleks dari data secara tepat. Adanya lapisan awal berfungsi untuk mengubah *input* berupa token angka yang merupakan hasil dari tokenisasi menjadi vektor yang berdimensi tempat. Dalam artian lain, *embedding layer* membantu mengartikan angka tersebut ke dalam bentuk yang bisa dimengerti oleh model LSTM. Serta terdapat parameter *dropout* 0.3 untuk mencegah adanya *overfitting*. *Overffting* merupakan sebuah kondisi yang mungkin saja terjadi apabila model mengalami turunnya performa pada data pengujian karena belum mampu memahami data baru [25]. Selanjutnya, digunakan *dense layer* dengan 32 *neuron* serta satu *ouput layer* beraktivasi *sigmoid* karena klasifikasi bersifat *biner*. Adapun susunan arsitektur yang digunakan dalam penerapan LSTM digambarkan pada Gambar 3.



Gambar 3 Arsitektur dengan *layer*

3.5 Evaluasi

Pada tahap evaluasi penelitian ini akan menginterpretasikan mengenai sistem yang dibangun menggunakan model LSTM dalam mendeteksi keluhan yang diberikan oleh para pengguna mengenai aplikasi Gojek. Beberapa metrik evaluasi yang digunakan meliputi, hasil akurasi akhir, *precision, recall* dan *F1-score.* Evaluasi digunakan untuk mengukur kemampuan model, dan melihat sejauh mana model dapat memberikan prediksi yang sesuai. Dalam penerapan model, evaluasi tentunya merupakan bagian yang cukup penting untuk melihat kinerja dari model yang digunakan. Selain itu, akan dilampirkan pula *confusion matrix* sebagai gambaran mengenai distribusi prediksi antara label sebenernya dan label yang diprediksi. Selain evaluasi kinerja terhadap model, penelitian ini juga akan memberikan evaluasi sistem hasil klasifikasi yang ditampilkan dalam bentuk *dashboard*. *Dashboard* dibangun melalui model LSTM untuk memvisualisasikan hasil *training.* Hasil evaluasi akan berupa total ulasan yang berupa Keluhan, kata – kata yang paling sering digunakan, ataupun distribusi dari setiap kata tersebut. Dengan begitu, sistem ini nantinya diharapkan dapat membantu perusahaan dalam pengambilan keputusan dengan lebih cepat dan berbasis data.

4 Hasil dan Pembahasan

Dataset yang diolah merupakan data yang berisi ulasan – ulasan yang diberikan para pengguna sebagai umpan balik aplikasi Gojek. Dengan total ulasan 225.002 yang menggunakan Bahasa Indonesia. *Dataset* tersebut diolah untuk bisa menghasilkan sistem otomatisasi pendeteksi keluhan dari para pengguna. Klasifikasi ulasan berupa keluhan dan bukan keluhan dipisahkan melalui ulasan teks dan juga bintang yang diberikan pengguna melalui Google Play Store. Pengguna yang

 ISSN:2302-8149
e-ISSN:2540-9719

memberikan penilaian bintang 1 sampai 3 diklasifikasikan sebagai keluhan dan penilaian bintang 4 sampai 5 diklasifikasikan sebagai bukan keluhan. Penelitian dibuat untuk memudahkan tim layanan mengetahui keluhan yang harus mereka atasi.

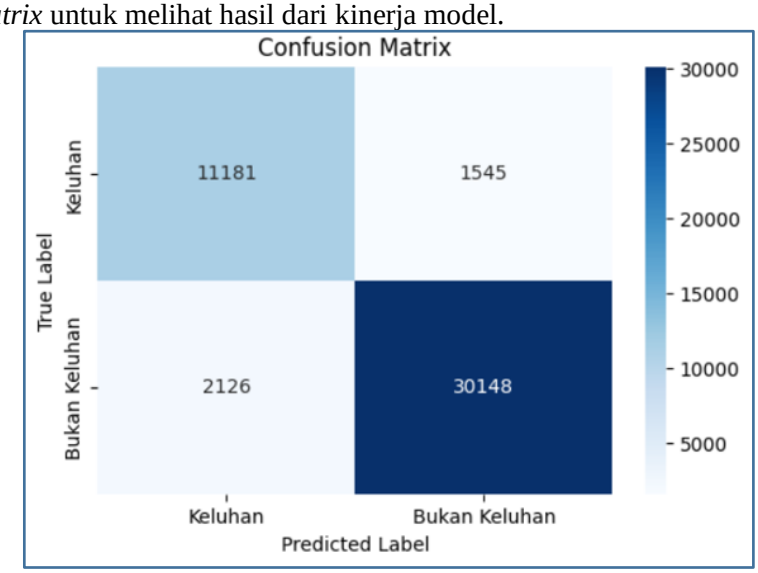
4.1 Pemodelan LSTM

Dalam penelitian ini dilakukan *training* data sebanyak tiga kali dengan rasio pembagian data yang berbeda. Hal itu dilakukan untuk melatih model agar lebih memahami pola dalam data, mengetahui apakah pembagian data yang berbeda akan memberikan pengaruh besar terhadap pengolahan data dan untuk menilai stabilitas dan konsistensi dari model LSTM. Maka dari itu, digunakan tiga rasio pembagian data yang berbeda, hasil dari penggunaan rasio pembagian data dapat dilihat pada Tabel 3.

