

ANALISIS KLASIFIKASI SENTIMEN PENGGUNA APLIKASI PEDULILINDUNGI BERDASARKAN ULASAN DENGAN MENGGUNAKAN METODE LONG SHORT TERM MEMORY

Ghifari Ahmad Lustiansyah¹, Didit Widiyanto², Bambang Tri Wahyono³
Informatika

Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Daerah Khusus Ibukota Jakarta 12450
ghifariahmad@upnvj.ac.id, didit.widiyanto@upnvj.ac.id, bambang.triwahyono@upnvj.ac.id

Abstrak. Aplikasi PeduliLindungi dibangun sebagai survei kesehatan dimasa pandemik COVID-19. Aplikasi ini digunakan sebagai wadah untuk penanganan dan memudahkan penelusuran riwayat kontak dengan penderita Covid-19 dapat lebih mudah ditangani dan segera dilakukan. Setelah aplikasi ini dirilis di Google Play Store banyak ulasan yang dikomentari oleh pengguna dari aplikasi ini. Oleh karena itu, penelitian ini akan menganalisis sentimen dari ulasan yang diberikan oleh pengguna aplikasi PeduliLindungi dengan menggunakan metode Long Short Term Memory. Penggunaan metode ini diharapkan mendapatkan akurasi yang tinggi sehingga dengan metode tersebut dapat mengklasifikasi komentar negatif dan komentar positif sehingga mendapatkan evaluasi yang dapat meningkatkan pelayanan kepada masyarakat melalui aplikasi ini. Tahapan-tahapan yang dilakukan dalam penelitian ini menggunakan tahapan pra-proses data seperti case folding, filtering, normalisasi kata, stopword removal, stemming, dan tokenisasi. Setelah itu data dilakukan pelatihan dengan menggunakan model LSTM dan mendapatkan hasil akurasi sebesar 82,44%, presisi sebesar 78,66% serta recall sebesar 87,03%.

Kata Kunci: PeduliLindungi, Long Short Term Memory, Analisis Sentimen, Google Play Store

1 Pendahuluan

Berdasarkan Keputusan Menteri Komunikasi dan Informatika Nomor 171 Tahun 2020, Menteri Komunikasi dan Informatika telah mengembangkan sebuah aplikasi bernama PeduliLindungi sebagai pendukung survei kesehatan, melengkapi Keputusan Menteri Komunikasi dan Informatika Nomor 159 Tahun 2019, tentang Upaya Penanganan virus disease melalui dukungan sektor pos dan Informatika[1]. Aplikasi ini digunakan sebagai wadah untuk penanganan dan memudahkan masyarakat ketika bepergian dengan membagikan data lokasi sehingga penelusuran riwayat kontak dengan penderita Covid-19 dapat lebih mudah ditangani dan segera dilakukan[2].

Penelitian ini akan berfokus kepada ulasan yang diberikan pengguna terhadap aplikasi PeduliLindungi. Ulasan yang diberikan dari pengguna ini bisa dipengaruhi oleh hal-hal yang belum menjadi perhatian dari pengembangannya yaitu Kementerian Komunikasi dan Informatika. Sudah banyak ulasan yang diberikan oleh pengguna Aplikasi PeduliLindungi sejak aplikasi ini resmi diluncurkan dari mulai komentar negatif mengenai salahnya informasi keamanan lokasi pengguna dari orang yang terpapar Covid-19, tidak munculnya sertifikat vaksinasi, dan informasi zona merah, kuning. Oleh karena itu, pemerintah khususnya KOMINFO harus tahu kepuasan dari masyarakat terhadap aplikasi PeduliLindungi sehingga dapat meningkatkan kualitas fitur aplikasi tersebut. Kepuasan pengguna aplikasi PeduliLindungi dapat dianalisis dengan melakukan analisis sentimen ulasan dari Aplikasi PeduliLindungi. Jadi dapat menjadi evaluasi bagi pengembang untuk meningkatkan kualitas serta pelayanan dari aplikasi PeduliLindungi.

2 Metodologi Penelitian

2.1 Analisis Sentimen

Pandangan yang didapat dari perasaan berlebihan terhadap sesuatu dapat diartikan sebagai sentimen. Sebuah sentimen sering ditemukan pada kalimat pernyataan dan kalimat yang memiliki opini. Studi komputasional yang mengelompokkan polaritas dari beberapa opini orang lain pada sebuah teks dalam dokumen dan terdapat di dalam entitas, dan atribut merupakan definisi dari analisis sentimen atau opinion mining[3].

Analisis sentimen memiliki tujuan dalam menentukan pengambilan keputusan dengan cara menentukan kesukaan atau ketidaksukaan seseorang terhadap sesuatu dan biasanya nilai yang sering digunakan adalah positif dan negatif.

