



**PERBANDINGAN METODE *DEEP LEARNING* DALAM ANALISA
SENTIMEN ULASAN APLIKASI JAKARTA KINI (JAKI)**

TUGAS AKHIR

Tengku Iqbal Nugraha

41518010147

UNIVERSITAS
MERCU BUANA

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MERCU BUANA

JAKARTA

2022



**PERBANDINGAN METODE *DEEP LEARNING* DALAM ANALISA
SENTIMEN ULASAN APLIKASI JAKARTA KINI (JAKI)**

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:
Tengku Iqbal Nugraha
41518010147

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA

2022

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM : 41518010147

Nama : Tengku Iqbal Nugraha

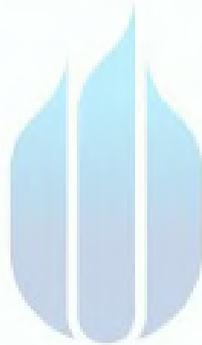
Judul Tugas Akhir : Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 07 Juli 2022



Tengku Iqbal Nugraha



UNIVERSITAS
MERCU BUANA

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Tengku Iqbal Nugraha
NIM : 41518010147
Judul Tugas Akhir : Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 07 Juli 2022

UNIVERSITAS
MERCU BUANA



Tengku Iqbal Nugraha

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Tengku Iqbal Nugraha
 NIM : 41518010147
 Judul Tugas Akhir : Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)

Menyatakan bahwa :

1. Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis		Status	
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi		Diajukan	√
		Jurnal Nasional Terakreditasi	√		
		Jurnal International Tidak Bereputasi		Diterima	
		Jurnal International Bereputasi			
Disubmit/dipublikasikan di :	Nama Jurnal	: Jurnal Informatika: Jurnal Pengembangan IT			
	ISSN	: 2477-5126			
	Link Jurnal	: ejournal.poltektegal.ac.id/index.php/informatika			
	Link File Jurnal Jika Sudah di Publish	:			

2. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.
3. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 27 Juli 2022



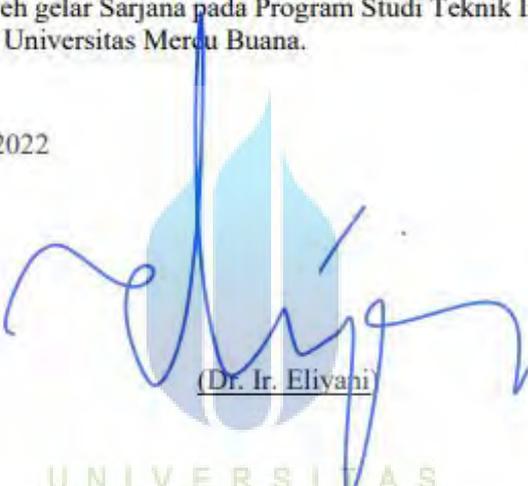
Tengku Iqbal Nugraha

LEMBAR PERSETUJUAN PENGUJI

NIM : 41518010147
Nama : Tengku Iqbal Nugraha
Judul Tugas Akhir : Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 27 Juli 2022



(Dr. Ir. Elivani)

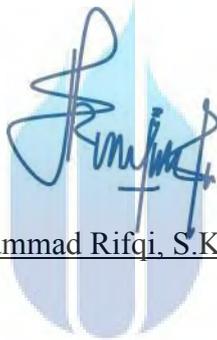
UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI

NIM : 41518010147
Nama : Tengku Iqbal Nugraha
Judul Tugas Akhir : Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 27 Juli 2022



(Muhammad Rifqi, S.Kom, M.Kom)

UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI

NIM : 41518010147
Nama : Tengku Iqbal Nugraha
Judul Tugas Akhir : Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 27 Juli 2022



UNIVERSITAS
MERCU BUANA

LEMBAR PENGESAHAN

NIM : 41518010147
Nama : Tengku Iqbal Nugraha
Judul Tugas Akhir : Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 27 Juli 2022

a. h Menyetujui,



Dr. Leonard Goeirmanto, ST., M.Sc
Dosen Pembimbing

Mengetahui,



(Wawan Gunawan, S.Kom, MT)
Koord. Tugas Akhir Teknik Informatika



(Ir. Emil R. Kaburuan, Ph.D., IPM.)
Ka. Prodi Teknik Informatika

KATA PENGANTAR

Puji syukur kita panjatkan kepada ALLAH SWT yang telah melimpahkan Rahmat dan Karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)” dengan baik dan dalam jangka waktu yang telah ditentukan. Tugas akhir ini disusun untuk memenuhi syarat untuk LULUS sebagai sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Mercu Buana. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, Tugas Akhir bisa saja tidak akan terselesaikan dengan baik. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Leonard Goeirmanto, ST., M.Sc selaku dosen pembimbing Tugas Akhir yang telah meluangkan waktunya untuk memberikan bimbingan serta arahan dalam penyusunan Tugas Akhir ini.
2. Ibu Saruni Dwiasnati, ST, MM, M.Kom selaku dosen pembimbing akademik yang telah membimbing saya sejak awal mulai semester hingga sampai saat ini dan selalu memberikan motivasi serta arahan agar dapat lulus tepat waktu.
3. Bapak Emil R. Kaburuan, Ph.D. selaku Kepala Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Mercu Buana.
4. Bapak Wawan Gunawan, S.Kom, MT selaku Koordinator Tugas Akhir Jurusan Teknik Informatika, Universitas Mercu Buana.
5. Keluarga dan orang tua yang selalu memberikan dukungan dan selalu mendoakan yang terbaik.
6. Teman-teman dan sahabat yang selalu memberikan dukungan dan memotivasi dalam melakukan penulisan tugas akhir ini agar dapat terselesaikan dengan baik.

Akhir kata, penulis berharap tugas akhir ini dapat menjadi bermanfaat bagi pembaca dan menambah wawasan pengetahuan semua pihak.

Jakarta, 27 Juli 2022

Penulis

DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL	i
LEMBAR PERNYATAAN ORISINALITAS.....	ii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR...	iii
SURAT PERNYATAAN LUARAN TUGAS AKHIR.....	iv
LEMBAR PERSETUJUAN PENGUJI	v
LEMBAR PENGESAHAN	viii
ABSTRAK	ix
ABSTRACT	x
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xii
NASKAH JURNAL	1
KERTAS KERJA.....	10
BAB 1. LITERATUR REVIEW.....	11
BAB 2. ANALISA DAN PERANCANGAN	18
BAB 3. <i>SOURCE CODE</i>	22
BAB 4. DATASET	33
BAB 5. TAHAPAN EKSPERIMEN.....	35
BAB 6. HASIL SEMUA EKSPERIMEN.....	46
DAFTAR PUSTAKA	55
LAMPIRAN DOKUMEN HAKI.....	58
LAMPIRAN KORESPONDENSI.....	59

Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)

Tengku Iqbal Nugraha^{1*)}, Leonard Goeirmanto²

^{1,2}Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana, Jakarta

^{1,2}Jln. Raya Meruya Selatan, Kec. Kembangan, Jakarta Barat, 11650, Indonesia

email: ¹41518010147@student.mercubuana.ac.id, ²leonard@mercubuana.ac.id

Abstract – The Jakarta Kini (JAKI) application is an application from the DKI Jakarta provincial government that is used to make it easier for the public to find official information and various public services in Jakarta. Since the JAKI application was first released on the play store and app store, reviews of the JAKI application already have data reviews from users. Reviews given by users vary widely from positive to negative. This review can be used as an evaluation material for the developer to find out what needs to be improved or what needs to be maintained. Therefore, to obtain information and to be able to conclude the polarity review of the JAKI application, sentiment analysis is needed using the application of data mining. Sentiment analysis was carried out to determine the polarity of the JAKI application review and determine the performance of the deep learning algorithm in classifying text. The method used in analyzing sentiment is the Long Short Term Memory (LSTM) algorithm and will be compared with the Convolutional Neural Network (CNN) algorithm and the Artificial Neural Network (ANN) algorithm. The results show that the LSTM algorithm has a higher accuracy rate with an average 10-fold accuracy rate of 94.78%, then the second is the CNN algorithm with an average 10-fold accuracy rate of 92.98%, and the last one is ANN with an average accuracy rate of 10 fold of 92.66%.