Tabel 3 Perbandingan rasio pembagian data

Rasio Pembagian Data	Hasil Akurasi
90:10	92,93%
80:20	91,84%
70:30	91,96%

Berdasarkan hasil yang dilampirkan pada Tabel 3, dapat diketahui bahwa pembagian data dengan rasio 80:20 menghasilkan nilai akurasi yang paling besar walaupun hanya memiliki perbedaan nilai yang sangat kecil. Hasil tersebut menunjukkan bahwa rasio pembagian 80:20 memberikan keseimbangan yang paling optimal bagi data latih dan data uji. Akan tetapi, dengan perbandingan hasil akurasi yang tidak terlalu banyak di antara rasio pembagian data membuktikan bahwa model yang dibangun sudah cukup stabil dan tidak terlalu dipengaruhi oleh perubahan proporsi data. Untuk memastikan performa model sudah cukup baik untuk digunakan mendeteksi label keluhan, digunakan pula *confusion matrix* untuk melihat hasil dari kinerja model.

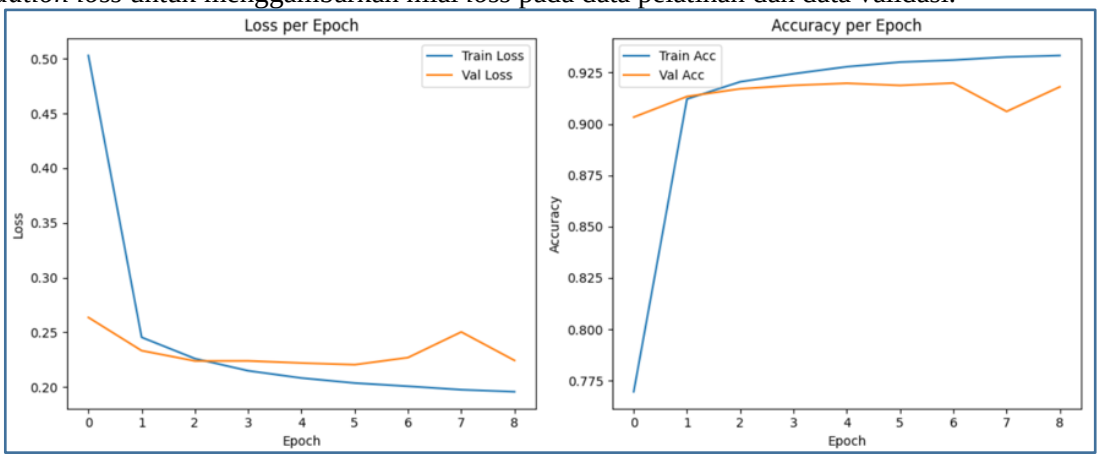


Gambar 4 *Confusion matrix* 80:20

Dalam *confusion matrix* yang dilampirkan pada Gambar 4, didapatkan bahwa model berhasil mengklasifikasikan label yang diminta. Model LSTM berhasil mendeteksi total keluhan yang ada pada data yang diberikan. Dalam *confusion matrix*, LSTM mampu mengklasifikasikan 11.181 ke dalam label Keluhan dengan benar, yang menandakan bahwa LSTM dapat membedakan bahwa ulasan tersebut mengandung kata keluhan. Walaupun LSTM masih salah mengklasifikasikan ulasan ke dalam label keluhan sebanyak 1545 ulasan yang dideteksi sebagai keluhan, padahal harusnya bukan keluhan. Terdapat juga 2126 ulasan yang gagal dideteksi sebagai keluhan sehingga tidak terhitung ke dalam label keluhan. Untuk label bukan keluhan, LSTM berhasil melakukan prediksi terhadap 30.148 ulasan. Adapun didapatkan nilai *precision* 92%, *recall* 92% dan F1-*score* 92%. Berdasarkan *confusion matrix* dan *classification report* dapat mengindikasikan bahwa model cukup

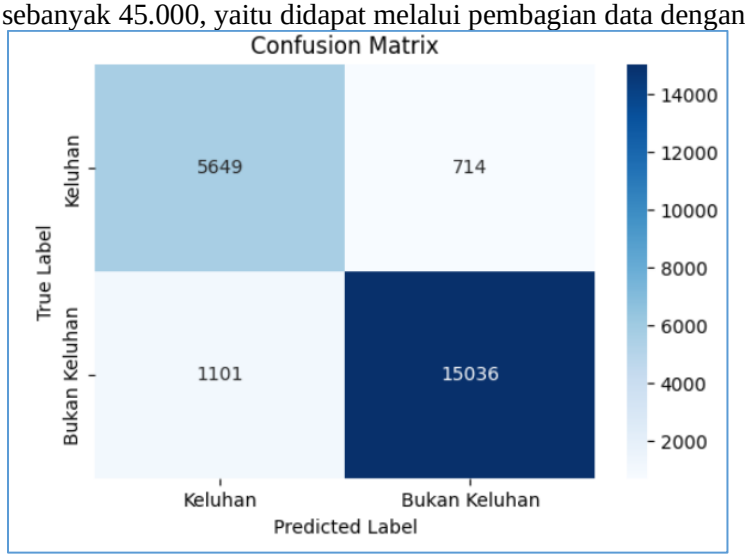
 ISSN:2302-8149
e-ISSN:2540-9719

seimbang dalam mengklasifikasikan label keluhan dan bukan keluhan. Adapun kurva *train loss* dan *validation loss* untuk menggambarkan nilai *loss* pada data pelatihan dan data validasi.