2.2 Web Scraping

Web Scraping merupakan sebuah Proses yang mengambil informasi dokumen yang memiliki bentuk semi terstruktur dan diekstrak menjadi data terstruktur yang dapat digunakan untuk kepentingan lainnya. Teknik dilakukan melalui Teknik regular expression, yaitu dengan cara memanipulasi dan mencocokkan string yang diinginkan[4].

2.3 Text Mining

Text mining memiliki proses yang lebih banyak daripada data mining walaupun text mining merupakan bagian dari data mining. Hal ini terjadi karena text mining memproses data yang tidak terstruktur dan perlu dilakukan beberapa tahapan proses untuk mengubah data tersebut menjadi data yang lebih terstruktur. Text mining memiliki beberapa fungsi utama yaitu, Searching, Information, Exctraction, Categorization, Summarization, Prioritization, Clustering, Information Monitor dan Question & answer[5].

2.4 Text Preprocessing

Preprocessing ini dilakukan untuk mengubah dan mengekstrak kata-kata yang tidak terstruktur yang bersumber dari dokumen dan diubah menjadi bentuk yang mudah dipahami dan berguna. Preprocessing ini merupakan bagian yang sangat penting dan sangat kritis di dalam data mining karena data yang diperoleh dari halaman website memiliki banyak kata yang tidak baku dan memiliki banyak noise[6].

a) Case Folding

Data yang berasal dari sebuah halaman website memiliki ke tidak konsisten dalam penulisannya seperti penggunaan huruf kapital, spasi, kata baca, dan lain-lain. Case Folding merupakan proses perubahan kata pada dokumen sehingga memiliki bentuk yang seragam[6].

b) Filtering

Membuang kata yang tidak memiliki nilai atau arti seperti angka, url, kata hubung, tanda baca, kata ganti dan sebagainya merupakan proses utama dalam tahap Filtering[7].

c) Normalisasi Kata

Mengubah kata singkatan, gaul atau kata yang tidak sesuai dengan EYD pada Bahasa Indonesia merupakan proses dari Normalisasi[8].

d) Stopword Removal

Stopword merupakan sebuah kata yang memiliki frekuensi yang banyak atau sering muncul serta tidak terlalu penting untuk dokumen tersebut. Kata-kata yang tidak memiliki arti juga merupakan Stopword[9].

e) Stemming

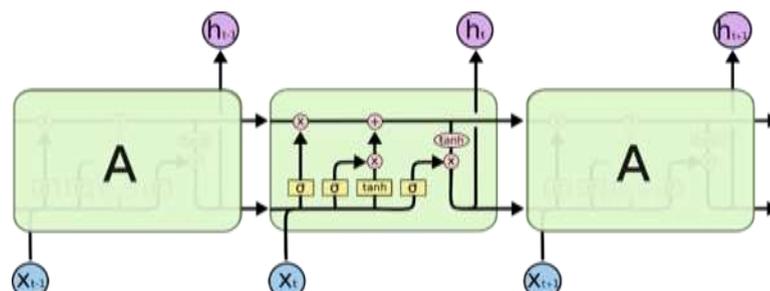
Stemming merupakan proses perubahan kata yang memiliki imbuhan menjadi kata dasar pada sebuah dokumen[9].

f) Tokenizing

Tokenisasi atau Tokenizing merupakan proses pembagian teks berupa paragraf atau kalimat menjadi beberapa bagian. Tokenisasi ini menggunakan delimiter atau pemisah pada setiap kata yaitu berupa spasi, enter, atau whitespace[8].

2.5 Long Short Term Memory

LSTM merupakan model yang lebih efisien dan memiliki keuntungan dibandingkan model RNN. LSTM dapat melakukan analisis walaupun terdapat jarak antar teks. Hal ini terjadi karena terdapat memori yang dapat menyimpan informasi dalam waktu lama[10]. Berikut merupakan arsitektur dalam LSTM: .



Gambar 1: Arsitektur metode LSTM

Gambar 1 tersebut merupakan arsitektur dari LSTM, pada bagian bawah arsitektur tersebut terdapat cell gates yang memiliki fungsi untuk meregulasi informasi untuk dilanjutkan ke cell state yang merupakan bagian atas yang berfungsi untuk meneruskan informasi ke jaringan berikutnya[11]. Pada jaringan LSTM terdapat beberapa gate yaitu forget gate, input gate, output gate dan update gate yang dihitung sebagai keluaran dari hidden layer pada jaringan selanjutnya.

