

**TESIS**

**ANALISIS SENTIMEN PADA REVIEW PRODUK SKINCARE  
MENGUNAKAN WORD EMBEDDING DAN METODE LONG SHORT-  
TERM MEMORY (LSTM)**



Disusun oleh:

**Nama : Rona Gulnes Purnasiwi**  
**NIM : 20.55.1425**  
**Konsentrasi : Informatics Technopreneurship**

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2023**

**TESIS**

**ANALISIS SENTIMEN PADA REVIEW PRODUK SKINCARE  
MENGUNAKAN WORD EMBEDDING DAN METODE LONG SHORT-  
TERM MEMORY (LSTM)**

**SENTIMENT ANALYSIS ON SKINCARE PRODUCT REVIEWS USING  
WORD EMBEDDING AND THE LONG SHORT-TERM MEMORY  
(LSTM) METHOD**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

**Nama : Rona Gulnes Purnaswi**  
**NIM : 20.55.1425**  
**Konsentrasi : Informatics Technopreneurship**

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2023**

## HALAMAN PENGESAHAN

**ANALISIS SENTIMEN PADA REVIEW PRODUK SKINCARE MENGGUNAKAN  
WORD EMBEDDING DAN METODE LONG SHORT-TERM MEMORY (LSTM)**

**SENTIMENT ANALYSIS ON SKINCARE PRODUCT REVIEWS USING WORD  
EMBEDDING AND THE LONG SHORT-TERM MEMORY (LSTM) METHOD**

Dipersiapkan dan Disusun oleh

**Rona Guines Purnasiwi**

**20.55.1425**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Senin, 06 November 2023

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 06 November 2023

**Rektor**

**Prof. Dr. M. Suyanto, M.M.**

**NIK. 190302001**

## HALAMAN PERSETUJUAN

**ANALISIS SENTIMEN PADA REVIEW PRODUK SKINCARE MENGGUNAKAN  
WORD EMBEDDING DAN METODE LONG SHORT-TERM MEMORY (LSTM)**

**ENTIMENT ANALYSIS ON SKINCARE PRODUCT REVIEWS USING WORD  
EMBEDDING AND THE LONG SHORT-TERM MEMORY (LSTM) METHOD**

Dipersiapkan dan Disusun oleh

**Rona Guines Purnasiwi**

**20.55.1425**

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Senin, 06 November 2023

**Pembimbing Utama**

**Prof. Dr. Kusriani, M.Kom**  
NIK. 190302106

**Anggota Tim Penguji**

**Prof. Dr. Ema Utami, S.Si., M.Kom**  
NIK. 190302037

**Pembimbing Pendamping**

**M. Hanafi, S.Kom, M.Eng, Ph.D.**  
NIK. 190302024

**Dr. Alva Hendi Muhammad, S.T, M.Eng**  
NIK. 190302493

**Prof. Dr. Kusriani, M.Kom**  
NIK. 190302106

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer.

Yogyakarta, 06 November 2023

**Direktur Program Pascasarjana**

**Prof. Dr. Kusriani, M.Kom.**  
NIK. 190302106

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Rona Guines Purnasiwi  
NIM : 20.55.1425  
Konsentrasi : Informatics Technopreneurship

Menyatakan bahwa Tesis dengan judul berikut:  
**Analisis Sentimen Pada Review Produk Skincare Menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM)**

Dosen Pembimbing Utama : Prof. Dr. Kusriani, M.Kom  
Dosen Pembimbing Pendamping : M. Hanafi, S.Kom, M.Eng, Ph.D

1. Karya tulis ini adalah benar-benar **ASLI** dan **BELUM PERNAH** diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian **SAYA** sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab **SAYA**, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini **SAYA** buat dengan **sesungguhnya**, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka **SAYA** bersedia menerima **SANKSI AKADEMIK** dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 06 November 2023  
Yang Menyatakan,



Rona Guines Purnasiwi

## HALAMAN PERSEMBAHAN

Dengan mengucapkan rasa syukur kepada Allah SWT telah memberikan nikmat sehat dan kemudahan dalam menyelesaikan skripsi ini, dan skripsi ini saya persembahkan untuk:

1. Allah SWT karena berkat izin-Nya telah memberi kemudahan penulis dalam menyelesaikan penelitian ini.
2. Kedua orang tua penulis serta keluarga yang selalu memberikan dukungan dan tak henti-henti berdo'a serta kasih sayang tak terhingga kepada penulis sampai terselesaikannya penelitian ini.
3. Ibu Prof. Dr. Kusriani, M.Kom dan Bapak M. Hanafi, S.Kom, M.Eng, Ph.D, selaku dosen pembimbing yang senantiasa membimbing dengan penuh kesabaran, memberi dukungan serta sampai penulis dapat menyelesaikan penelitian ini.
4. DKUI Family dan Tim Podcast @amikomjogja.
5. Seluruh Pihak yang mendukung jalannya penelitian ini.

Harapan penulis semoga jasa dan kebaikan dari semua pihak yang telah membantu terselesaikannya skripsi ini mendapat balasan pahala yang berlipat ganda dari Allah SWT.