Gambar 5 Kurva *Loss* dan *Accuracy* 80:20

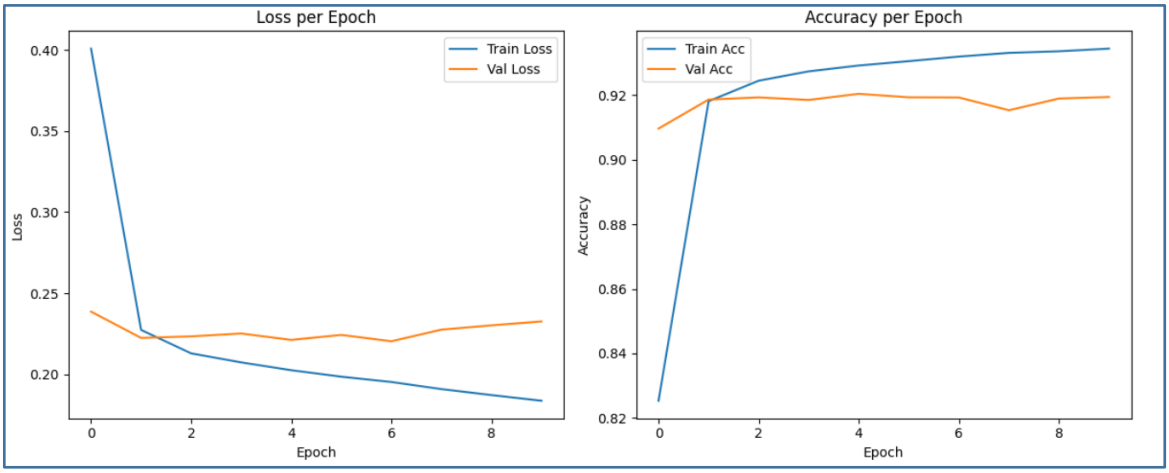
Pada Gambar 5, dapat dilihat bahwa nilai *loss* pada data pelatihan cenderung menurun sehingga hampir mendekati stabil pada saat *epoch* 3. Akan tetapi, terdapat kenaikan nilai *loss* pada data validasi yang diyakini sebagai gejala *overfitting* ringan, di mana model LSTM terlalu menyesuaikan diri dengan data pelatihan. Adapun pada grafik *train accuracy* menampilkan kenaikan yang cukup tajam hingga mencapai angka 92% sejak pada titik *epoch* 2. Nilai *validation accuracy* juga menunjukan bahwa model LSTM sudah positif dan cukup stabil, meskipun masih mengalami sedikit penurunan pada *epoch* akhir. Secara keseluruhan, kurva ini menginterpretasikan bahwa LSTM berhasil mempelajari data dengan baik sehingga dapat memberikan hasil klasifikasi yang baik di antara dua label yang diminta. Tentunya hasil yang didapatkan pada *confusion matrix* dan juga kurva hanya memuat data sebanyak 45.000, yaitu didapat melalui pembagian data dengan rasio 80:20.



Gambar 6 *Confusion matrix* 90:10

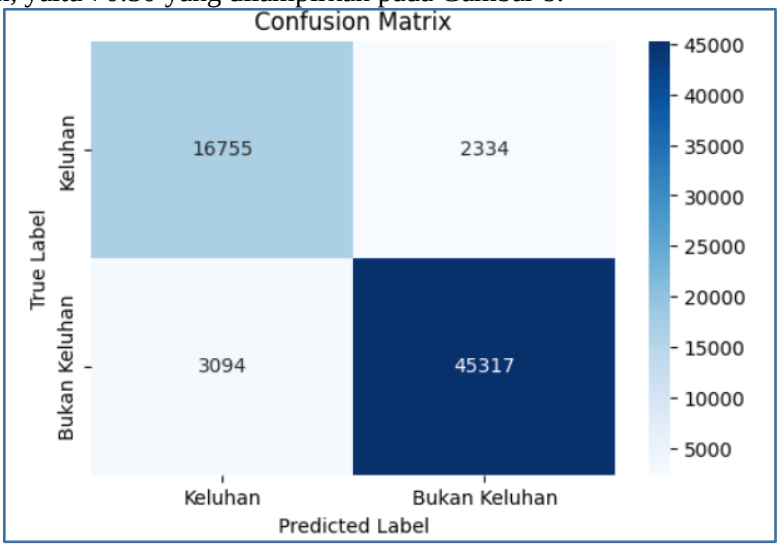
Adapun pada pembagian data dengan rasio 90:10, didapatkan hasil *confusion* matrix pada Gambar 6. Dalam rasio ini, LSTM mengenali label Keluhan sebanyak 5.649 ulasan. Jumlah yang diklasifikasikan jumlahnya lebih sedikit dibandingkan rasio 80:20. Untuk label Bukan Keluhan, LSTM mampu mengenali 15.036 ulasan. Akan tetapi, masih ada ulasan yang diklasifikasikan ke dalam label yang salah, dengan total jumlah ulasan sebanyak 1.815. Adapun didapatkan nilai *precision* 89%, *recall* 91% dan F1-*score* 90%. Pembagian data dengan rasio 90:10 juga menghasilkan kurva yang cukup baik yang.