- *Forget Gate*

Pada Layer ini terdapat dua masukan yaitu nilai output sebelumnya dan input saat ini yang digabungkan dengan fungsi aktivasi sigmoid. Pada tahap inilah informasi akan diteruskan atau dilupakan jika diteruskan informasi dilanjutkan ke cell state. Persamaan yang digunakan dalam tahapan ini :

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

Keterangan :

f_t	: forget gate	h_{t-1}	: nilai output sebelumnya
σ	: fungsi sigmoid	x_t	: input saat ini
W_f	: nilai bobot untuk forget gate	b_f	: nilai bias pada forget gate

- *Input Gate*

Pada *Input gate* ini output sebelumnya dan input saat ini digabungkan untuk melewati dua fungsi aktivasi yaitu dengan menggunakan aktivasi sigmoid untuk nilai input dan fungsi tanh untuk memperbaharui nilai cell state. Persamaan yang digunakan pada tahap ini adalah sebagai berikut:

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

$$\bar{C}_t = \tanh(WC \cdot [h_{t-1}, x_t] + bC) \quad (3)$$

Keterangan:

i_t	: input gate	h_{t-1}	: nilai output sebelumnya
σ	: fungsi sigmoid	x_t	: input saat ini
W_i	: nilai bobot untuk input gate	b_i	: nilai bias pada input gate
WC	: nilai bobot untuk cell state	\tanh	: fungsi tanh
\bar{C}_t	: nilai baru untuk cell state	bC	: nilai bias untuk cell state

- *Cell State*

pada tahap pembaharuan cell state, tahapan ini terjadi penggabungan antara dua nilai yaitu, nilai cell state sebelumnya dengan nilai yang didapat dari forget gate dan nilai input gate yang dikalikan dengan nilai terbaru cell state. Persamaan yang digunakan pada tahap ini adalah sebagai berikut:

$$C_t = f_t \times C_{t-1} + i_t \times \bar{C}_t \quad (4)$$

Keterangan:

C_t	: nilai baru cell state	\bar{C}_t	: nilai cell state sebelumnya
f_t	: nilai forget gate	i_t	: nilai input gate

- *Output Gate*

Pada tahapan ini, dihasilkan nilai output, yang bersumber dari gabungan nilai sebelumnya yang telah melewati fungsi aktivasi sigmoid dan hasil dari penggabungan tersebut dikalikan dengan fungsi tanh untuk menghasilkan output untuk hidden layer berikutnya. Persamaan yang digunakan dalam tahap ini adalah :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

Keterangan:

o_t	: nilai output gate	h_{t-1}	: nilai output sebelumnya
σ	: fungsi sigmoid	x_t	: input saat ini
W_o	: nilai bobot untuk output gate	b_o	: nilai bias pada output gate

Atau persamaan yang didapat untuk mendapatkan nilai output orde ke-t dapat dijabarkan pada persamaan berikut:

$$h_t = o_t \times \tanh(C_t) \quad (6)$$

Keterangan:

h_t	: nilai output gate ke-t	\tanh	: fungsi tanh
o_t	: nilai output gate		

2.6 Evaluasi Model

Pada setiap klasifikasi yang dilakukan setelah pelatihan data maka dilakukan evaluasi terhadap model tersebut yaitu dengan menilai himpunan data latih dengan himpunan data uji dan akan mendapatkan akurasi atau tingkat pengenalan. Confusion

matrix merupakan hasil dari klasifikasi biner yang didapat dari evaluasi model terhadap dataset dan dipresentasikan menjadi matrix 2x2 untuk mengukur kinerja yang didapat dari beberapa parameter[7].

Tabel 1. Confusion Matrix

		Kelas Prediksi	
		Positif	Negatif
Kelas Sesungguhnya	Positif	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
	Negatif	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Tabel 1 diatas menunjukkan beberapa variabel yang berperan untuk proses evaluasi model klasifikasi. Variabel *True Negative (TN)* merupakan data yang tepat diklasifikasi oleh sistem sebagai nilai negatif atau salah, kemudian variabel *True Positif (TP)* merupakan data yang tepat diklasifikasi sebagai nilai positif atau benar. Lalu untuk variabel *False Positive (FP)* merupakan data yang diklasifikasikan tidak tepat apabila keluaran berupa positif atau benar kemudian untuk variabel *False Negative (FN)* merupakan data yang diklasifikasikan dengan kurang tepat.

Rumus akurasi dapat dihitung sebagai berikut:

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

Rumus *Precision* dapat dihitung sebagai berikut:

$$\frac{TP}{TP + FN} \quad (8)$$

Rumus *Recall* dapat dihitung sebagai berikut:

$$\frac{TP}{TP + FP} \quad (9)$$

3 Hasil dan Pembahasan

3.1 Data

Pada penelitian ini data yang digunakan bersumber dari ulasan pada aplikasi PeduliLindungi yang ada pada Google Play Store. Pengumpulan data dilakukan dengan bantuan teknik scraping menggunakan Bahasa pemrograman Python dengan dukungan modul yaitu *google-play-scraper*. Data yang didapat dari hasil scraping berjumlah 3000 data dari bulan November 2021 sampai bulan Maret 2022 dengan mengambil ulasan yang relevan saja terhadap aplikasi PeduliLindungi.