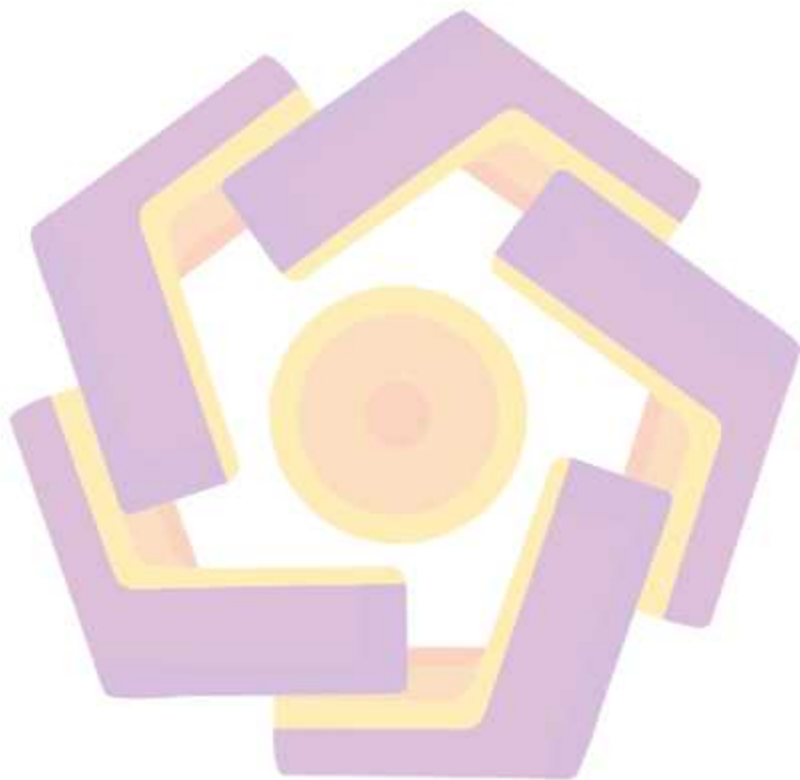
Dalam penyusunan skripsi ini penulis menyadari masih banyak kekurangan, oleh sebab itu penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan skripsi ini. Semoga skripsi ini bermanfaat bagi penulis serta pembaca.



## HALAMAN MOTTO

“Jadilah apa yang kamu inginkan, bukan karena ingin dilihat orang lain”

-Park Jimin-



## KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat-Nya, kami dapat menyelesaikan penyusunan skripsi dengan tepat waktu yang menjadi salah satu faktor kelulusan di Universitas Bina Nusantara. Pada kesempatan ini, kami ucapkan terima kasih kepada berbagai pihak yang telah memberikan bimbingan, dukungan, dan partisipasi dalam penyelesaian skripsi ini.

Kami ucapkan terima kasih kepada:

1. **Dr. Kusriani, M.Kom** selaku Direktur Program Pascasarjana.
2. M. Hanafi, S.Kom, M.Eng. Ph.D sebagai Pembimbing Pendamping.
3. Dr. Alva Hendi Muhammad, S.T.M.Eng dan Prof. Dr. Ema Utami, S.Si., M.Kom sebagai Tim Penguji.
4. Dosen-dosen Program Pascasarjana Universitas Amikom Yogyakarta yang telah memberikan ilmu dan tuntunan selama kuliah sehingga laporan skripsi ini dapat diselesaikan dengan baik.
5. Orang tua dan keluarga besar kami yang telah memberikan dukungan, baik moral, material, serta spiritual yang begitu besar sehingga kami mendapatkan dukungan yang sangat berarti sehingga kami dapat menyelesaikan penyusunan laporan skripsi dengan tepat waktu.
6. Teman-teman, kerabat, dan semua pihak yang tidak dapat disebutkan satu persatu yang secara langsung maupun tidak langsung terlibat dalam penyusunan laporan skripsi ini.

Akhir kata, dengan rendah hati penulis sadari bahwa skripsi ini tidak luput dari kesalahan sehingga penulis meminta maaf atas kata yang kurang berkenan. Semoga skripsi ini dapat bermanfaat bagi pembaca dan penulis terbuka terhadap kritik dan saran. Atas perhatian pembaca, saya ucapkan terima kasih.