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 7 Kurva *loss* dan a*ccuracy* 90:10

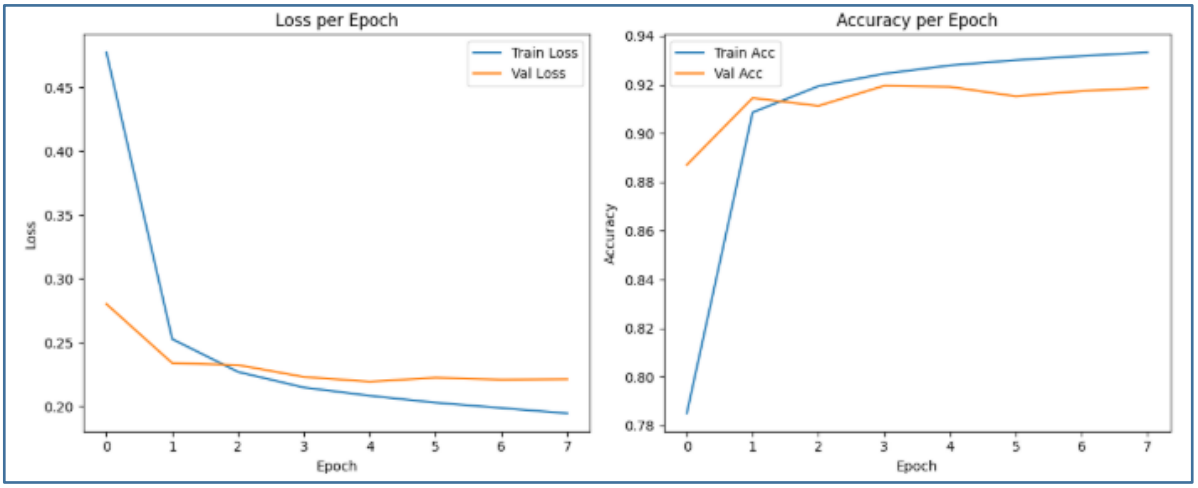
Di dalam kurva yang dilampirkan pada Gambar 7, dapat dianalisis bahwa data pelatihan mengalami *loss* dari awal hingga akhir yang berarti model mampu memahami pola – pola dari data pelatihan. Akan tetapi, pada data validasi cenderung stabil dan bertahan posisi yang sama, bahkan sedikit mengalami kenaikan nilai *loss* pada *epoch* 6 hingga 8. Kenaikan *loss* tersebut menggambarkan bahwa model sempat kehilangan kemampuan dalam generalisasi daya yang belum pernah dikenali sebelumnya. Hal serupa terjadi pada kurva *accuracy*, di mana data pelatihan terus meningkat, namun data validasi mengalami stagnasi hingga *epoch* akhir. Adapun pada rasio pembagian data yang terakhir digunakan, yaitu 70:30 yang dilampirkan pada Gambar 8.



Gambar 8 *Confusion matrix* 70:30

Berdasarkan *confusion matrix* yang dilampirkan pada Gambar 8, dengan rasio 70:30, LSTM masih mampu mengklasifikasikan label dengan cukup baik. Rasio pembagian data ini berhasil mengklasifikasikan 16.755 ulasan sebagai label Keluhan. Rasio 70:30 berhasil mengklasifikasikan label Keluhan dengan jumlah terbanyak dibandingkan dengan rasio pembagian data yang lain. Hal itu tentunya juga dipengaruhi oleh jumlah data pelatihan yang lebih besar jumlahnya, yaitu 30%. Serta, terdapat 45.317 ulasan yang dikenali sebagai Bukan Keluhan. Rasio pembagian data ini juga menghasilkan kurva yang cenderung membaik di setiap *epoch* yang dilalui. Pada kurva *loss*, data pelatihan dan data validasi berhasil mengalami penurunan dalam *loss* data. Akan tetapi, pada beberapa *epoch*, data validasi masih sedikit mengalami kenaikan hilangnya data. Serta, pada kurva akurasi, data pelatihan mengalami kenaikan akurasi yang pesat. Sedangkan, data validasi masih mengalami fluktuatif seperti yang diintrepretasikan pada Gambar 9.

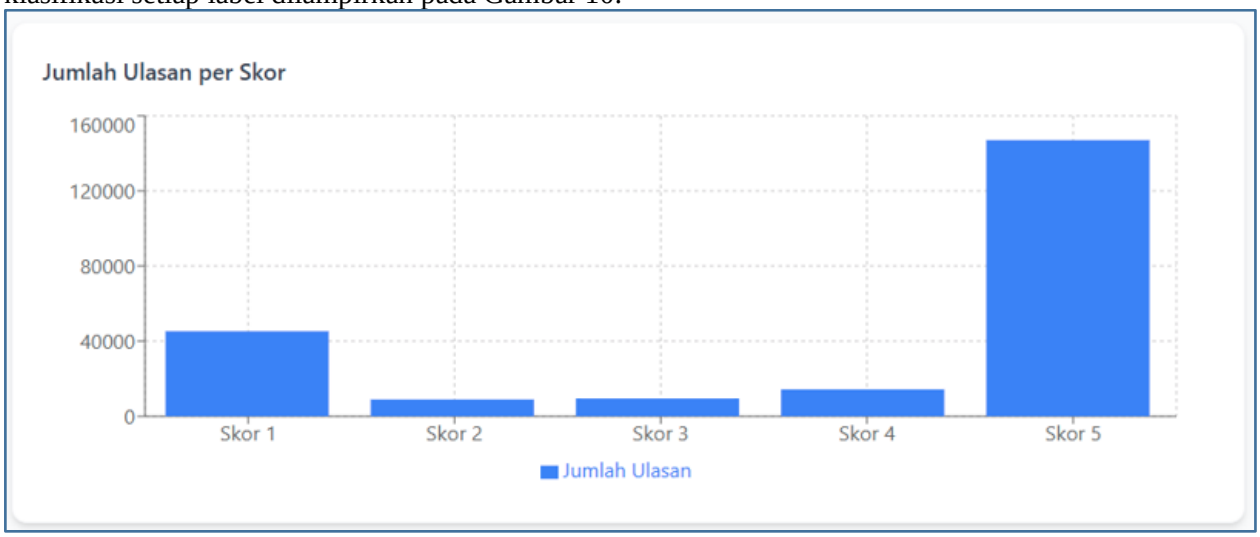
 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 9 Kurva *loss* dan *accuracy* 70:30