3.2 Pelabelan Data

Setelah dilakukan pengumpulan data, dilanjutkan dengan proses pelabelan data. Pada data ini akan diberikan dua label sentimen yaitu sentimen positif dan sentimen negatif. Data akan dinilai oleh 3 annotator untuk mendapatkan persamaan persepsi pada kalimat ulasan sehingga didapati label mana yang cocok dengan ulasan tersebut.

Tabel 2. Contoh Pelabelan Data Ulasan

Data	Penilai 1	Penilai 2	Penilai 3	Hasil Akhir

Alhamdulillah dengan adanya aplikasi PeduliLindungi mudah mudahan bisa mempercepat menghilangnya wabah virus corona Aamiin karena selama wabah virus corona ini membuat tidak tenang apalagi rakyat kecil seperti kami	Positif	Positif	Positif	Positif
Aplikasi gak jelas perbaiki lah biar gampang orang akses daftar tanggal lahir nyarinya bikin emosi udah bener pas di klik gak sesuai kan bikin emosi udah vaksin 2x sertifikat vaksin gak ada daftarnya Tolong lah di perbaiki saya udah uninstal karena bener2 gak guna	Negatif	Negatif	Negatif	Negatif
Sertifikat vaksin sudah muncul Hanya saja kalau scan barcode di beberapa tempat suka susah	Positif	Positif	Negatif	Positif

Pelabelan yang dilakukan oleh 3 annotator akan didapatkan satu hasil akhir seperti pada Tabel 2. Hasil tersebut didapatkan berdasarkan banyaknya jumlah label yang diberikan pada ulasan tersebut, hasil akhir akan diberikan positif jika terdapat dua atau lebih penilai memberikan ulasan dengan label positif pada ulasan tersebut dan juga sebaliknya jika dua atau lebih penilai memberikan ulasan dengan label negatif maka hasil akhir ulasan tersebut akan berlabel negatif.

3.3 Preprocessing

Setelah melalui tahap pelabelan maka selanjutnya masuk ke tahap *Preprocessing* yang bertujuan untuk menghilangkan *noise* pada data sehingga data mudah diproses pada tahap klasifikasi. Untuk hasil dari *preprocessing* dapat dilihat pada Tabel 3 berikut.

Dok.	Sebelum Preprocessing	Sesudah Preprocessing
D1	Terimakasih admin, atas respon dan perbaikannya sertifikatnya sudah bisa saya klaim yang mana sebelumnya tidak muncul!!	['terima', 'kasih', 'admin', 'respon', 'baik', 'sertifikat', 'klaim', 'muncul']
D2	Knapa ya aplikasi pedulilindungi saya gak bisa lihat sertifikat vaksin gak bisa di buka ? Mohon bantuannya!!	['iya', 'aplikasi', 'pedulilindungi', 'lihat', 'sertifikat', 'vaksin', 'buka', 'mohon', 'bantu']
D3	Sangat membantu masyarakat dalam mendapatkan info dan sertifikat vaksin yang sangat dibutuhkan masyarakat Indonesia, sistem kerjanya cepat Terima kasih atas kerja keras pihak terkait TUHAN selalu berkahi pengabdianya untuk bangsa dan negara kita Indonesia tercinta.	['bantu', 'masyarakat', 'info', 'sertifikat', 'vaksin', 'butuh', 'masyarakat', 'indonesia', 'sistem', 'kerja', 'cepat', 'terima', 'kasih', 'kerja', 'keras', 'kait', 'tuhan', 'berkah', 'abdi', 'bangsa', 'negara', 'indonesia', 'cinta']

Pada Tabel 3 tersebut data melalui beberapa tahapan pada preprocessing yaitu tahapan *Case Folding*, *Filtering*, *Normalisasi Kata*, *Stopword Removal*, *Stemming*, dan *Tokenizing*.