Abstrak – Aplikasi Jakarta Kini (JAKI) merupakan aplikasi dari pemerintah provinsi DKI Jakarta yang digunakan untuk mempermudah masyarakat dalam mencari informasi resmi dan berbagai layanan masyarakat di Jakarta. Sejak aplikasi JAKI pertama kali rilis di *play store* dan *app store*, ulasan terhadap aplikasi JAKI sudah memiliki ribuan data ulasan dari pengguna. Ulasan yang diberikan dari pengguna sangat beragam dari yang positif hingga negatif. Ulasan tersebut bisa menjadi bahan evaluasi bagi pihak pengembang untuk dapat mengetahui apa saja yang perlu di tingkatkan maupun yang perlu di pertahankan. Oleh karena itu untuk mendapatkan informasi dan dapat menarik kesimpulan yang didapatkan dari polaritas ulasan aplikasi JAKI maka diperlukannya analisa sentimen dengan menggunakan penerapan data *mining*. Analisa sentimen dilakukan agar dapat mengetahui polaritas ulasan aplikasi JAKI dan mengetahui kinerja dari sebuah algoritma *deep learning* dalam melakukan klasifikasi teks. Metode yang digunakan dalam analisa sentimen adalah dengan algoritma Long Short Term Memory (LSTM) dan akan dikomparasikan dengan algoritma Convolutional Neural Network (CNN) dan algoritma Artificial Neural Network (ANN). Hasil menunjukkan bahwa algoritma LSTM memiliki tingkat akurasi yang lebih

tinggi dengan rata-rata tingkat akurasi dari 10 *fold* sebesar 94,78%, kemudian yang kedua adalah algoritma CNN dengan rata-rata tingkat akurasi dari 10 *fold* sebesar 92,98%, dan terakhir adalah algoritma ANN dengan rata-rata tingkat akurasi dari 10 *fold* sebesar 92,66%.

Kata Kunci – Analisa sentimen, *deep learning*, data mining, JAKI, klasifikasi.

I. PENDAHULUAN

Sentimen merupakan suatu pandangan atau pendapat emosional yang berlebihan terhadap sesuatu yang dapat mempengaruhi berbagai aspek. Untuk mengetahui penilaian terhadap pandangan masyarakat maka perlu di lakukannya analisa sentimen. Analisa sentimen adalah suatu teknik *natural language processing* (NLP) dengan mengekstrak dan mengidentifikasi pendapat, penilaian, dan emosi seseorang yang diungkapkan dalam bentuk teks [1]. Analisa sentimen banyak digunakan oleh para pelaku bisnis untuk mencari kebutuhan dan keinginan pelanggan serta analisa sentimen dapat menjadi tolak ukur untuk melihat polaritas terhadap sesuatu di masyarakat [2]. Dengan perkembangan teknologi yang pesat, data sentimen masyarakat dapat dengan mudah di dapatkan di internet. Salah satunya adalah melalui aplikasi twitter, setiap hari jutaan orang menggunakan twitter sehingga banyak data sentimen *tweet* yang bisa di dapatkan. Selain twitter data sentimen masyarakat juga bisa di dapatkan melalui ulasan pengguna di sebuah *marketplace* aplikasi *play store* dan *app store*. Data ulasan aplikasi di *play store* dan *app store* bisa menjadi bahan evaluasi terhadap kinerja sebuah aplikasi, sehingga bisa dapat mengetahui apa saja yang perlu di tingkatkan maupun yang perlu di pertahankan oleh pihak pengembang. Banyaknya data sentimen yang perlu di analisa maka di butuhkan peran dari data *mining*. Data *mining* adalah metode untuk menemukan pengetahuan dalam *database* agar dapat menemukan informasi yang berguna dari sebuah data [3].

Jakarta Kini (JAKI) adalah aplikasi yang berisi beragam informasi resmi dan berbagai layanan masyarakat dari pemerintah provinsi DKI Jakarta. Aplikasi JAKI dirancang agar dapat mengubah kota Jakarta menjadi kota metropolitan yang cerdas dan lebih baik dengan mewujudkan kota pintar yang efektif dan efisien [4]. Sejak dari aplikasi JAKI pertama kali rilis, ulasan terhadap aplikasi JAKI di *play store* maupun *app store* sudah memiliki ribuan data ulasan dari pengguna. Ulasan yang diberikan dari pengguna terdiri dari berbagai macam pendapat dari yang positif hingga negatif. Ulasan dari pengguna aplikasi JAKI tersebut bisa menjadi evaluasi bagi pihak pengembang untuk menentukan fitur mana yang harus dipertahankan atau yang perlu dilakukannya peningkatan.

*) penulis korespondensi: Tengku Iqbal Nugraha

Email: 41518010147@student.mercubuana.ac.id

Untuk mendapatkan informasi dan dapat menarik kesimpulan yang didapatkan dari polaritas ulasan aplikasi JAKI maka diperlukannya analisa sentimen dengan menggunakan penerapan data *mining*. Hasil analisa sentimen tersebut bisa menjadi tolak ukur kinerja layanan yang terdapat di aplikasi JAKI dan dapat mengetahui kinerja dari sebuah algoritma *deep learning* dalam melakukan klasifikasi teks.

Berdasarkan permasalahan di atas, penelitian ini mengusulkan tentang analisa sentimen ulasan pengguna aplikasi JAKI di *play store* dan *app store* dengan menggunakan algoritma *deep learning*. Analisa sentimen dilakukan dengan mengklasifikasi polaritas ulasan ke dalam kategori positif dan negatif. Data sentimen yang digunakan adalah data ulasan pengguna aplikasi JAKI yang tersedia di *play store* dan *app store* dalam rentang waktu 1 Januari 2021 hingga 5 Mei 2022. Penelitian ini menggunakan algoritma LSTM, CNN, dan ANN yang kemudian di komparasi untuk menentukan algoritma terbaik dalam analisa sentimen ulasan pengguna aplikasi JAKI.

II. PENELITIAN YANG TERKAIT

Analisa sentimen merupakan teknik *natural language processing* yang digunakan untuk melihat polaritas opini masyarakat dengan cara mengekstrak dan mengidentifikasi pendapat, penilaian, dan emosi seseorang yang diungkapkan dalam bentuk teks [1] [2]. Dalam penerapannya analisa sentimen sudah banyak di implementasikan. Seperti penelitian sebelumnya yang dilakukan oleh Cassey K. N. Papatung dan Agustinus Jacobus [5] melakukan analisa sentimen di twitter mengenai 3 topik yaitu tokoh, pariwisata, dan produk dengan menggunakan algoritma *Long-Short Term Memory*. Hasil pengujian yang telah dilakukan dalam melakukan klasifikasi sentimen tweet, dapat disimpulkan bahwa algoritma *Long-Short Term Memory* dapat melakukan klasifikasi sentimen twitter dengan tingkat akurasi 77%.