4.2 Hasil Evaluasi

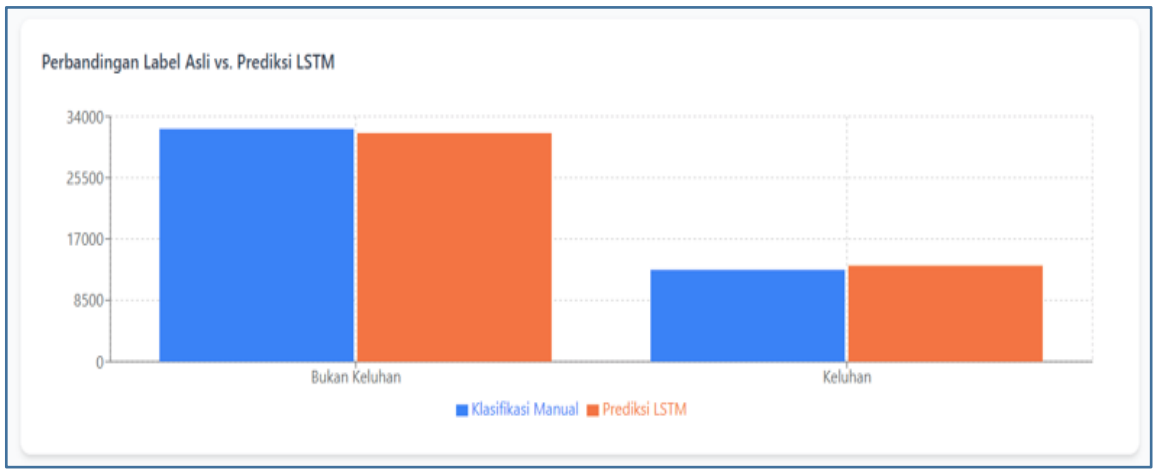
Berdasarkan perbedaan rasio pembagian data yang digunakan, tentunya akan memengaruhi jumlah ulasan yang diklasifikasikan. Oleh karena itu, dibutuhkan evaluasi yang lebih mendalam lagi dalam klasifikasi dua label dengan total seluruh ulasan, bukan hanya sebagian. Dalam penelitian ini, kami juga menggunakan hasil evaluasi melalui *dashboard* untuk memvisualisasikan lebih dalam dan secara keseluruhan data yang ada, bukan hanya sebagian. Evaluasi *dashboard* didapatkan melalui hasil pemodelan dengan LSTM dengan yang sudah dilakukan sebelumnya. Adapun diagram distribusi klasifikasi setiap label dilampirkan pada Gambar 10.



Gambar 10 Distribusi klasifikasi label

Pada diagram batang dapat dilihat bahwa penilaian terbaik, yaitu bintang 5 mendapatkan total terbanyak dibandingkan penilaian bintang lainnya. Klasifikasi ini menunjukkan bahwa ulasan para pengguna terhadap Gojek sudah cukup baik. Akan tetapi, tidak sedikit juga pengguna yang memberikan ulasan berupa keluhan yang didapatkan melalui distribusi bintang 1, 2 dan 3. Distribusi ulasan yang diklasifikasikan sebagai keluhan hampir mencapai angka 8.000. Keluhan – keluhan yang sudah dideteksi oleh sistem ini tentunya tidak boleh dibiarkan saja. Perusahaan terkhususnya tim layanan pengguna, harus melihat dan mengetahui keluhan di bagian apa saja yang paling banyak diberikan oleh pengguna. Dengan mengetahui titik kekurangan dari aplikasi, tim pelayanan akan lebih mudah untuk mencari solusi untuk mengatasi gangguan yang dialami oleh para pengguna. Selain itu, sistem *dashboard* ini juga mampu membandingkan label asli dan label prediksi yang didapatkan melalui *training* LSTM.