3.4 Ekstrasi Fitur

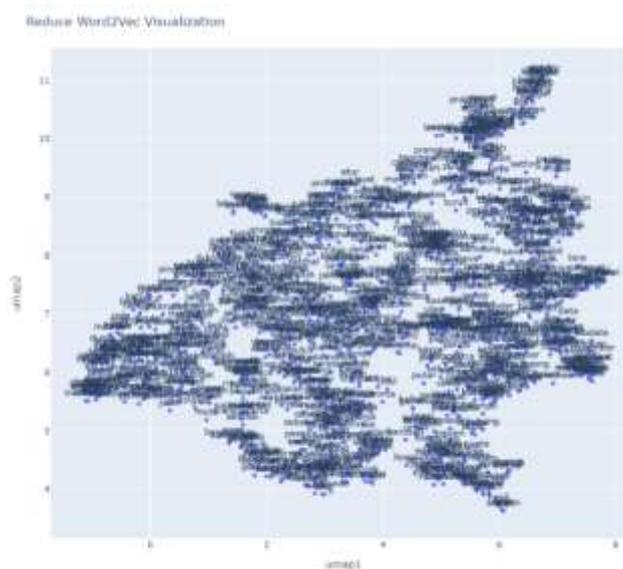
Setelah tahap *preprocessing* maka dilakukan tahap ekstraksi fitur dengan melakukan perubahan kata pada data yang sudah bersih menjadi bentuk vektor. Hasil vektor yang ada merupakan representasi kedekatan antar kata pada data. Penggunaan vektor pada data ini menggunakan model word2vec dengan metode Skip-gram, penggunaan metode ini digunakan karena memiliki kelebihan dalam mendeteksi kata dasar yang jarang muncul.

Pada penelitian ini akan menggunakan ukuran dimensi sebesar 100 dimensi, sehingga akan terbentuk representasi kata dengan ukuran $n \times 100$ dimensi. Berikut merupakan salah satu representasi vektor yang berasal dari kata “aplikasi” setelah dilakukannya perubahan kata dengan skip-gram berdimensi 100.

```
w2v_model.wv["aplikasi"]
array([-0.07806958, -0.22496965, -0.25242248,  0.25362784,  0.18444236,
        0.03259543, -0.36962885,  0.12715591, -0.04807323,  0.05418178,
        0.14624617, -0.08666758,  0.06889473, -0.26185774, -0.02838952,
        0.21746318, -0.85599523,  0.20607466, -0.08291369,  0.13181873,
        -0.28305745, -0.1337068 , -0.13272756, -0.13073558,  0.21228889,
        -0.14728888,  0.18178331, -0.02390883,  0.07452788, -0.18491187,
        0.15188714,  0.27581757, -0.26284666, -0.21232804,  0.25714156,
        0.42825103,  0.02299856,  0.2611432 , -0.09905295,  0.09294281,
        0.17490761,  0.09107136,  0.00382294, -0.18493147, -0.13082215,
        0.02830287,  0.1415286 , -0.03859519,  0.06651187,  0.06981702,
        0.22784883,  0.35917923,  0.28954785,  0.09028599, -0.38388828,
        -0.06106532, -0.0840192 , -0.11251891,  0.1532189 ,  0.36623864,
        0.06746758, -0.21224281,  0.29370532, -0.27955647, -0.44472197,
        -0.11731679, -0.05139199,  0.16112599,  0.1883317 ,  0.18661447,
        0.17341281,  0.11839583,  0.04279615, -0.18695915, -0.19611476,
        0.11687483,  0.01824752, -0.38913057,  0.16883945, -0.1387167 ,
        0.06896681,  0.1885607 , -0.22073366,  0.24545692,  0.5715955 ,
        -0.32886797,  0.19449165, -0.18822906,  0.06551002, -0.16365827,
        0.09359469, -0.09472021, -0.28013941, -0.15386277, -0.15417995,
        0.42298004, -0.23295064, -0.11672648,  0.18725986,  0.1179633 ]),
      dtype=float32)
```

Gambar 2: Vektor kata dari “aplikasi”

Berdasarkan Gambar 2 diatas kata aplikasi memiliki berbagai macam vektor dengan ukuran 100 dimensi. Setelah dilakukannya word vector maka setiap kata yang berbentuk vektor akan direpresentasikan ke dalam sebuah vector space. Gambar berikut merupakan gambar vector space yang didapatkan dari word vector dataset ulasan PeduliLindungi dengan ukuran 100 dimensi yang digambarkan ke dalam bentuk 2 dimensi seperti gambar 3.



Gambar 3: Vektor space

Pada Gambar 3 tersebut merupakan bentuk dari vector space yang merupakan gambaran kedekatan antar masing-masing kata setelah dilakukannya ekstraksi fitur dengan menggunakan skip-gram.

3.5 Klasifikasi Model LSTM

Untuk mendapatkan nilai yang terbaik dari model yang dibentuk dapat dihasilkan dari beberapa parameter. Beberapa parameter yang digunakan untuk mendapatkan nilai yang optimal dalam menentukan model yaitu melakukan pengujian

berdasarkan jumlah Unit, pengujian berdasarkan Epoch, serta pengujian berdasarkan Batch. Berikut merupakan hasil dari pengujian parameter:

3.5.1 Pengujian Parameter Jumlah Units

Pada pengujian ini dilakukan dengan beberapa percobaan berdasarkan jumlah units lapisan pada model LSTM. Pengujian dilakukan dengan menggunakan unit yang berjumlah 100, 200, 300, 400, dan 500. Untuk inisiasi epoch dan batch yang digunakan adalah 20 dan 32..