Yogyakarta, 06 November 2023

Penulis



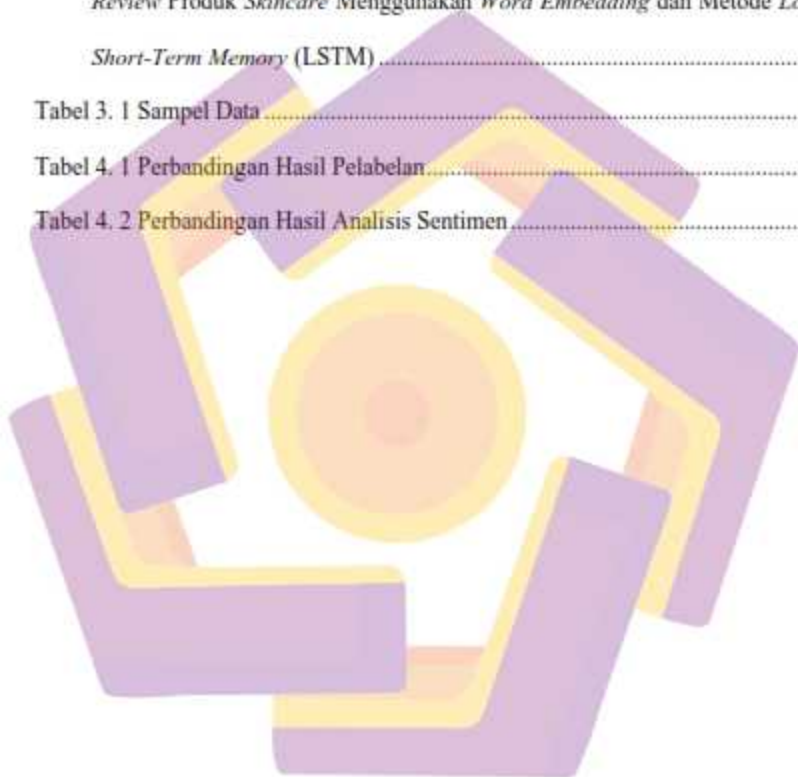
## DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS .....	v
HALAMAN PERSEMBAHAN .....	vi
HALAMAN MOTTO .....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
INTISARI.....	xv
<i>ABSTRACT</i> .....	xvi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah .....	1
1.2. Rumusan Masalah.....	9
1.3. Batasan Masalah .....	9
1.4. Tujuan Penelitian .....	10
1.5. Manfaat Penelitian .....	11
BAB II TINJAUAN PUSTAKA.....	12
2.1. Tinjauan Pustaka .....	12
2.2. Keaslian Penelitian.....	16
2.3. Landasan Teori.....	23
2.3.1 Analisis Sentimen.....	23
2.3.2 <i>Preprocessing</i> .....	23
2.3.3 <i>Word2Vec</i> .....	24
2.3.4 Metode <i>Long Short-Term Memory (LSTM)</i> .....	25
2.3.5 <i>Confussion Matrix</i> .....	28
BAB III METODE PENELITIAN.....	30

3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	30
3.2. Metode Pengumpulan Data.....	30
3.3. Metode Analisis Data.....	34
3.4. Alur Penelitian.....	35
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>37</b>
4.1. Pengumpulan Data.....	37
4.2. <i>Preprocessing</i> .....	37
4.3. Algoritma LSTM Dengan Word2Vec.....	39
4.5.1 Splitting Data 70:30.....	40
4.5.2 Splitting Data 80:20.....	44
4.5.3 Splitting Data 90:10.....	48
4.4. Evaluasi <i>Confusion Matrix</i> Dengan Word2Vec.....	52
4.4.1 Hasil Evaluasi LSTM dengan Splitting Data 70:30.....	52
4.4.2 Hasil Evaluasi LSTM dengan Splitting Data 80:20.....	54
4.4.3 Hasil Evaluasi LSTM dengan Splitting Data 90:10.....	56
4.5. Algoritma LSTM Tanpa Word2Vec.....	58
4.5.1 Splitting Data 70:30.....	59
4.5.2 Splitting Data 80:20.....	61
4.5.3 Splitting Data 90:10.....	64
4.6. Evaluasi <i>Confusion Matrix</i> Tanpa Word2Vec.....	67
4.4.4 Hasil Evaluasi LSTM dengan Splitting Data 70:30.....	67
4.4.5 Hasil Evaluasi LSTM dengan Splitting Data 80:20.....	69
4.4.6 Hasil Evaluasi LSTM dengan Splitting Data 90:10.....	71
4.7. Pembahasan.....	72
<b>BAB V PENUTUP.....</b>	<b>77</b>
5.1. Kesimpulan.....	77
5.2. Saran.....	78
<b>DAFTAR PUSTAKA.....</b>	<b>79</b>

## DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian Analisis Sentimen pada <i>Review Produk Skincare Menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM)</i> .....	16
Tabel 3. 1 Sampel Data .....	32
Tabel 4. 1 Perbandingan Hasil Pelabelan .....	73
Tabel 4. 2 Perbandingan Hasil Analisis Sentimen .....	75



## DAFTAR GAMBAR

Gambar 2. 1	Arsitektur CBOV dan Skip-gram .....	25
Gambar 2. 2	Desain Algoritma <i>Long Short-Term Memory</i> .....	26
Gambar 3. 1	Alur Penelitian.....	35
Gambar 4. 1	<i>Web Scrapping</i> dari Femaledaily.com .....	37
Gambar 4. 2	Hasil <i>Preprocessing</i> dan Pelabelan Data .....	38
Gambar 4. 3	Total Label .....	39
Gambar 4. 4	Null Value .....	39
Gambar 4. 5	Proses Penggabungan Data <i>Train</i> .....	40
Gambar 4. 6	Total Parameter .....	43
Gambar 4. 7	Proses Epoch .....	43
Gambar 4. 8	Akurasi Train dan Validation (a), Loss Train dan Validation.....	44
Gambar 4. 9	Proses Penggabungan Data <i>Test</i> .....	45
Gambar 4. 10	Total Parameter .....	47
Gambar 4. 11	Proses Epoch .....	47
Gambar 4. 12	Akurasi Train dan Validation (a), Loss Train dan Validation.....	48
Gambar 4. 13	Proses Penggabungan Data <i>Train</i> .....	48
Gambar 4. 14	Proses Penggabungan Data <i>Test</i> .....	49
Gambar 4. 15	Total Parameter .....	51
Gambar 4. 16	Proses Epoch .....	52
Gambar 4. 17	Akurasi Train dan Validation (a), Loss Train dan Validation.....	52
Gambar 4. 18	<i>Confusion Matrix</i> .....	53