Kemudian pada penelitian yang dilakukan oleh Adyan Marendra Ramadhani dan Hong Soon Goo [6] melakukan analisa sentimen twitter dengan menggunakan metode *deep learning*. Penelitian tersebut melakukan eksperimen dengan menggunakan total 4000 *dataset*. Proses *training* dilakukan dengan 100 *epochs* dan *learning rate* sebesar 0.1 dan 0.001. Hasil menunjukkan bahwa algoritma *deep neural network* memiliki tingkat akurasi lebih tinggi dengan akurasi 77,45 % untuk data latih dan 75,03 % data uji, sedangkan untuk algoritma *multi layer perceptron* memiliki tingkat akurasi 67,45% untuk data latih dan 52,6% untuk data uji.

Kemudian pada penilitan yang dilakukan oleh Naila Aslam, Waheed Yousuf Ramay, Kewen Xia, dan Nadeem Sarwar [7] melakukan analisa sentimen ulasan aplikasi dengan algoritma *convolutional neural network*. Hasil menunjukkan bahwa dari metode yang diusulkan dalam penelitian tersebut mampu meningkatkan rata rata *precision* dari 75,72% menjadi 95,49%, rata-rata *recall* dari 69,40% menjadi 93,94%, dan *f-measure* dari 72,41% menjadi 94,71%.

III. METODE PENELITIAN

Dalam melakukan analisa sentimen ada beberapa tahapan yang dilakukan agar mendapatkan hasil model yang baik. Tahapan yang dilakukan meliputi pengumpulan data, *text preprocessing*, pembagian data uji dan latih, validasi dengan

k-fold cross validation dan evaluasi terhadap model dengan menggunakan *confusion matrix*. Keseluruhan tahapan analisa sentimen dapat dilihat pada Gbr. 1

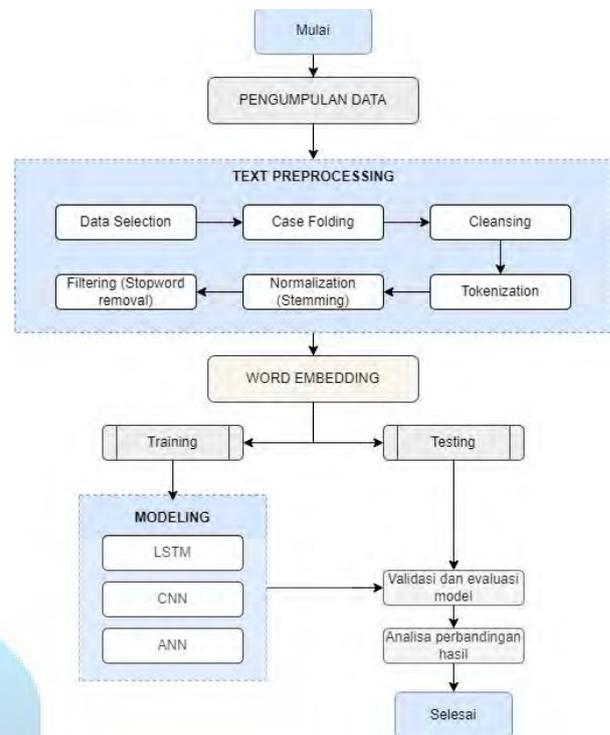


Fig. 1. Alur diagram analisa sentimen.

A. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data ulasan aplikasi JAKI yang terdapat di *play store* dan *app store*. Data didapatkan dengan melakukan *scrapping* menggunakan *library google play scrapper* dan *app store scrapper* dengan rentang waktu dari tanggal 1 Januari 2021 hingga 5 Mei 2022, sehingga data yang berhasil dikumpulkan yaitu sebanyak 2403 data ulasan. Kemudian data tersebut diseleksi dan *cleaning* kembali dengan menghapus beberapa ulasan yang tidak memiliki komentar, sehingga data menjadi 1980 data ulasan. Setelah data dikumpulkan selanjutnya dilakukan proses *labeling* pada setiap ulasan yang dilakukan secara manual dan dengan menambahkan 1 atribut yaitu sentimen. Sehingga data yang telah dikumpulkan memiliki 10 atribut yaitu *reviewId*, *userName*, *userImage*, *review*, *rating*, *date*, *replyReview*, *repliedAt*, *Source*, dan Sentimen. Proses *labeling* dilakukan dengan menilai ulasan pengguna kedalam kategori positif dan negatif. Penilaian ulasan tersebut tidak didasarkan pada *rating* yang diberikan oleh pengguna akan tetapi berdasarkan ulasan yang diberikan, hal ini dilakukan karena ada beberapa *rating* yang diberikan tinggi akan tetapi ulasan yang diberikan bernilai negatif dan juga sebaliknya. Setelah dilakukan proses *labeling* data memiliki perbandingan 991 ulasan negatif dan 989 ulasan positif.

TABEL I
DESKRIPSI ATRIBUT DATASET

Atribut	Dataset
<i>reviewId</i>	Kode unik untuk setiap ulasan dari user.

Atribut	Dataset
<i>userName</i>	<i>Username</i> dari pengguna yang memberikan ulasan.
<i>userImage</i>	Image dari pengguna yang memberikan ulasan.
<i>review</i>	Ulasan pengguna terhadap aplikasi.
<i>rating</i>	Nilai ulasan yang diberikan pengguna.
<i>date</i>	Tanggal atau waktu pengguna ketika memberikan ulasan.
<i>replyReview</i>	Ulasan balasan terhadap ulasan dari pengguna.
<i>repliedAt</i>	Tanggal atau waktu balasan ulasan.
<i>Source</i>	Sumber data ulasan didapatkan (<i>google play/ app store</i>).
Sentimen	Kategori sentimen ulasan pengguna (positif/negatif).

B. Text Preprocessing

Setelah data dikumpulkan dan dilakukan proses *labeling*, maka tahap selanjutnya adalah melakukan *text preprocessing*. *Text preprocessing* dilakukan dengan membersihkan hal-hal yang tidak digunakan agar mendapatkan data yang berkualitas dan dapat diproses oleh model. Ada beberapa tahapan dalam melakukan *text preprocessing* yaitu data *selection*, *case folding*, *cleansing*, *tokenization*, *normalization*, dan *filtering*. Setiap tahapan akan dijelaskan lebih detail dibawah berikut.

- **Data Selection**
Pada tahap data *selection* dilakukan dengan menghapus beberapa kolom yang tidak digunakan dalam proses *training* yaitu dengan menghapus kolom *reviewId*, *userImage*, *rating*, *date*, *replyReview*, *repliedAt*, dan *Source*. Kemudian pada kolom sentimen dilakukan konversi dari data kategorik menjadi data numerik. Dimana sentimen label positif akan diubah menjadi angka 1 dan negatif menjadi angka 0.
- **Case Folding**
Data mentah ulasan yang didapatkan dari *play store* maupun *app store* memiliki format ukuran huruf yang berbeda, sehingga data tersebut harus dilakukan penyamaan format atau *case folding*. *Case folding* dilakukan dengan cara mengubah semua huruf kapital menjadi huruf kecil [5]. Hal ini dilakukan agar menghindari ambiguitas karena perbedaan format pada suatu kata. Contoh hasil dari *case folding* dapat dilihat pada tabel 2 dibawah berikut:

TABEL III
HASIL CASE FOLDING

Sebelum	Sesudah
Tolong ditingkatkan kembali aplikasinya, agar pengguna mudah dalam penggunaan aplikasi tsb. Tampilan UI nya juga sudah bagus untuk ditampilkan di layar.	tolong ditingkatkan kembali aplikasinya, agar pengguna mudah dalam penggunaan aplikasi tsb. tampilan ui nya juga sudah bagus untuk ditampilkan di layar.