 ISSN:2302-8149
e-ISSN:2540-9719



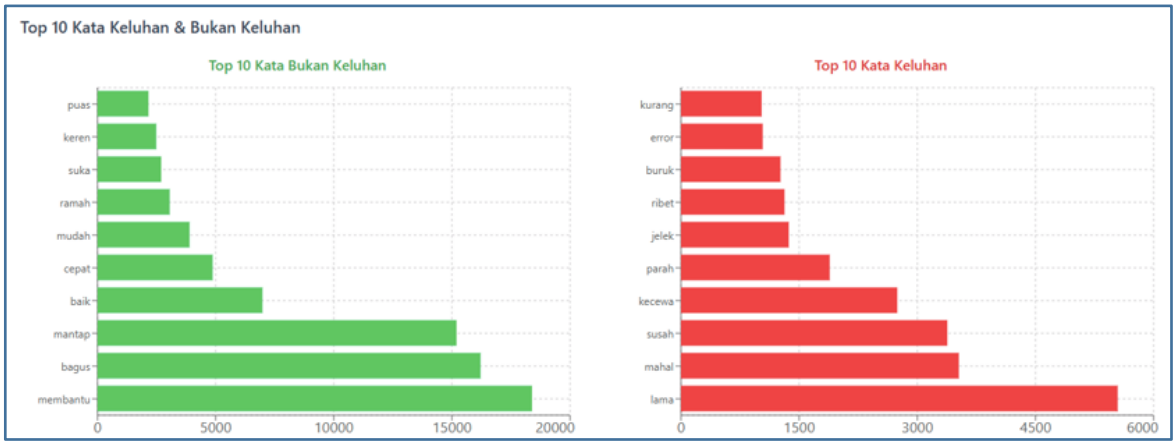
Gambar 11 Perbandingan label

Pada diagram yang ada di Gambar 11, dapat dilihat bahwa LSTM mampu mengklasifikasikan label dengan sangat baik. Klasifikasi label asli yang ada dan label hasil prediksi LSTM memang tidak sama persis. Akan tetapi, perbandingan hasil dari klasifikasi label tersebut juga tidak mengalami perbedaan yang sangat jauh. Bahkan, untuk pendeteksi label keluhan, LSTM menghasilkan prediksi yang mendeteksi lebih banyak ulasan dibandingkan label asli. Diagram ini menandakan bahwa LSTM mampu mengenali dan mengklasifikasikan seluruh ulasan ke dalam label dengan cukup tepat.

Gambar 12 Kata kunci

Dashboard pada juga mampu merancang *word cloud* atau kata kunci dari seluruh ulasan yang ada seperti pada Gambar 12. Kata – kata yang ditulis dengan ukuran *font* yang lebih besar berarti merupakan kata yang paling sering muncul dan kata dengan ukuran yang paling sering merupakan kata yang jarang sekali muncul. Pada kata kunci untuk ulasan yang bukan keluhan, kata 'membantu, bagus dan mantap' merupakan kata – kata yang paling sering muncul dan kata 'diskon, *voucher*, aman' merupakan kata – kata yang jarang muncul. Sedangkan, pada kata kunci ulasan yang termasuk keluhan terdapat kata 'nya, aplikasi, gak' sebagai kata yang paling sering muncul dan kata 'nunggu, parah, 'kecewa' merupakan kata yang jarang muncul. Dengan adanya bantuan pendeteksi keluhan ini, tim pelayanan mengetahui bahwa salah satu kekurangan dari Gojek yang paling sering dikeluhkan pengguna adalah harga tinggi dari jasa layanan yang mereka berikan sedikit memberatkan pengguna. Selain itu, sistem juga mampu mengklasifikasikan 10 kata yang paling sering digunakan di dalam data ulasan Gojek. Hampir sama dengan penggunaan *word cloud* pada Gambar 12, namun untuk klasifikasi berikut ini lebih berfokus pada kata – kata yang paling banyak digunakan saja. Adapun diagram distribusi klasifikasi setiap kata yang paling sering muncul terdapat pada Gambar 13.

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 13 Klafikasi kata ulasan

Diagram batang pada Gambar 13 menunjukan hasil distribusi dari 10 kata yang paling banyak muncul di dalam ulasan Gojek. Untuk ulasan yang berupa bukan keluhan, kata 'membantu' menempati tempat pertama sebagai kata yang paling sering digunakan dengan total distribusi hampir mencapai 20.000 kali dipakai. Adapun kata 'bagus, mantap dan baik' yang juga menjadi kata – kata yang cukup sering digunakan hingga muncul dalam ulasan melebihi 5.000 kali. Untuk ulasan yang ditandai sebagai keluhan, kata 'lama' mendapat tempat sebagai kata yang paling sering digunakan oleh pengguna hingga hampir mencapai angka 6.000. Selain itu, terdapat kata 'mahal, susah, kecewa' sebagai kata – kata yang cukup sering muncul di dalam ulasan yang mencapai angka melebihi 2.000 kali digunakan. Diagram ini menunjukan hasil yang lebih rinci dari dibandingkan dengan *word cloud* sebelumnya. Diagram ini membantu perusahaan untuk mengetahui bahwa bukan hanya satu atau dua pengguna saja yang mengalami gangguan yang sama. Dengan adanya pengklasifikasian setiap kata seperti ini, Perusahaan Gojek bisa dengan mudah mengetahui kelemahan dari aplikasi yang harus segera diatasi dan keunggulan dari aplikasi yang harus dipertahankan.