Gambar 4. 19 Hasil Perhitungan <i>Confusion Matrix</i> .....	53
Gambar 4. 20 Hasil Evaluasi Label Negative .....	54
Gambar 4. 21 Hasil Evaluasi Label Positive.....	54
Gambar 4. 22 <i>Confusion Matrix</i> .....	55
Gambar 4. 23 Hasil Perhitungan <i>Confusion Matrix</i> .....	55
Gambar 4. 24 Hasil Evaluasi Label Negative .....	56
Gambar 4. 25 Hasil Evaluasi Label Positive.....	56
Gambar 4. 26 <i>Confusion Matrix</i> .....	57
Gambar 4. 27 Hasil Perhitungan <i>Confusion Matrix</i> .....	57
Gambar 4. 28 Hasil Evaluasi Label <i>Negative</i> .....	58
Gambar 4. 29 Hasil Evaluasi Label <i>Positive</i> .....	58
Gambar 4. 30 Proses Penggabungan Data <i>Train</i> .....	59
Gambar 4. 31 Proses Penggabungan Data <i>Test</i> .....	60
Gambar 4. 32 Total Parameter.....	60
Gambar 4. 33 Proses <i>Epoch</i> .....	61
Gambar 4. 34 Akurasi <i>Train</i> dan <i>Validation</i> (a), <i>Loss Train</i> dan <i>Validation</i> .....	61
Gambar 4. 35 Proses Penggabungan Data <i>Train</i> .....	62
Gambar 4. 36 Proses Penggabungan Data <i>Test</i> .....	62
Gambar 4. 37 Total Parameter .....	63
Gambar 4. 38 Proses <i>Epoch</i> .....	63
Gambar 4. 39 Akurasi <i>Train</i> dan <i>Validation</i> (a), <i>Loss Train</i> dan <i>Validation</i> .....	64
Gambar 4. 40 Proses Penggabungan Data <i>Train</i> .....	64
Gambar 4. 41 Proses Penggabungan Data <i>Test</i> .....	65



Gambar 4. 42 Total Parameter .....	65
Gambar 4. 43 Proses <i>Epoch</i> .....	66
Gambar 4. 44 Akurasi <i>Train</i> dan <i>Validation</i> (a), <i>Loss Train</i> dan <i>Validation</i> .....	66
Gambar 4. 45 <i>Confusion Matrix</i> .....	67
Gambar 4. 46 Hasil Perhitungan <i>Confusion Matrix</i> .....	68
Gambar 4. 47 Hasil Evaluasi Label Negative .....	68
Gambar 4. 48 Hasil Evaluasi Label Positive.....	68
Gambar 4. 49 <i>Confusion Matrix</i> .....	69
Gambar 4. 50 Hasil Perhitungan <i>Confusion Matrix</i> .....	70
Gambar 4. 51 Hasil Evaluasi Label Negative .....	70
Gambar 4. 52 Hasil Evaluasi Label Positive.....	70
Gambar 4. 53 <i>Confusion Matrix</i> .....	71
Gambar 4. 54 Hasil Perhitungan <i>Confusion Matrix</i> .....	71
Gambar 4. 55 Hasil Evaluasi Label Negative .....	72
Gambar 4. 56 Hasil Evaluasi Label Positive.....	72



## INTISARI

Female daily network merupakan salah satu platform kecantikan yang memiliki beberapa layanan kecantikan seperti layanan kategori skincare. Selain itu, platform ini juga terdapat beberapa fitur seperti review, blog dan forum. Pada fitur review terdapat informasi pengalaman pengguna dan rating dari suatu produk yang telah digunakan sebelumnya, seperti pembersih wajah, pelembap tabir surya, serum, toner, facial mask dan lain sebagainya. Dalam memilih sebuah produk, fitur review ini sangatlah bermanfaat bagi konsumen yang akan menggunakan produk tersebut. Hal ini penting untuk mengetahui keputusan yang diambil tidak mendapat tanggapan atau pengalaman negatif dari konsumen sebelumnya, maka dari itu diperlukan analisis sentimen guna mengetahui persepsi pengalaman konsumen terhadap produk skincare tersebut.

Penelitian ini akan melakukan analisis sentimen *review* produk skincare *female daily* menggunakan LSTM dengan menggunakan *word embedding* Word2Vec. Pada penelitian sebelumnya, metode LSTM mampu menghasilkan akurasi yang lebih baik dibandingkan dengan metode lainnya. Penggunaan Word2Vec dalam penelitian ini dikarenakan Word2Vec mampu merepresentasi makna yang lebih baik dari keterbatasan informasi pada teks pendek yang didapat. Selain itu, sebagai pembandingan, penelitian akan menguji keakuratan pada dataset yang sama menggunakan LSTM tanpa menggunakan *word embedding* Word2Vec.

Berdasarkan proses implementasi dan pembahasan yang telah dilakukan diatas, dapat disimpulkan bahwa hasil pengujian yang telah dilakukan dengan menggunakan *confusion matrix* didapatkan nilai akurasi dari metode algoritma LSTM menggunakan *word2vec* dengan data training sebesar 90% dan testing sebesar 10% didapatkan nilai akurasi sebesar 75%, pada model LSTM dengan data *training* sebesar 80% dan *testing* sebesar 20% didapatkan nilai akurasi sebesar 72% dan pada pada model LSTM dengan data *training* sebesar 70% dan *testing* sebesar 30% didapatkan nilai akurasi sebesar 71%.