- **Cleansing**
Cleansing adalah tahapan untuk menghapus karakter dan tanda baca yang tidak diperlukan. Karakter yang dihapus adalah karakter yang mengandung baris kosong, *emoticon*, *mention*, *link*, *URL*, dan *special character* lainnya. Contoh hasil dari *cleansing* dapat dilihat pada tabel 3 dibawah berikut:

TABEL IIIII
HASIL CLEANSING

Sebelum	Sesudah
Awesome!!!, Good aplikasi. Di tunggu update selanjut nya	Awesome Good aplikasi Di tunggu update selanjut nya

- **Tokenization**
Pada tahapan ini kalimat yang sudah di *cleansing* akan di lakukan *tokenization* dengan memecah kalimat ulasan menjadi potongan kata per kata. Hasil dari potongan kata per kata tersebut disebut dengan *token*. Contoh dari hasil *tokenization* dapat dilihat pada tabel 4 dibawah berikut:

TABEL IV
HASIL TOKENIZATION

Sebelum	Sesudah
mantap masalah status vaksinasi saya di terima dengan baik dan sudah selesai di atasi sekarang status vaksinasi saya sudah benar terima kasih	['mantap', 'masalah', 'status', 'vaksinasi', 'saya', 'di', 'terima', 'dengan', 'baik', 'dan', 'sudah', 'selesai', 'di', 'atasi', 'sekarang', 'status', 'vaksinasi', 'saya', 'sudah', 'benar', 'terima', 'kasih']

- **Normalization**
Normalization dilakukan untuk membersihkan teks yang berupa *slang words* atau bahasa keseharian menjadi kata Indonesia yang benar dan mengubah kata menjadi bahasa bakunya. Selain itu dalam proses ini juga akan dilakukan *stemming*. *Stemming* dilakukan dengan mengubah kata imbuhan menjadi kata dasarnya. Contoh dari hasil *normalization* dan *stemming* dapat dilihat pada tabel 5 dan tabel 6 dibawah berikut.

TABEL V
HASIL NORMALIZATION

Sebelum	Sesudah
['waktu', 'blm', 'di', 'upgrade', 'ataupun', 'udah', 'upgrade', 'sama', 'ajamau']	['waktu', 'belum', 'di', 'upgrade', 'ataupun', 'sudah', 'upgrade', 'sama', 'ajamau']

Sebelum	Sesudah
'daftar', 'ke', 'antrian', 'faskes', 'gangguan', 'terus', 'udah', 'berharhari', 'ga', 'bener', 'jg', 'tolong', 'diperbaiki', 'kerjanya']	'daftar', 'ke', 'antrian', 'faskes', 'gangguan', 'terus', 'sudah', 'berharhari', 'enggak', 'benar', 'juga', 'tolong', 'diperbaiki', 'kerjanya']

TABEL VI
HASIL STEMMING

Sebelum	Sesudah
['waktu', 'belum', 'di', 'upgrade', 'ataupun', 'sudah', 'upgrade', 'sama', 'ajamau', 'daftar', 'ke', 'antrian', 'faskes', 'gangguan', 'terus', 'sudah', 'berharhari', 'enggak', 'benar', 'juga', 'tolong', 'diperbaiki', 'kerjanya']	['waktu', 'belum', 'di', 'upgrade', 'atau', 'sudah', 'upgrade', 'sama', 'ajamau', 'daftar', 'ke', 'antri', 'faskes', 'ganggu', 'terus', 'sudah', 'berharhari', 'enggak', 'benar', 'juga', 'tolong', 'baik', 'kerja']

- **Filtering**

Pada tahapan ini dilakukan dengan menghapus beberapa kata yang sering muncul dalam jumlah yang banyak akan tetapi tidak memiliki makna atau *stopwords*. *Filtering* ini dilakukan agar dapat fokus pada kata yang lebih penting dan tidak terlalu umum penggunaannya [3]. Contoh kata yang dihapus dapat dilihat pada tabel 7 dibawah berikut:

TABEL VII
HASIL FILTERING

anda	lain	yang	mereka	di
jika	maka	lagi	sehingga	saya
telah	bisa	bahwa	yaitu	sampai

C. Word Embedding

Word embedding dilakukan untuk mengetahui persamaan *semantic* atau makna dari sebuah kata yang berada pada konteks yang sama [8]. Proses ini dilakukan karena dalam mengolah data dalam bentuk teks perlu dilakukannya perubahan format data menjadi bentuk *vektor* sebelum dimasukan kedalam model *neural network*. Terdapat 2 *framework* populer yang sering digunakan untuk mempelajari *word embedding* yaitu *Word2Vec* dan *GloVe*. Pada penelitian ini menggunakan *embedding layer* untuk mengolah data teks. *Embedding layer* memungkinkan untuk dapat mengubah setiap kata menjadi *vektor* dengan panjang tetap dengan ukuran yang telah ditentukan. *Vektor* yang dihasilkan adalah *vektor* padat dengan nilai riil, bukan hanya 0 dan 1. Panjang *vektor* kata yang sama membantu kita untuk merepresentasikan kata-kata dengan cara yang lebih baik bersamaan dengan pengurangan dimensi.

D. Analisa Sentimen

Setelah dilakukan *text preprocessing* maka tahap berikutnya adalah analisa sentimen. Analisa sentimen dilakukan dengan mengklasifikasi polaritas ulasan ke dalam

kategori positif atau negatif. Pada tahap ini akan dilakukan pembagian data latih dan data uji. Dimana data latih akan dilakukan proses *modeling* dengan menggunakan 3 algoritma *deep learning* yang berbeda yaitu LSTM, CNN, dan ANN.

- **Long short-term memory**

Long Term-Short Memory merupakan jenis model pengembangan dari *recurrent neural network* (RNN) yang diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997. Kemampuan dalam mempelajari informasi yang tidak digunakan dan informasi yang harus tetap disimpan dalam *memory cell* merupakan salah satu hal yang menjadi kelebihan LSTM dibandingkan dengan RNN [9]. *Neuron LSTM* memiliki *memory cell* yang berfungsi untuk menyimpan informasi dan *gate units* untuk mengatur informasi dalam jangka waktu yang lebih lama. Pembuatan model LSTM diusulkan agar dapat menjadi solusi untuk mengatasi *vanishing gradient* yang terjadi pada RNN saat memproses *sequential* yang panjang [9] [10]. Dalam LSTM memiliki 3 gerbang yang digunakan untuk mengendalikan dan memperbarui informasi yaitu gerbang masukan (*input gate*), gerbang *forget* (*forget gate*), dan gerbang keluaran (*output gate*).

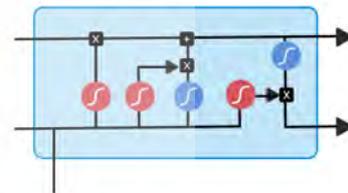


Fig. 2. Struktur LSTM

Dalam membangun model jaringan layer LSTM pada penelitian ini menggunakan beberapa layer lainnya. Lapisan yang pertama adalah layer *Embedding*, lapisan ini digunakan untuk mengubah data teks yang bentuknya alphanumeric menjadi *vector* dengan ukuran atau panjang input yang sama. Hal ini dilakukan agar data dapat diolah oleh model. Selanjutnya pada lapisan kedua adalah lapisan LSTM dengan jumlah units 64. Kemudian juga ditambahkan lapisan *dropout* yang digunakan untuk mengontrol *overfitting* pada model LSTM dan lapisan *dense* sebagai *hidden layer* dan juga *output layer*. Untuk arsitektur LSTM yang digunakan pada penelitian ini dapat dilihat pada Gbr. 3

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 60, 512)	2560000
lstm (LSTM)	(None, 64)	147712
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 16)	1040
dropout_1 (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 1)	17

Fig. 3. Arsitektur LSTM

- *One Dimensional Convolutional Neural Network* *Convolutional neural network* awalnya banyak digunakan untuk dalam pemrosesan gambar, akan tetapi saat ini juga dapat digunakan dalam melakukan klasifikasi teks dan telah mengungguli pendekatan berbasis *sequence*. *Convolutional neural network* membangun *feature map* dengan menggunakan lapisan konvolusi dan lapisan subsampling (*maximum pooling layer*) [11].