5 Kesimpulan

Penelitian ini bertujuan untuk membangun sistem pendeteksi keluhan pada ulasan yang diberikan oleh para pengguna Gojek melalui Google Play Store. Sistem ini dibangun agar tim pelayanan bisa dengan mudah mengetahui keluhan apa saja yang diberikan oleh para pengguna. Keluhan yang sudah dideteksi dengan sistem tentunya akan lebih mudah untuk diselesaikan. Sistem ini dibangun menggunakan model LSTM yang memanfaatkan data – data berupa ulasan teks dan bintang yang diberikan sebagai bentuk penilaian terhadap Gojek. Model LSTM digunakan untuk melatih data dengan total 225.002 ulasan yang menggunakan tiga perbandingan rasio yang berbeda. Rasio perbandingan 90:10 menghasilkan nilai akurasi sebesar 92,1%, rasio 80:20 menghasilkan nilai akurasi sebesar 91,84% dan rasio 70:30 menghasilkan nilai akurasi sebesar 91,87%. Hasil tersebut membuktikan bahwa LSTM mampu melatih data dan mengklasifikasikannya dengan tepat. *Confusion matrix* dan juga kurva yang dilampirkan menggambarkan bahwa LSTM berhasil melatih data dengan tepat sehingga perbandingan label asli dan prediksi label tidaklah terpaut jauh. Bahkan, kurva juga menunjukkan berkurangnya data *loss* selama pelatihan. Selain itu, penelitian ini menggunakan *dashboard* sebagai sarana visualisasi dari sistem pendeteksi yang sudah dibangun. Terdapat diagram sebagai gambaran distribusi setiap bintang yang diberikan, terdapat diagram perbandingan label asli dan label hasil prediksi serta terdapat *word cloud* dan kata yang paling banyak digunakan di dalam ulasan guna mempermudah tim layanan mengetahui kekurangan yang dirasakan pengguna dari Gojek. Sistem dapat membantu perusahaan untuk mengidentifikasi keluhan secara otomatis tanpa memakan banyak waktu, sistem mampu menyusun prioritas keluhan yang harus segera diatasi, dan sistem juga dapat meningkatkan kualitas pelayanan yang dikeluhkan pengguna. Dengan kata lain, sistem ini mampu dijadikan alat bantu analisis umpan balik yang diberikan oleh para pengguna yang dapat diintegrasikan ke dalam strategi manajemen layanan perusahaan.

 ISSN:2302-8149
e-ISSN:2540-9719

Referensi

[1] T. M. Adilah, H. Supendar, R. Ningsih, S. Muryani, and K. Solecha, "*Sentimen Analysis of Online Transportation Service using the Naive Bayes Methods*," *J Phys Conf Ser*, Vol. 1641, No. 1, p. 012093, 2020.

[2] D. Nugraha and D. Gustian, "Analisis Sentimen Penggunaan Aplikasi Transportasi *Online* pada *Ulasan Google Play Store* Metode *Naive Bayes Classfier*," *Jurnal Penerapan Sistem Informasi*, Vol. 5, No. 1, pp. 326–335, Jan. 2024.

[3] S. Heristian, M. Napiah, and W. Erawati, "Analisis Sentimen Ulasan Pelanggan menggunakan *Algoritma Naive Bayes* pada Aplikasi Gojek," *Jurnal Computer Science*, Vol. 5, No. 1, pp. 35–41, Jan. 2025. DOI: https://doi.org/10.31294/coscience.v5i1.7775.

[4] J. A. Wibowo, V. C. Mawardi, and T. Sutisno, "Penerapan *Support Vector Machine* untuk Analisis Sentimen Fitur Layanan pada Ulasan Gojek," *Jurnal Ilmu Komputer dan Sistem Informasi*, Vol. 12, No. 1, pp. 1–8, Jan. 2024. DOI: https://doi.org/10.24912/jiksi.v12i1.28211.

[5] N. Z. Rania and R. D. Syah, "Analisis Sentimen Terhadap Aplikasi Gojek pada *Play Store* menggunakan Metode *Random Forest Classfier*," *Jurnal Ilmiah Informatika Komputer*, Vol. 29, No. 2, pp. 144–153, Aug. 2024. tersedia pada: 10.35760/ik.2024.v29i2.11877.

[6] A. Hermawan, I. Jowensen, Junaedi, and Edy, "Implementasi *Text-Mining* untuk Analisis Sentimen pada *Twitter* dengan *Algoritma Support Vector Machine*," *Jurnal Sains dan Teknologi*, Vol. 12, No. 1, pp. 129–137, Apr. 2023. DOI: https://doi.org/10.23887/jstundiksha.v12i1.52358.

[7] S. Khairunnisa and A. S. Al Faraby, "Pengaruh *Text Preprocessing* terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial *Twitter* (Studi Kasus Pandemi COVID-19)," *Jurnal Media Informatika Budidarma*, Vol. 5, No. 2, pp. 406–414, Apr. 2021. DOI:https://doi.org/10.30865/mib.v5i2.2835.

[8] E. V. Anthony and S. P. Barus, "Perbandingan Algoritma *Machine Learning* dan *Deep Learning* dalam Analisis Sentimen Komentar Video YouTube Berjudul Hidup Tanpa Sosial Media," *Jurnal Informatika dan Komputer*, Vol. 9, No. 2, pp. 370–380, Jun. 2025. DOI: http://dx.doi.org/10.26798/jiko.v9i2.1619.

[9] Y. K. Bintang and H. Immanuddin, "Pengembangan Model *Deep Learning* untuk Deteksi Retinopati Diabetik menggunakan Metode *Transfer Learning*," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, Vol. 9, No. 2, pp. 1442–1455, Sep. 2024. DOI: https://doi.org/10.29100/jipi.v9i3.5588.

[10] M. S. Raiya, M. R. P. Khamil, N. Fadillah, and R. A. Saputra, "Implementasi Algoritma *Long Short Term Memory* pada Isu Kenaikan Uang Kuliah Tunggal terhadap Minat Kuliah Mahasiswa," *Jurnal Informatika Terpadu*, Vol. 11, No. 1, pp. 37–43, Apr. 2025. DOI: https://doi.org/10.54914/jit.v11i1.1656.

[11] A. Rolagon, A. Weku, and G. A. Sandag, "Perbandingan Algoritma LSTM untuk Analisis Sentimen Pengguna *Twitter* terhadap Layanan Rumah Sakit saat Pandemi Covid-19," *Jurnal TelKa*, Vol. 13, No. 1, pp. 31–40, Apr. 2023. DOI: https://doi.org/10.36342/teika.v13i01.3063.