Tabel 1: Hasil Pengujian Parameter Jumlah Unit

Unit	Epoch	Batch	Test Akurasi	Test Loss
50	20	32	78,56%	21,44%
100	20	32	82,44%	17,56%
200	20	32	81%	19%
300	20	32	80%	20%
400	20	32	79,77%	20,23%
500	20	32	80,77%	19,23%

Pada Tabel 1 jumlah unit sebesar 100 memiliki hasil akurasi sebesar 82,44% dengan epoch sebesar 20 dan batch sebesar 32. Hasil akurasi dari pengujian jumlah units ini memiliki nilai yang dapat berubah-ubah oleh karena itu, nilai yang didapatkan berdasarkan trial and error.

3.5.2 Pengujian Parameter Epoch

Pada pengujian ini dataset akan dilakukan training sebanyak Epoch yang diinginkan. Epoch merupakan banyaknya algoritma dalam mempelajari dan bekerja dengan menggunakan dataset. Pengujian ini akan menggunakan beberapa nilai epoch berjumlah 10, 20, 30, 40, dan 50. Jumlah Unit yang digunakan berasal dari pengujian sebelumnya yang memiliki akurasi terbaik yaitu 100.

Tabel 2: Hasil Pengujian Parameter Epoch

Unit	Epoch	Batch	Test Akurasi	Test Loss
100	10	32	81,11%	18,89%
100	20	32	82,44%	17,56%
100	30	32	81,77%	18,23%
100	40	32	80,55%	19,45%
100	50	32	77,77%	22,23%

Pada Tabel 2 tersebut jumlah epoch 20 memiliki akurasi tertinggi yaitu 82,44% dan epoch dengan jumlah 50 memiliki akurasi yang terendah yaitu 77,77%..

3.5.3 Pengujian Parameter Batch

Pengujian pada parameter Batch ini dilakukan berdasarkan jumlah batch size. Batch merupakan total dari sebaran dari data training yang akan proses oleh model LSTM. Pengujian ini dilakukan dengan menggunakan jumlah batch sebesar 16, 32, 48, 64, dan 80. Untuk jumlah unit dan epoch yang menggunakan jumlah yang memiliki nilai akurasi terbaik sebelumnya yaitu 100 units dan 20 epoch.

Tabel 3: Hasil Pengujian Parameter Batch

Unit	Epoch	Batch	Test Akurasi	Test Loss
------	-------	-------	--------------	-----------

100	20	16	82,11%	17,89%
100	20	32	82,44%	17,56%
100	20	48	81,11%	18,89%
100	20	64	81,55%	18,45%
100	20	80	81,88%	18,12%

Pada Tabel 3 tersebut nilai akurasi tertinggi didapatkan dari menggunakan batch sejumlah 32 dengan nilai 82,44% dan untuk nilai terendah yaitu pada batch sejumlah 48.

3.6 Evaluasi Hasil

Setelah melakukan klasifikasi dengan beberapa pengujian dilakukan evaluasi hasil dengan melakukan pengujian dengan data test. Pada data test terdapat 900 data yang terbagi 478 data ulasan positif dan 422 data ulasan negatif. Hasil evaluasi ini diukur dengan menggunakan Confusion matrix sebagai ukuran kinerja klasifikasi. Hasil dari evaluasi dapat dilihat pada tabel sebagai berikut:

Tabel 4: Hasil Evaluasi *Confusion Matrix*

		Kelas Sebenarnya	
		Positif	Negatif
Kelas Prediksi	Positif	376 (TP)	56 (FP)
	Negatif	102 (FN)	366 (TN)

Oleh karena itu, performa model LSTM dapat dihitung dengan menggunakan persamaan (7), (8), (9) sebagai berikut:

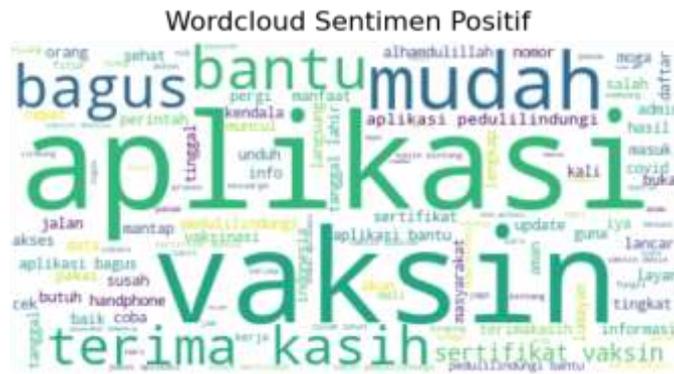
$$\begin{aligned}
 \text{Akurasi} &= \frac{TP + TN}{TP + TN + FP + FN} = \frac{376 + 366}{376 + 366 + 56 + 102} = 82,44\% \\
 \text{Precision} &= \frac{TP}{TP + FN} = \frac{376}{376 + 102} = 78,66\% \\
 \text{Recall} &= \frac{TP}{TP + FP} = \frac{376}{376 + 56} = 87,03\%
 \end{aligned}$$

Maka performa dari model LSTM dalam mengklasifikasikan label positif dan label negatif memiliki nilai akurasi sebesar 82,44% dengan 376 data label positif dan 366 data label negatif dari 900 data ulasan berhasil diklasifikasikan dengan tepat oleh model ini. Setelah itu terdapat nilai Precision sebesar 78,66% dan Recall sebesar 87,03%.

3.6 Analisis Hasil Klasifikasi

Setelah melakukan klasifikasi dengan beberapa pengujian dilakukan evaluasi hasil dengan melakukan pengujian dengan data test. Pada data test terdapat 900 data yang terbagi 478 data ulasan positif dan 422 data ulasan negatif. Hasil evaluasi ini diukur dengan menggunakan Confusion matrix sebagai ukuran kinerja klasifikasi. Hasil dari evaluasi yang didapat adalah bahwa nilai akurasi sebesar 82,44% menunjukkan bahwa model LSTM berhasil memprediksi data uji dengan benar sesuai dengan kelas sebenarnya sebesar 82,44% dengan 376 label Positif dan 366 label negatif berhasil diprediksi benar oleh model. Nilai Precision sebesar 78,66% menunjukkan ketepatan model dalam mengklasifikasi data kelas label cukup baik karena dari 376 label positif yang berhasil di klasifikasi dengan benar terdapat 102 label positif yang salah prediksi yang seharusnya label positif dan nilai Recall sebesar 87,03% menunjukkan bahwa model LSTM dapat memprediksi nilai positif dengan benar sebesar 376 label positif dari keseluruhan data yang berlabel positif pada data uji.

Berikut merupakan hasil analisis dengan representasi dari wordcloud yang berasal dari ulasan yang mengandung sentimen positif dan negatif:



Gambar 4: Wordcloud Sentimen Positif

Pada Gambar 4 diatas hasil dari wordcloud pada data ulasan aplikasi PeduliLindungi menunjukkan bahwa kata kunci yang sering muncul didalam data ulasan adalah “aplikasi”, “vaksin”, “mudah”, “bagus”, “bantu”, ”terima”, dan “kasih”. Hal ini menunjukkan bahwa didalam data ulasan aplikasi PeduliLindungi pada sentimen positif kata-kata tersebut sering disebut oleh para pengguna. Kata-kata tersebut menunjukan bahwa pengguna aplikasi PeduliLindungi ini mudah dalam penggunaannya dan dapat membantu dalam mengetahui pendaftaran vaksin dengan menggunakan aplikasi ini, serta memuji aplikasi PeduliLindungi ini merupakan aplikasi yang bagus..



Gambar 5: Wordcloud Sentimen Negatif

Pada Gambar 5 diatas hasil wordcloud dari sentimen negatif menunjukkan bahwa terdapat beberapa kata kunci yang muncul dalam ulasan sentimen negatif terhadap aplikasi PeduliLindungi seperti “aplikasi”, “sertifikat”, “vaksin”, “tanggal lahir”, “susah”, “muncul”, dan “klaim”. Dari kata-kata tersebut menunjukkan bahwa pengguna aplikasi PeduliLindungi ini mengkritik mengenai sertifikat vaksin yang susah untuk keluar di dalam aplikasi PeduliLindungi ini serta susahnya menginput data tanggal lahir di dalam aplikasi ini.