Kata kunci: Analisis Sentimen, Skincare, LSTM, Word2Vec

## **ABSTRACT**

*Female Daily Network is a beauty platform that offers various beauty services, including skincare. It also features review, blog, and forum sections. The review section provides user experiences and ratings for skincare products such as facial cleansers, sunscreen moisturizers, serums, toners, facial masks, and more. This review feature is highly beneficial for consumers in selecting products, as it helps them avoid negative experiences or feedback from previous consumers. Therefore, sentiment analysis is necessary to understand consumers' perceptions and experiences with skincare products.*

*This study aims to analyze the sentiment of skincare product reviews on Female Daily using LSTM with Word2Vec word embedding. Previous research has shown that LSTM outperforms other methods in terms of accuracy. Word2Vec is utilized in this study because it can better represent the meaning of limited information in short texts. Additionally, as a comparison, the study evaluates the accuracy on the same dataset using LSTM without Word2Vec word embedding.*

*Based on the implementation and discussion process that has been carried out above, it can be concluded that the test results that have been carried out using the confusion matrix obtained the accuracy value of the LSTM algorithm method using word2vec with training data of 90% and testing of 10% obtained an accuracy value of 75%, on the LSTM model with training data of 80% and testing of 20% obtained an accuracy value of 72% and on the LSTM model with training data of 70% and testing of 30% obtained an accuracy value of 71%.*

*Keyword: sentiment analysis, skincare, LSTM, Word2Vec*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

*Skincare* merupakan serangkaian perawatan kulit yang dapat merawat kesehatan serta kecantikan kulit. Beberapa *brand* produk *skincare* seperti pembersih wajah, pelembap tabir surya, dan produk *skincare* lainnya sudah beredar di pasaran (Ikhsania, 2020). Semua produk *skincare* dari berbagai *brand* tersebut memiliki kualitas yang baik sesuai dengan aturan yang telah ditetapkan oleh Badan Pengawas Obat dan Makanan RI, sehingga keamanan bahan produk *skincare* yang terkandung tidak perlu dikhawatirkan oleh para konsumen (Astuti & Astuti, 2022). Selain itu, peran *skincare* untuk konsumen khususnya perempuan sangatlah penting, hal ini dikarenakan perempuan sangat memperhatikan penampilan dan wajah yang bersih sehingga banyak perempuan yang mempelajari dan memahami pengetahuan terkait jenis kulitnya dan bahan kandungan *skincare* sebelum melakukan perawatan kulit (Pratiwi, et al., 2021).

*Female daily network* merupakan salah satu *platform* kecantikan yang memiliki beberapa layanan kecantikan seperti layanan kategori *skincare*. Selain itu, *platform* ini juga terdapat beberapa fitur seperti *review*, *blog* dan *forum* (Female Daily, 2022). Pada fitur *review* terdapat informasi pengalaman pengguna dan *rating* dari suatu produk yang telah digunakan sebelumnya, seperti pembersih wajah, pelembap tabir surya, serum, *toner*, *facial mask* dan lain sebagainya. Dalam memilih sebuah produk, fitur *review* ini sangatlah bermanfaat bagi konsumen yang

akan menggunakan produk tersebut. Namun *review* dan *rating* yang diberikan terkadang tidak benar-benar sesuai dengan kenyataan. Banyak pelaku penjual yang membuat *review* dan *rating* palsu agar produknya selalu mendapat penilaian positif (Darmawan & Laily, 2022). Permasalahan lainnya adalah banyaknya ulasan dalam suatu produk tidak memungkinkan membaca ulasan secara keseluruhan satu persatu hal ini pastinya memakan waktu (Fritama, Ramadhan, & Komara, 2023). Oleh sebab itu diperlukan sebuah analisis sentimen guna mengetahui persepsi pengalaman konsumen terhadap produk *skincare* tersebut sehingga konsumen mengetahui keputusan yang diambil berdasarkan *review* yang telah diberikan oleh konsumen sebelumnya.

*Female Daily Network* memiliki banyak *review* produk yang dapat dijadikan data penelitian, seperti penelitian yang dilakukan oleh (Hidayat & Handayani, 2022). Penelitian tersebut menerapkan ID-CNN untuk analisis sentiment *review* produk di *female daily review*. Hasil penelitian ini yaitu dengan sebanyak 30 percobaan, didapatkan model terbaik dalam menganalisis sentimen dengan akurasi sebesar 80.22%, namun penggunaan jumlah dataset baik data positif, negatif, dan netral dipenelitian ini tidak seimbang hal ini akan mengakibatkan *overfitting* pada hasil.

Penelitian analisis sentiment untuk ulasan produk *toner* di *female daily* dilakukan oleh (Nabila, 2022) menggunakan metode *Naïve Bayes Classifier* dan *Support Vector Machine*. Penelitian ini menunjukkan bahwa algoritma *Support Vector Machine* lebih baik daripada *Naïve Bayes*, namun metode yang digunakan



masih perlu dikembangkan dengan tujuan untuk meningkatkan akurasi hasilnya misalnya menggunakan *word embeddings*.