Dalam penelitian ini jaringan *convolutional neural network* yang digunakan adalah Conv1D. Adapun konfigurasi yang digunakan untuk membangun model CNN-1D terdiri dari lapisan *embedding*, lapisan Conv1D, lapisan *dropout* untuk menangani *overfitting* pada model, *max pooling layer*, *flatten layer*, dan lapisan *dense* sebagai *hidden layer* dan juga *output layer*. Untuk arsitektur CNN-1D yang digunakan dapat dilihat pada Gbr. 4

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 60, 512)	2560000
conv1d (Conv1D)	(None, 58, 64)	98368
dropout (Dropout)	(None, 58, 64)	0
max_pooling1d (MaxPooling1D)	(None, 29, 64)	0
flatten (Flatten)	(None, 1856)	0
dense (Dense)	(None, 16)	29712
dropout_1 (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 1)	17

Fig. 4. Arsitektur CNN

- *Artificial Neural Network* *Artificial Neural Network* adalah teknik pembelajaran mesin yang populer dan dibuat berdasarkan cara kerja jaringan syaraf otak manusia. Penentuan jumlah *neuron* pada lapisan tersembunyi dan parameter sangat penting dalam kinerja *artificial neural network* [12]. *Artificial Neural Network* terdiri dari 3 lapisan, yaitu *input layer* sebagai masukan, *hidden layer* untuk memproses hasil dari *input layer*, dan *output layer* untuk menghasilkan keluaran.

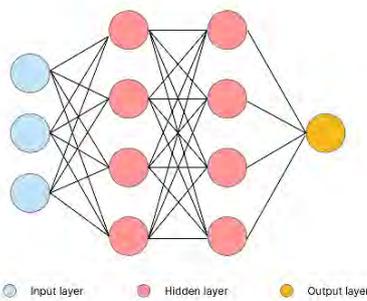


Fig. 5. Artificial Neural Network Layer

Pada penelitian ini arsitektur ANN menggunakan lapisan *embedding*, 1 lapisan *dropout*, 1 lapisan *hidden*

layer dengan *activation function* menggunakan *Relu* dan 1 lapisan *output* dengan *activation function* menggunakan *sigmoid*. Untuk arsitektur ANN yang digunakan pada penelitian ini dapat dilihat pada Gbr. 6

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 60, 512)	2560000
flatten (Flatten)	(None, 30720)	0
dropout (Dropout)	(None, 30720)	0
dense (Dense)	(None, 16)	491536
dropout_1 (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 1)	17

Fig. 6. Arsitektur ANN

E. Validasi dan Evaluasi

Setelah data di *training* menggunakan 3 algoritma, selanjutnya dilakukan validasi dan evaluasi terhadap model algoritma. Validasi menggunakan *K-Fold cross-validation* dengan iterasi yang dilakukan sebanyak 10 *fold*. *K-Fold cross-validation* adalah teknik validasi model dengan membagi dataset secara acak ke dalam tiap bagian sebanyak K lipatan [13]. Pada setiap *fold* akan digunakan sebagai data uji dan sisanya sebagai data latih yang dilakukan secara berulang sebanyak nilai *K-fold*.

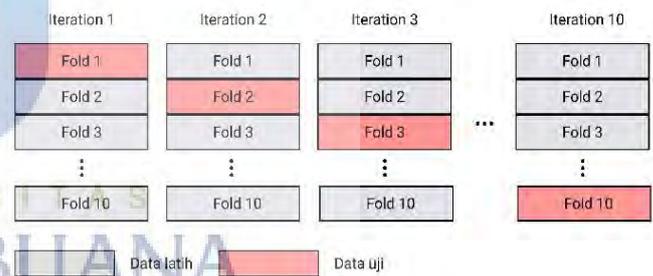


Fig. 7. K-Fold Cross Validation

Sedangkan untuk evaluasi akan diukur menggunakan *confusion matrix* untuk mendapatkan nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

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

$$Recall = \frac{TP}{TP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$F1-Score = \frac{2x(recall \times precision)}{recall+precision}$$

		PREDICTED	
		0	1
ACTUAL	0	TN	FP
	1	FN	TP

Fig. 8. Confusion Matrix

Untuk 0 merupakan negatif dan 1 merupakan positif. *Confusion matrix* memiliki 4 kemungkinan yaitu:

- 1) *True Negative* (TN) : model memprediksi ulasan adalah kalimat negatif dan ulasan sebenarnya adalah kalimat negatif.
- 2) *False Positive* (FP) : model memprediksi ulasan adalah kalimat positif dan ulasan sebenarnya adalah kalimat negatif.
- 3) *False Negative* (FN) : model memprediksi ulasan adalah kalimat negatif dan ulasan sebenarnya adalah kalimat positif.
- 4) *True Positive* (TP) : model memprediksi ulasan adalah kalimat positif dan ulasan sebenarnya adalah kalimat positif.

IV. HASIL DAN PEMBAHASAN

Eksperimen dilakukan dengan melakukan beberapa percobaan. Percobaan dilakukan dengan mengubah komposisi pembagian data latih dan data uji, kemudian juga akan dilakukan *hyperparameter tuning* dan *dropout regularization tuning*. Pada setiap eksperimen akan menggunakan pengaturan parameter dasar yaitu dengan komposisi pembagian data 70% data latih dan 30% data uji, nilai *dropout* = 0.5, jumlah *batch size* = 64, *learning rate* = 0.0001, dan menggunakan *optimizer* = adam.

A. Pembagian Data Latih dan Data Uji

Eksperimen pertama yang dilakukan adalah pembagian data latih dan data uji. Pembagian data latih dan data uji akan dilakukan 3 tahap skenario, yaitu pertama dengan membagi data pada komposisi 90% data latih dan 10% data uji, skenario kedua dengan komposisi 80% data latih dan 20% data uji, dan skenario ketiga dengan komposisi 70% data latih dan 30% data uji. Eksperimen menggunakan parameter dasar yaitu jumlah *batch size* = 64, *learning rate* = 0.0001, dan menggunakan *optimizer* = adam. Pada tabel 8 menyajikan seluruh hasil pengujian dengan 3 skenario pembagian data latih dan data uji. Hasil dari semua komposisi pembagian data yang telah dilakukan menunjukkan tingkat akurasi yang baik pada ketiga algoritma. Untuk algoritma LSTM dan CNN memiliki tingkat akurasi tertinggi pada komposisi 70% data latih dan 30% data uji dengan masing masing tingkat akurasinya adalah 95,12% untuk LSTM, dan 93,10% untuk CNN. Sedangkan untuk algoritma ANN memiliki tingkat akurasi tertinggi pada komposisi 80% data latih dan 20% data uji dengan tingkat akurasi sebesar 92,93%. Keseluruhan hasil

pengujian pembagian komposisi data latih dan data uji dapat dilihat pada tabel 8.