4 Kesimpulan dan Saran

4.1 Kesimpulan

Data ulasan bersumber dari Google Play Store dengan menggunakan teknik web scrapping mengenai ulasan aplikasi PeduliLindungi serta berjumlah 3000 data. Data tersebut telah dilakukan praproses data seperti case folding, filtering, normalisasi kata, stopwords removal, stemming, dan Tokenisasi. Setelah dilakukannya praproses data tahap selanjutnya data tersebut akan dilakukan ekstraksi fitur dengan mengubah data teks tersebut menjadi vektor dengan menggunakan metode word2vec. Hasil tersebut selanjutnya akan dilakukan pelatihan data dengan menggunakan model long short term memory dengan menggunakan perbandingan 70% data training dan 30% data testing. Pada proses pengujian dilakukan dengan melakukan pengujian beberapa hyperparameter seperti pengujian dengan jumlah units, pengujian dengan beberapa epoch, dan pengujian dengan jumlah batch. Hasil dari pada proses tersebut didapatkan konfigurasi hyperparameter terbaik dengan jumlah unis sebesar 100, epoch sebesar 20, serta dengan menggunakan 32 batch dari konfigurasi parameter tersebut model LSTM dapat menghasilkan akurasi sebesar 82,44% kemudian mendapatkan nilai precision sebesar 78,66%, serta recall sebesar 87,03%.

4.2 Saran

Berdasarkan hasil analisis dan penelitian yang telah dilakukan berikut merupakan beberapa saran yang dapat digunakan untuk pengembangan penelitian lebih lanjut:

- a) Diharapkan untuk penelitian selanjutnya dapat menggunakan jasa para ahli untuk melabelkan data serta dapat menggunakan lebih dari 2 label.
- b) Harapan untuk penelitian selanjutnya, dapat menggunakan lebih banyak konfigurasi hyperparameter untuk mendapatkan model yang lebih baik lagi.
- c) Dalam melakukan praposes pada data diharapkan untuk penelitian berikutnya lebih memperhatikan tahap dalam praproses data sehingga data akan lebih bersih dan dapat meningkatkan performa model ini.
- d) Diharapkan untuk penelitian selanjutnya dapat membandingkan model long short term memory dengan model algoritma lainnya dalam mengklasifikasi data ulasan sentimen mungkin dengan menggunakan algoritma supervised learning atau deep learning lainnya.

5 Referensi

- [1] KEMENTERIAN KOMUNIKASI DAN INFORMATIKA RI, “Keputusan Menteri KOMINFO. Diambil kembali dari JDIIH Kementerian Komunikasi dan Informatika RI,” Nov. 2021. <https://jdih.kominfo.go.id/> (accessed Nov. 10, 2021).
- [2] C. Eka Putri and dan Radja Erland Hamzah, “APLIKASI PEDULILINDUNGI MITIGASI BENCANA COVID-19 DI INDONESIA,” vol. 4, no. 1, p. 66, 2021.
- [3] F. A. Nugraha, *Analisis Sentimen Terhadap Pembatasan Sosial Menggunakan Deep Learning*, 1st ed. Bandung: Kreatif Industri Nusantara, 2020.
- [4] Y. Sahria, “Implementasi Teknik Web Scraping pada Jurnal SINTA,” pp. 297–306, 2020, [Online]. Available: <http://sinta2.ristekdikti.go.id/journals/detail>
- [5] D. Joko Haryanto, L. Muflikhah, and M. Ali Fauzi, “Analisis Sentimen Review Barang Berbahasa Indonesia Dengan Metode Support Vector Machine Dan Query Expansion,” 2018. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [6] A. Faadilah, “ANALISIS SENTIMEN PADA ULASAN APLIKASI TOKOPEDIA DI GOOGLE PLAY STORE MENGGUNAKAN METODE LONG SHORT TERM MEMORY,” 2020.
- [7] O. : Rara and A. Puspita, “Analisis Sentimen Terhadap Review E-Commerce Pada Google Play Store Menggunakan Metode Naïve Bayes Classifier (NBC) dengan Seleksi Fitur Information Gain (IG),” 2021, [Online]. Available: <http://repository.unimus.ac.id>
- [8] I. Najiyah and I. Haryanti, “SENTIMEN ANALISIS COVID-19 DENGAN METODE PROBABILISTIC NEURAL NETWORK DAN TF-IDF,” *JURNAL RESPONSIF*, vol. 3, no. 1, pp. 100–111, 2021, [Online]. Available: <http://ejournal.ars.ac.id/index.php/jti>
- [9] A. Erfina, S. Basryah, A. Saepulrohman, and D. Lestari, “MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM),” *Seminar Nasional Informatika*, vol. 2020, 2020.
- [10] J. Nurvania and K. Muslim Lhaksamana, “Analisis Sentimen Pada Ulasan di TripAdvisor Menggunakan Metode Long Short-Term Memory (LSTM),” vol. 8, p. 4121, Jun. 2021.
- [11] D. Junggu Manggala Pasaribu, Kusri, and Sudarmawan, “PENINGKATAN AKURASI KLASIFIKASI SENTIMEN ULASAN MAKANAN AMAZON DENGAN BIDIRECTIONAL LSTM DAN BERT EMBEDDING,” *Jurnal Teknologi Informasi dan Komunikasi*, vol. 10, pp. 9–20, 2020.