Analisis sentimen merupakan bidang studi yang mempelajari terkait opini, sentimen, evaluasi, sikap dan emosi *reviewer* yang ditumpahkan kedalam bahasa tulisan. Sentimen tersebut dibagi menjadi 2 jenis yaitu sentimen positif dan sentimen negatif (Pratiwi, et al., 2021). Dari banyaknya metode analisis sentimen, dalam penelitian ini menggunakan metode LSTM. Penggunaan metode LSTM dikarenakan memiliki akurasi yang baik untuk data berupa teks, serta LSTM merupakan pengembangan dari metode *deep learning* RNN yang memiliki kelebihan mampu memproses data yang relatif panjang (*long-term dependency*) (Widayat, 2021).

Penelitian yang membahas penggunaan metode LSTM dilakukan oleh (Ihsan, Negara, & Agustian, 2022) yang melakukan klasifikasi sentiment vaksin Covid-19 di Twitter dengan menggunakan 3 kelas sentiment yaitu positif, netral dan negative. Penelitian tersebut menggunakan metode Word2Vec sebagai input, untuk menguji penggunaan pretrained model Bahasa Indonesia dari korpus Wikipedia dan dilatih pada dataset yang digunakan. Data yang digunakan 2563 data latih, 778 data validasi dan 400 data uji dengan 1802 data netral, 1066 data negatif dan 566 data positif. Hasil terbaik dari berbagai proses optimasi parameter memberikan nilai F1-Score 54% dengan akurasi 66%. Hasil dari penelitian tersebut menghasilkan model yang dapat mengklasifikasi sentimen dengan kalimat yang baru.

Penelitian lain yang melakukan analisis sentimen dilakukan oleh (Fadly, Marlina, Kurniawan, Zakaria, & Farahnasihah, 2022) terhadap produk perawatan kulit alami menggunakan 5 metode klasifikasi yaitu *naïve bayes*, KNN, SVM, *decision tree* dan *deep learning*. Dataset yang digunakan didapatkan dari *web Sephora.com* dengan jumlah 10.000 data. Hasil penelitian menunjukkan bahwa *Deep Learning* dan *Decision Tree* merupakan pengklasifikasi dalam sentimen analisis dengan akurasi hampir 80% dan pengukuran F1 60%.

Analisis sentiment pada produk kecantikan dilakukan oleh (Astuti & Astuti, 2022) terhadap review produk *skincare* pada *female daily* menggunakan metode *naïve bayes* berbasis *Particle Swarm Optimization* (PSO). Metode PSO digunakan untuk meningkatkan akurasi pengklasifikasi *Naïve Bayes*. Dataset yang digunakan sebanyak 800 data *review* dan diujikan menggunakan *10-Fold Cross Validation*. Hasil penelitian menunjukkan peningkatan akurasi dari 77.96% menjadi 79.85%.

Penelitian analisis sentimen oleh (Widayat, 2021) dilakukan terhadap *review movie* menggunakan metode *Word2Vec* dan metode *LSTM*. Dataset yang digunakan adalah dataset *movie review* yang terdiri dari 25.000 dokumen *review*, dengan panjang rata-rata per *review* adalah 233 kata. Penelitian menggunakan metode *CBOV* dan *Skip-Gram* pada *word2vec* untuk membentuk vektor representasi dari setiap kata (*word vector*) di *corpus*. Ukuran dimensi *word vector* yang digunakan yaitu dimensi 50, 60, 100, 150, 200 dan 500, untuk mengetahui pengaruhnya terhadap akurasi yang dihasilkan. Akurasi terbaik diperoleh pada ukuran dimensi *word vector* 100 sebesar 88.17% dan akurasi terendah sebesar 85.86% pada ukuran dimensi *word vector* 500.



Penelitian analisis sentiment terhadap ulasan TripAdvisor menggunakan metode LSTM pernah dilakukan oleh (Nurvania, Jondri, & Lhaksamana, 2021). Penelitian bertujuan untuk mengklasifikasikan ulasan pengunjung tentang pengaruh COVID-19 terhadap tempat wisata di Bali dari Tripadvisor. Setiap teks pada ulasan akan divektorisasi dengan word2vec, kemudian dilakukan analisis sentiment pada metode LSTM. Hasil pengujian pada model yang dibangun didapatkan nilai akurasi sebesar 71,67%.

Penelitian analisis sentiment pada review skincare female daily dilakukan oleh (Pratiwi, et al., 2021) menggunakan metode *Support Vector Machine* (SVM). Dataset penelitian didapatkan dari *website female daily*. Hasil akurasi menggunakan dataset 80% data *training* dan 20% data *testing* mendapatkan akurasi sebesar 87% dengan *recall* sebesar 90%, *precision* sebesar 84,90%, dan *f1 score* sebesar 87,37%.

Penelitian analisis ulasan pada hotel Arab berdasarkan LSTM dilakukan oleh (Nejjari & Meziane, 2020). Ulasan tersebut ditulis menggunakan Bahasa Arab Standar Modern (MSA) yang diimplementasikan kedalam *Latent Semantic Analysis* (LSA) dan *Chi-Square*. Hasil penelitian menunjukkan nilai akurasi 83,6% pada metode LSA dan Chi-Square dan 92% pada Model klasifikasi LSTM.

Penelitian deteksi infeksi demam berdarah/flu pada Tweet menggunakan LSTM dan word embedding dilakukan oleh (Amin, et al., 2020). Metode *word embedding* yang digunakan yaitu Word2Vec dengan Skip-gram (SG) dan Word2Vec dengan *Continuous-bag-of-words* (CBOW). Hasil penelitian menunjukkan bahwa LSTM Word2Vec dengan CBOW lebih baik dibandingkan

dengan LSTM dengan teknik penyisipan fitur Word2Vec SG. Nilai akurasi yang didapatkan yaitu 94% sehingga LSTM berkinerja lebih baik daripada metode terkemuka lainnya dalam mendeteksi orang yang terinfeksi penyakit di tweet.