TABEL VIII
HASIL PENGUJIAN PEMBAGIAN DATA

Metode	Performance				
	Komposisi	Accuracy	Precision	Recall	F1- Score
LSTM	10/90	93,94%	95,74%	91,83%	93,5%
	20/80	94%	94%	94%	94%
	30/70	95,12%	95,51%	94,53%	95,02%
CNN	10/90	90,91%	91,66%	89,79%	90,72%
	20/80	91,6%	93,26%	90%	91,60%
	30/70	93,10%	93,75%	92,15%	92,94%
ANN	10/90	92,42%	92,78%	91,83%	92,30%
	20/80	92,68%	93,84%	91,5%	92,65%
	30/70	91,25%	90,57%	91,80%	91,18%

B. Hyperparameter Tuning

Eksperimen selanjutnya adalah dengan melakukan *hyperparameter tuning*. Proses dilakukan dengan mengubah beberapa pengaturan seperti jumlah *batch size*, *learning rate*, dan jenis *optimizer*. Hal ini dilakukan agar mendapatkan performa terbaik dari ketiga algoritma dengan beberapa pengaturan yang berbeda. Proses *hyperparameter tuning* menggunakan rasio pembagian data sebesar 70:30 dan pengaturan parameter dasarnya yaitu *optimizer* = adam, *batch size* = 64, dan *learning rate* = 0.0001.

Skenario pertama pada proses *hyperparameter tuning* adalah dengan mencoba beberapa pengaturan yang berbeda pada jumlah *batch size*. *Batch size* adalah parameter yang digunakan dalam *neural network* untuk mengatur jumlah sampel *training* dalam satu iterasi [14]. Hasil percobaan dari skenario pengaturan jumlah *batch size* didapatkan hasil terbaik algoritma LSTM adalah pada jumlah *batch size* = 32 dengan tingkat akurasi sebesar 95,29%, kemudian untuk algoritma CNN memiliki tingkat akurasi terbaik pada jumlah *batch size* = 64 dengan akurasi sebesar 93,10% dan untuk algoritma ANN memiliki tingkat akurasi terbaik pada jumlah *batch size* = 32 dengan tingkat akurasi sebesar 93,27%. Keseluruhan hasil percobaan skenario *hyperparameter tuning* jumlah *batch size* dapat dilihat pada tabel 9.

TABEL IX
HASIL TUNING PARAMETER BATCH SIZE

Metode	Performance				
	Batch size	Accuracy	Precision	Recall	F1- Score
LSTM	32	95,29%	94,91%	95,56%	95,23%
	64	95,12%	95,51%	94,53%	95,02%
	128	91,58%	90,36%	92,3%	91,58%
	256	86,53%	84,91%	88,39%	86,62%
CNN	32	92,93%	93,12%	92,49%	92,80%
	64	93,10%	93,75%	92,15%	92,94%
	128	90,57%	88,85%	92,49%	90,63%

Metode	Performance				
	Batch size	Accuracy	Precision	Recall	F1- Score
	256	85,69%	83,76%	88,05%	85,85%
ANN	32	93,27%	93,47%	92,83%	93,15%
	64	91,25%	90,57%	91,80%	91,18%
	128	90,57%	89,36%	91,80%	90,57%
	256	86,36%	86,30%	86%	86,15%

Skenario kedua adalah dengan melakukan tuning parameter pada nilai *learning rate*. *Learning rate* adalah parameter *training* yang digunakan untuk menghitung nilai koreksi bobot selama proses *training*. Besaran nilai pada *learning rate* dapat mempengaruhi akurasi, jika nilai *learning rate* semakin besar maka akurasi menjadi tidak optimal dan jika besaran nilai *learning rate* semakin kecil maka akurasi menjadi meningkat akan tetapi proses *training* akan memakan waktu lebih lama [8]. Hasil menunjukkan bahwa algoritma LSTM dan CNN memiliki tingkat akurasi terbaik pada *learning rate* sebesar 0.0001 yaitu dengan tingkat akurasi masing-masing adalah LSTM = 95,12% dan CNN = 93,77%. Sedangkan untuk algoritma ANN memiliki tingkat akurasi terbaik pada *learning rate* sebesar 0.001 yaitu dengan tingkat akurasinya adalah 91,58%. Keseluruhan hasil percobaan skenario *tuning* parameter nilai *learning rate* dapat dilihat pada tabel 10.

TABEL X
HASIL TUNING PARAMETER LEARNING RATE

Metode	Performance				
	Learning rate	Accuracy	Precision	Recall	F1- Score
LSTM	0.0001	95,12%	95,51%	94,53%	95,02%
	0.001	92,42%	91,05%	93,85%	92,43%
	0.01	92,93%	93,42%	92,15%	92,78%
CNN	0.0001	93,10%	93,75%	92,15%	92,94%
	0.001	92,26%	90,75%	93,85%	92,28%
	0.01	92,76%	92,80%	92,49%	92,64%
ANN	0.0001	91,25%	90,57%	91,80%	91,18%
	0.001	91,58%	90,63%	92,49%	91,55%
	0.01	90,91%	88,17%	94,19%	91,08%

Skenario ketiga adalah dengan melakukan *tuning* parameter dengan mengubah jenis *optimizer* yang digunakan. Hasil menunjukkan bahwa algoritma LSTM memiliki tingkat akurasi terbaik dengan menggunakan jenis *optimizer* = Adam dengan tingkat akurasinya adalah 95,12%. Kemudian untuk algoritma CNN memiliki tingkat akurasi terbaik dengan menggunakan jenis *optimizer* = Nadam dengan tingkat akurasinya sebesar 93,60% dan algoritma ANN memiliki tingkat akurasi terbaik dengan menggunakan jenis *optimizer* = RMSprop dengan tingkat akurasinya sebesar 92,26%. Keseluruhan hasil percobaan skenario *tuning* parameter jenis *optimizer* dapat dilihat pada tabel 11.

TABEL XI

HASIL TUNING PARAMETER JENIS OPTIMIZER

Metode	Performance				
	Optimizer	Accuracy	Precision	Recall	F1- Score
LSTM	Adam	95,12%	95,51%	94,53%	95,02%
	Nadam	93,75%	91,83%	95,90%	93,82%
	Adamax	84,18%	82,83%	85,66%	84,22%
	RMSprop	94,14%	94,17%	93,85%	94,01%
CNN	Adam	93,10%	93,75%	92,15%	92,94%
	Nadam	93,60%	94,11%	92,83%	93,47%
	Adamax	82,66%	82,31%	82,59%	82,45%
	RMSprop	92,76%	93,10%	92,15%	92,62%
ANN	Adam	91,25%	90,57%	91,80%	91,18%
	Nadam	92,09%	90,72%	93,51%	92,10%
	Adamax	81,65%	81,22%	81,50%	81,36%
	RMSprop	92,26%	94,26%	89,76%	91,95%

Dari hasil eksperimen dengan melakukan *hyperparameter tuning* dapat disimpulkan bahwa ketiga algoritma memperoleh akurasi yang lebih tinggi dari sebelum melakukan *hyperparameter tuning* atau hanya dengan menggunakan parameter dasar. Algoritma LSTM mendapatkan akurasi tertinggi dengan pengaturan parameter *batch size* = 32, *learning rate* = 0.0001, dan *optimizer* = adam. Kemudian untuk algoritma CNN mendapatkan akurasi tertinggi dengan pengaturan parameter *batch size* = 64, *learning rate* = 0.0001, dan *optimizer* = Nadam. Sedangkan untuk algoritma ANN mendapatkan akurasi tertinggi dengan pengaturan parameter *batch size* = 32, *learning rate* = 0.001, dan *optimizer* = RMSprop. Tidak ada acuan nilai parameter yang tepat untuk mendapatkan model dengan performa yang baik, setiap kasus akan berbeda tergantung dari data yang digunakan. Tabel 12 menunjukkan perbandingan akurasi sebelum dan sesudah *hyperparameter tuning*.