[12] F. Putra, R. M. Ihsan, H. F. Tahiyat, L. Efrizom, and Rahmaddeni, "Evaluasi Performa Aplikasi Gojek melalui Klasifikasi Kata Ulasan Pengguna dengan Metode SVM," *Jurnal Teknologi Informasi*, Vol. 23, No. 2, pp. 704–715, Aug. 2024. DOI: https://doi.org/10.62411/tc.v23i3.11379.

[13] C. N. Daiman, A. Y. Rahman, and F. Nudiyansyah, "Klasifikasi Teks Berita *Breaking News* di Manggarai menggunakan *Long Short Term Memory (LSTM)*," *Jurnal Ilmu Komputer dan Teknik Informatika*, Vol. 7, No. 2, pp. 170–174, Sep. 2024. DOI: https://doi.org/10.36040/mnemonic.v7i2.9939.

[14] Juniardi, Tukino, B. Priyatna, and S. S. Hilabi, "Klasifikasi Sentimen Analisis Ulasan Aplikasi Alfagift menggunakan *Algoritma Long Short Term Memory* ," *Jurnal Ilmiah Teknik dan Ilmu Komputer*, Vol. 4, No. 2, pp. 48–55, May 2025. DOI: https://doi.org/10.55123/storage.v4i2.5138.

[15] E. Cahyono and S. Kacung, "Analisis Tingkat Kepuasan Pengguna Sirekap menggunakan Metode *Support Vector Machine* dan *Long Short Term Memory*," *Jurnal Informasi Teknologi dan Sains*, Vol. 7, No. 2, pp. 913–920, May 2025. DOI: https://doi.org/10.51401/jinteks.v7i2.5867.

 ISSN:2302-8149
e-ISSN:2540-9719

[16] M. Novela and T. Basaruddin, "Dataset Suara dan Teks Berbahasa Indonesia pada Rekaman *Podcast* dan *Talk Show*," *Jurnal Fasilkom*, Vol. 11, No. 2, pp. 61–66, Aug. 2021. DOI: https://doi.org/10.37859/jf.v11i2.2628.

[17] A. F. Aufar, M. A. Rosid, A. Eviyanti, and I. R. I. Astutik, "Mengoptimalkan *Preprocessing* Teks untuk Analisis Sentimen yang Akurat pada Ulasan *E-Wallet*," *Journal of Information and Computer Technology Education*, Vol. 7, No. 2, pp. 42–50, Oct. 2023. DOI: https://doi.org/10.21070/ups.6279.

[18] H. Syahputra, L. K. Basyar, and A. A. S. Tamba, "*Sentimen Analysis of Public Opinion on the Gojek Indonesia Through Twitter Using Algorithm Support Vector Machine*," *J Phys Conf Ser*, Vol. 1462, No. 012063, 2020.

[19] A. Aljabar and B. M. Karomah, "Mengungkap Opini Publik: Pendekatan *BERT-basedcaused* untuk Analisis Sentimen pada Komentar Film," *Journal of System and Computer Engineering*, Vol. 5, No. 1, pp. 36–43, Jan. 2024.

[20] Susilawati, Z. Sitorus, M. Iqbal, D. Nasution, and R. F. Wijaya, "Penerapan *Deep Learning* dan Analisis Sentimen terhadap *Gap* Kompetensi Lulusan Lembaga Pendidikan dan Pelatihan Vokasi terhadap Dunia Kerja dengan Metode *Long Short-Term Memory (LSTM)*," *Jurnal Bulletin Of Information Technology*, Vol. 6, No. 2, pp. 161–172, Jun. 2025. DOI: https://doi.org/10.47065/bit.v6i2.2031.

[21] R. Z. N. Ahmad, N. S. Harahap, S. Agustian, I. Iskandar, and S. Sanjaya, "Perbandingan Performa *Random Forest* dan *Long Short-Term Memory* dalam Klasifikasi Teks Multilabel Terjemahan Hadits Bukhari," *Indonesia Journal of Machine Learning and Computer Science*, Vol. 5, No. 3, pp. 862–874, Jul. 2025. DOI: https://doi.org/10.57152/malcom.v5i3.2046.

[22] B. U. Fahnun and S. A. S. Muhammad, "Perbandingan Model *Machine Learning* dalam Analisis Sentimen Ulasan Pengguna Aplikasi *E-Commerce*," *Jurnal Ilmiah Teknologi dan Rekayasa*, Vol. 30, No. 1, pp. 73–94, 2025. DOI: http://dx.doi.org/10.35760/tr.2025.v30i1.13530.

[23] A. Winata, M. Lauro, and T. Handhayani, "Perbandingan LSTM dan ELM dalam Memprediksi Harga Pangan Kota Tasikmalaya," *Jurnal Ilmu Komputer dan Sistem Informasi*, Vol. 11, No. 2, Aug. 2023. DOI: https://doi.org/10.24912/jiksi.v11i2.26015.

[24] F. Landi, L. Baladi, M. Cornia, and R. Cucchiara, "*Working Memory Connections for LSTM*," *Neural Network*, Vol. 144, pp. 334–341, 2021. DOI: https://doi.org/10.1016/j.neunet.2021.08.030.

[25] E. Verianto, "Penerapan LSTM dengan Regularisasi untuk Mencegah *Overfitting* pada Model Prediksi Tingkat Inflasi di Indonesia," *Jurnal Sistem Informasi dan Sistem Komputer*, Vol. 9, No. 2, Jul. 2024. DOI: https://doi.org/10.51717/simkom.v9i2.460.