Perbandingan metode *word embeddings* dilakukan oleh (Kang, Chen, Chandrasekaran, & Kan, 2017). Metode *word embeddings* yang digunakan yaitu word2vec, Glove, Collobert & Weston. Hasil penelitian menunjukkan bahwa metode word2vec menghasilkan akurasi lebih baik dibandingkan metode lainnya yang digunakan. Selain itu, perbandingan metode *word embeddings* juga dilakukan oleh (Hendrawan, 2022) untuk klasifikasi teks sentimen masyarakat terhadap produk lokal di Indonesia. Metode yang dibandingkan yaitu Word2Vec dan TF-IDF. Dalam penelitian tersebut membandingkan penggunaan Word2Vec + XGBoost menghasilkan F1 Score lebih tinggi yaitu sebesar 0.941 diikuti dengan TF-IDF + XGBoost menghasilkan 0.940. Dapat disimpulkan bahwa klasifikasi menggunakan Word2Vec dalam merepresentasikan kata menjadi vektor lebih baik dibandingkan dengan TF-IDF. Hal ini dikarenakan Word2Vec mampu mengolah kesamaan kata dengan optimal.

Penelitian analisis sentimen pada produk kosmetik menggunakan metode *naïve bayes* dilakukan oleh (Ardian & Kosasi, 2019). Dalam penelitian tersebut untuk meningkatkan kinerja dari metode *naïve bayes* menggunakan TF-IDF. Hasil dari penelitian yaitu mengembangkan sistem yang dapat mengklasifikasikan *review* menjadi positif atau negatif menggunakan metode *naïve bayes* dengan mendapatkan akurasi pada *confusion matrix* sebesar 69% - 82%.

Penelitian lain yang menggunakan metode LSTM dilakukan oleh (Nurrohmat & SN, 2019) untuk analisis review pada novel. Review yang digunakan akan diklasifikasi menjadi 3 jenis yaitu positif, netral dan negatif. Dataset yang digunakan didapatkan dari situs goodreads.com. Dalam penelitian ini membandingkan hasil akurasi metode LSTM dengan naïve bayes untuk mengetahui akurasi terbaik dalam analisis *review* novel. Hasil penelitian menunjukkan bahwa metode *Long Short-Term Memory* memiliki hasil akurasi yang lebih baik dibandingkan dengan metode *naïve bayes* dengan nilai akurasi 72.85%, *precision* 73%, *recall* 72%, dan *f-measure* 72% dibandingkan dengan hasil akurasi metode Naïve Bayes dengan nilai akurasi 67.88%, *precision* 69%, *recall* 68%, dan *f-measure* 68%.

Penelitian yang menggunakan metode *word embedding* Word2Vec dilakukan oleh (Pane & Ramdan, 2022) untuk analisis sentiment masyarakat terhadap kebijakan PPKM menggunakan data twitter. Penelitian menghasilkan nilai akurasi sebesar 0,917. Penggunaan *word embedding* secara spesifik apabila diterapkan dalam penelitian tersebut, akan mendapatkan hasil yang berbeda, maka dari itu dalam penelitian ini menggunakan *word embedding* Word2Vec.

Dari uraian diatas, maka dalam penelitian ini akan melakukan analisis sentimen review produk skincare female daily menggunakan LSTM. Alasan menggunakan tidak menggunakan metode seperti naïve bayes, decision tree karena metode-metode tersebut masih tradisional yang belum memenuhi makna konteks (Siswa & Prihandoko, 2018). Sehingga pada penelitian ini menggunakan LSTM sebagai algoritma modern yang dapat menangkap makna konteks yang dapat

membedakan makna yang layak dan tidak sehingga dapat mencapai tujuan dari penelitian ini (Arissinta, Sulistiyawati, & Kurnianto, 2022). Selain itu, penggunaan LSTM didukung oleh penelitian sebelumnya, metode LSTM mampu menghasilkan akurasi yang lebih baik dibandingkan dengan metode lainnya. Namun disisi lain, terdapat penelitian yang dilakukan (Ihsan, Negara, & Agustian, 2022) dengan menerapkan metode LSTM masih menghasilkan akurasi yang rendah yaitu 66%. Oleh karena itu, dalam penelitian ini akan menguji seberapa besar tingkat keakuratan metode LSTM untuk analisis sentimen pada *review* produk *skincare* dengan menggunakan word embedding Word2Vec. Penggunaan Word2Vec dalam penelitian ini dikarenakan Word2Vec mampu merepresentasi makna yang lebih baik dari keterbatasan informasi pada teks pendek yang didapat (Rahman & Sari, 2021). Selain itu, sebagai pembanding, penelitian pada dataset yang sama menggunakan LSTM tanpa menggunakan word embedding Word2Vec, karena tanpa menggunakan *Word2Vec* kata hanya akan disimpan dengan ID unik tanpa arti dan memungkinkan terjadinya *data sparsity* atau banyaknya data kosong (Satriyo, Yuliani, & Nisa, 2021). Oleh sebab itu, pada penelitian ini akan mengetahui keakuratan menggunakan LSTM menggunakan *Word2Vec* dan tanpa menggunakan *Word2Vec*. Sehingga dengan dilakukannya analisis sentimen dapat digunakan untuk mengetahui persepsi pengalaman konsumen terhadap produk *skincare* tersebut.