TABEL XII
PERBANDINGAN HASIL HYPERPARAMETER TUNING

Metode	Akurasi	
	Sebelum hyperparameter tuning	Setelah hyperparameter tuning
LSTM	95,12%	95,29%
CNN	93,10%	93,60%
ANN	91,25%	92,59%

C. Dropout Regularization Tuning

Eksperimen selanjutnya adalah melakukan *dropout regularization tuning*. *Dropout* merupakan suatu teknik regularisasi yang digunakan untuk meningkatkan kinerja *neural network* dalam mengatasi permasalahan *overfitting* [15]. *Overfitting* terjadi ketika model memiliki akurasi yang baik pada data *training* akan tetapi memiliki performa yang buruk pada data *testing*. *Dropout* secara acak akan menjatuhkan (*dropping out*) sejumlah *neuron* dalam satu lapisan, sehingga *neuron* tersebut tidak digunakan kembali.

Hasil pengujian menunjukkan bahwa teknik *dropout* mampu mengurangi *overfitting* yang terjadi pada ketiga algoritma. Algoritma LSTM dan CNN memiliki selisih akurasi *training* dan *testing* terdekat dengan nilai *rate dropout* = 0.5. sedangkan untuk algoritma ANN memiliki selisih akurasi *training* dan *testing* terdekat dengan nilai *rate dropout* = 0.3. Tabel 13 menunjukkan perbandingan akurasi *testing* dan *training* dari hasil *dropout regularization tuning*.

TABEL XIII
PERBANDINGAN HASIL DROPOUT REGULARIZATION

Metode	Dropout	Train Accuracy	Test Accuracy	Selisih
LSTM	0.3	98,92%	95,62%	3,30%
	0.5	97,33%	95,12%	2,21%
	Without Dropout	99,68%	94,70%	4,98%
CNN	0.3	98,63%	93,10%	5,53%
	0.5	97,33%	93,77%	3,56%
	Without Dropout	99,35%	92,93%	6,42%
ANN	0.3	97,19%	93,43%	3,76%
	0.5	95,60%	91,25%	4,35%
	Without Dropout	99,42%	93,27%	6,15%

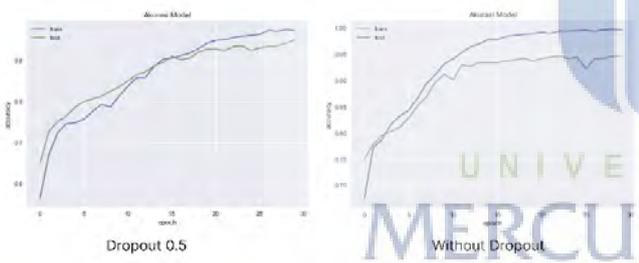


Fig. 9. Grafik akurasi LSTM

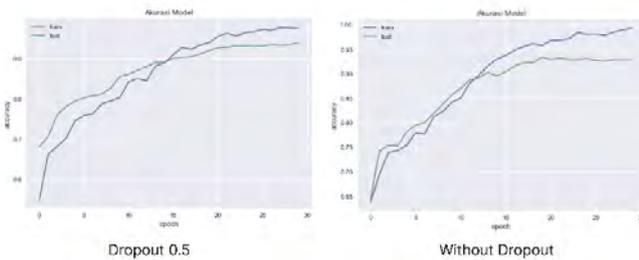


Fig. 10. Grafik akurasi CNN

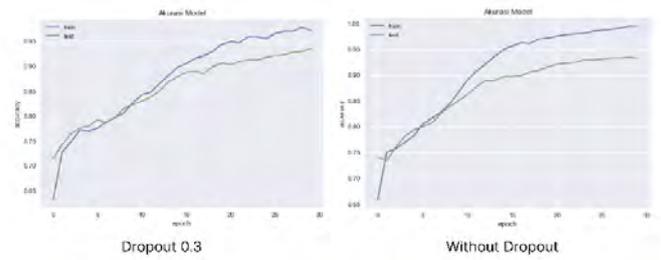


Fig. 11. Grafik akurasi ANN

D. Cross Validation dan Confusion Matrix

Selanjutnya adalah melakukan validasi terhadap model, validasi menggunakan *k-fold cross validation* dengan iterasi yang dilakukan sebanyak 10 *fold*. Pada setiap *fold* akan digunakan sebagai data uji dan sisanya sebagai data latih yang dilakukan secara berulang sebanyak nilai *K-fold*. Validasi model dilakukan agar mendapatkan akurasi yang lebih akurat dan menguji model apakah memiliki performa yang konsisten pada setiap data yang berbeda. Hasil pengujian menunjukkan bahwa dari ketiga algoritma memiliki performa akurasi yang tinggi dan konsisten pada setiap iterasinya. Pada algoritma LSTM memiliki rata-rata akurasi sebesar 94,78% dengan selisih akurasi setiap *fold* adalah +/- 0.28%. Selanjutnya pada algoritma CNN memiliki rata-rata akurasi sebesar 92,98% dengan selisih akurasi setiap *fold* adalah +/- 0.45%. kemudian untuk algoritma ANN memiliki rata-rata akurasi sebesar 92,66% dengan selisih akurasi setiap *fold* adalah +/- 0.61%. Tabel 14 menunjukkan perbandingan akurasi dari setiap *fold*.

TABEL XIV
HASIL PENGUJIAN K-FOLD CROSS VALIDATION

Fold	Metode		
	LSTM	CNN	ANN
1	94,44%	93,10%	93,43%
2	94,44%	92,93%	93,27%
3	95,12%	92,76%	93,43%
4	94,78%	92,59%	91,75%
5	94,44%	92,42%	92,59%
6	95,29%	93,60%	93,10%
7	94,95%	93,77%	92,26%
8	94,95%	93,43%	92,76%
9	94,61%	92,59%	91,75%
10	94,78%	92,59%	92,26%
Mean	94,78% (+/-0.28%)	92,98% (+/-0.45%)	92.66% (+/- 0.61%)

Kemudian model dilakukan evaluasi dengan menggunakan *confusion matrix*. Evaluasi ini dilakukan agar dapat melihat hasil prediksi analisa sentimen dari model dalam 4 kemungkinan yaitu *true positive*, *false positive*, *true negative*, dan *false negative*. Tabel 15 menunjukkan hasil perbandingan evaluasi *confusion matrix*.

TABEL XV
HASIL PENGUJIAN *CONFUSION MATRIX*

Metode	True Positive	False Positive	True Negative	False Positive
LSTM	277	13	288	16
CNN	276	20	281	17
ANN	268	17	284	25

V. KESIMPULAN

Dari seluruh eksperimen yang telah dilakukan dengan hasil klasifikasi polaritas ulasan pengguna aplikasi JAKI pada setiap algoritma memberikan beberapa hasil perbandingan. Pada eksperimen pertama yaitu membandingkan pembagian data uji dan data latih, komposisi pembagian data terbaik diperoleh pada komposisi 70% data latih dan 30% data uji untuk algoritma LSTM (95,12%) dan CNN (93,10%), sedangkan untuk algoritma ANN (92,68%) mendapatkan akurasi terbaik pada komposisi 80% data latih dan 20% data uji (lihat tabel 8).

Pada eksperimen kedua yaitu melakukan *hyperparameter tuning*. Hasil dari eksperimen ini menunjukkan bahwa *hyperparameter tuning* dapat meningkatkan akurasi dari ketiga algoritma dengan mencari parameter optimal, dimana akurasi LSTM meningkat sebesar 0,17%, CNN meningkat sebesar 0,5% dan ANN meningkatkan sebesar 1,34% (lihat tabel 12).

Pada eksperimen ketiga yaitu melakukan *dropout regularization tuning*. Hasil dari eksperimen ini menunjukkan bahwa teknik *dropout regularization* mampu menangani permasalahan *overfitting* pada model *neural network*, dimana algoritma LSTM dan CNN memiliki selisih akurasi *training* dan *testing* terdekat dengan nilai *rate dropout* = 0,5, sedangkan untuk algoritma ANN memiliki selisih akurasi *training* dan *testing* terdekat dengan nilai *rate dropout* = 0,3 (lihat tabel 13).

Pada tahap validasi dengan menggunakan *cross validation* menunjukkan hasil bahwa ketiga algoritma memiliki performa yang optimal dan konsisten, dimana algoritma LSTM memiliki rata-rata tingkat akurasi dari 10 *fold* sebesar 94,78%, kemudian yang kedua adalah algoritma CNN memiliki rata-rata tingkat akurasi dari 10 *fold* sebesar 92,98%, dan terakhir adalah algoritma ANN dengan rata-rata tingkat akurasi dari 10 *fold* sebesar 92,66% (lihat tabel 14). Selanjutnya dilakukan evaluasi model dengan menggunakan *confusion matrix*. Hasil evaluasi menunjukkan dari setiap model mampu memprediksi data sentimen ulasan dengan tepat yang lebih banyak dibandingkan dengan hasil prediksi yang salah. Dimana algoritma LSTM memprediksi 277 *true positive*, 13 *false positive*, 288 *true negative*, dan 16 *false negative*. Kemudian untuk algoritma CNN memprediksi 276 *true positive*, 20 *false positive*, 281 *true negative*, dan 17 *false negative*. Dan untuk algoritma ANN memprediksi 268 *true positive*, 17 *false positive*, 284 *true negative*, dan 25 *false negative* (lihat tabel 15).

Dilihat dari semua eksperimen yang telah dilakukan menunjukkan hasil bahwa algoritma LSTM memiliki performa yang lebih baik dalam melakukan analisa sentimen ulasan pengguna aplikasi JAKI dibandingkan dengan algoritma CNN dan ANN.

DAFTAR PUSTAKA

- [1] A. J. Nair, G. Veena, and A. Vinayak, "Comparative study of Twitter Sentiment on COVID - 19 Tweets," in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, Apr. 2021, pp. 1773–1778. doi: 10.1109/ICCMC51019.2021.9418320.
- [2] F. Pradana Rachman, H. Santoso, and R. Artikel, "Perbandingan Model Deep Learning Untuk Klasifikasi Sentiment Analisis Dengan Teknik Natural Language Processing," *Jurnal Teknologi dan Manajemen Informatika*, vol. 7, no. 2, pp. 103–112, 2021, [Online]. Available: <http://http://jurnal.unmer.ac.id/index.php/jtmi>
- [3] A. Bayhaqy, S. Sfenrianto, K. Nainggolan, and E. R. Kaburuan, "Sentiment Analysis about E-Commerce from Tweets Using Decision Tree, K-Nearest Neighbor, and Naïve Bayes," *2018 International Conference on Orange Technologies (ICOT)*, pp. 1–6, Jul. 2018, doi: 10.1109/ICOT.2018.8705796.
- [4] D. Andriyanto, F. Said, F. Titiani, and E. Erni, "Analisis Kesuksesan Aplikasi Jakarta Kini (JAKI) Menggunakan Model Delone and McLean," *Paradigma - Jurnal Komputer dan Informatika*, vol. 23, no. 1, pp. 43–48, Mar. 2021, doi: 10.31294/p.v23i1.10018.
- [5] C. K. N. Paputungan and A. Jacobus, "Sentiment Analysis of Social Media Users Using Long-Short Term Memory Method," *Jurnal Teknik Elektro dan Komputer*, vol. 10, no. 2, pp. 99–106, 2021.
- [6] A. Marendra Ramadhani and H. Soon Goo, "Twitter Sentiment Analysis using Deep Learning Methods," *2016 6th International Annual Engineering Seminar (InAES)*, pp. 6–10, 2017, doi: 10.1109/INAES.2016.7821897.
- [7] N. Aslam, W. Y. Ramay, K. Xia, and N. Sarwar, "Convolutional neural network based classification of app reviews," *IEEE Access*, vol. 8, Institute of Electrical and Electronics Engineers Inc., pp. 185619–185628, 2020. doi: 10.1109/ACCESS.2020.3029634.
- [8] A. S. Saksesi, M. Nasrun, and C. Setianingsih, "Analysis Text of Hate Speech Detection Using Recurrent Neural Network," *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pp. 242–248, 2018, doi: 10.1109/ICCEREC.2018.8712104.
- [9] A. Yahyadi, F. Latifah, P. Studi Informatika, and F. T. Informasi, "ANALISIS SENTIMEN TWITTER TERHADAP KEBIJAKAN PPKM DI TENGAH PANDEMI COVID-19 MENGGUNAKAN MODE LSTM," *464 Journal of Information System, Applied, Management, Accounting and Research. Issue Period*, vol. 6, no. 2, pp. 464–470, 2022, doi: 10.52362/jisamar.v6i2.791.
- [10] M. Q. Andiyantama, I. Zahira, and A. Irawan, "Prediksi Energi Listrik Kincir Angin Berdasarkan Data Kecepatan Angin Menggunakan LSTM," *JITCE (Journal of Information Technology and Computer Engineering)*, vol. 5, no. 01, pp. 1–7, Mar. 2021, doi: 10.25077/jitce.5.01.1-7.2021.
- [11] B. Jang, M. Kim, G. Harerimana, S. U. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention mechanism," *Applied Sciences (Switzerland)*, vol. 10, no. 17, pp. 1–14, Sep. 2020, doi: 10.3390/app10175841.
- [12] M. M. Saritas and A. Yasar, "Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification," *Original Research Paper International Journal of Intelligent Systems and Applications in Engineering IJISAE*, vol. 7, no. 2, pp. 88–91, 2019, doi: 10.1039/b000000x.
- [13] S. Dwiansnati and Y. Devianto, "Optimasi Prediksi Keputusan Calon Nasabah Potensial Berbasis Particle Swarm Optimization," *JURNAL INFORMATIKA*, vol. 6, no. 2, pp. 286–292, 2019, [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- [14] N. Rochmawati *et al.*, "Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer Adam," *Journal Information Engineering and Educational Technology*, vol. 5, pp. 44–48, 2021.
- [15] S. Mehta, C. Paunwala, and B. Vaidya, "CNN based Traffic Sign Classification using Adam Optimizer," *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 1293–1298, 2019, doi: 10.1109/ICCS45141.2019.9065537.

KERTAS KERJA

Ringkasan

Kertas kerja ini merupakan material kelengkapan artikel jurnal dengan judul "Perbandingan Metode *Deep Learning* Dalam Analisa Sentimen Ulasan Aplikasi Jakarta Kini (JAKI)". Kertas kerja berisi semua material hasil penelitian Tugas Akhir yang tidak dimuat/atau disertakan di artikel jurnal. Di dalam kertas kerja ini disajikan: *literature review*, dataset yang digunakan, *source code*, dan hasil eksperimen secara keseluruhan.

Pada bagian *literature review* membahas mengenai beberapa penelitian terdahulu yang berkaitan dengan penelitian dan disajikan dalam bentuk tabel yang berisi judul penelitian, metode penelitian, dan hasil yang didapatkan dari penelitian. Kemudian pada bagian analisa dan perancangan menjelaskan mengenai latar belakang masalah penelitian dan perancangan penelitian. Pada bagian *source code* menjabarkan seluruh *code* yang digunakan mulai dari tahap *import library*, pengumpulan data, *text preprocessing*, pembuatan model, validasi, dan evaluasi model. Pada bagian *dataset* menjelaskan mengenai bagaimana data didapatkan, sumber data didapatkan, dan atribut-atribut pada *dataset*. Pada bagian hasil eksperimen membahas mengenai keseluruhan eksperimen yang dilakukan, seperti pembagian data uji dan data latih, *hyperparameter tuning*, *dropout regularization tuning*, validasi dengan *cross validation* dan evaluasi dengan *confusion matrix*