## 1.2. Rumusan Masalah

Dari latar belakang yang telah dijelaskan, maka terdapat rumusan masalah dalam penelitian ini yaitu:

- a. Bagaimana membangun algoritma analisis sentiment pada produk skincare menggunakan produk review?
- b. Bagaimana hasil sentimen pengguna terhadap review produk skincare female daily network?
- c. Berapa nilai keakuratan yang dihasilkan dari metode LSTM dengan word embedding Word2Vec ataupun tidak menggunakan word embedding Word2Vec dalam analisis sentimen review produk skincare pada website female daily network?

## 1.3. Batasan Masalah

Batasan masalah dalam penelitian ini yaitu sebagai berikut:

- a. Dataset yang digunakan berupa *review* produk *skincare* dari website *female daily network*.
- b. *Review* produk yang akan dijadikan dataset adalah 5 buah produk *skincare* yang terdapat pada website *female daily network* yaitu *pond's vitamin micellar water brightening rose*, *pond's bright beauty triple glow serum*, *citra natural glowing white uv hand and body lotion*, *citra fresh glow multifunction gel aloe bright UV* dan *azarine cosmetics c white night moisturizer*.
- c. Proses pelabelan dilakukan secara manual atau melibatkan pakar ahli.
- d. Platform penelitian menggunakan Jupyter Notebook atau Google Colaboratory.

- e. Sentimen analisis diklasifikasikan menjadi 3, yaitu positif, negatif dan netral.
- f. Komposisi dataset yang digunakan untuk menghasilkan performa yang tinggi dengan menggunakan perbandingan 70% data *training* dan 30% data *testing*, 80% data *training* dan 20% sebagai data *testing* dan 90% data *training* 10% sebagai data *testing*.
- g. Metode yang digunakan untuk melakukan sentimen analisis adalah metode LSTM dengan menggunakan *word embedding* Word2Vec.
- h. *Output* dari penelitian ini adalah tingkat keakuratan algoritma LSTM menggunakan *word embedding* Word2Vec ataupun tidak menggunakan *word embedding* Word2Vec

#### 1.4. Tujuan Penelitian

Dalam penelitian ini terdapat tujuan penelitian yaitu sebagai berikut:

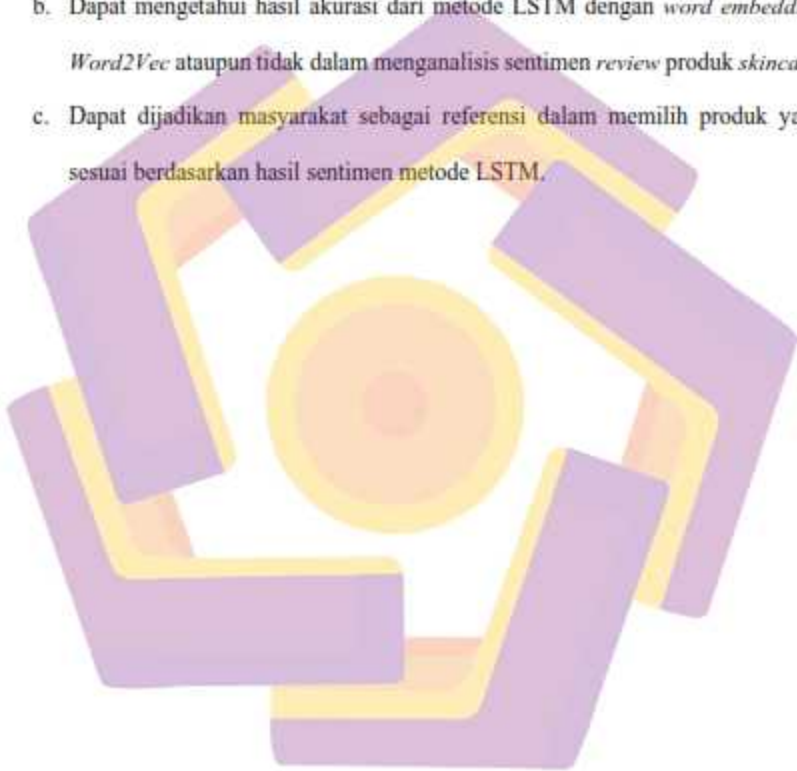
- a. Membangun algoritma analisis sentiment pada produk skincare menggunakan produk review.
- b. Mengetahui hasil sentimen pengguna terhadap review produk skincare female daily network.
- c. Mengetahui nilai keakuratan yang dihasilkan dari metode LSTM dengan word embedding Word2Vec ataupun tidak menggunakan word embedding Word2Vec dalam analisis sentimen review produk skincare pada website female daily network.



### 1.5. Manfaat Penelitian

Manfaat penelitian yaitu sebagai berikut:

- a. Dapat dijadikan bahan referensi penelitian selanjutnya dalam melakukan analisis sentimen *review* produk *skincare* pada *female daily network*.
- b. Dapat mengetahui hasil akurasi dari metode LSTM dengan *word embedding* *Word2Vec* ataupun tidak dalam menganalisis sentimen *review* produk *skincare*.
- c. Dapat dijadikan masyarakat sebagai referensi dalam memilih produk yang sesuai berdasarkan hasil sentimen metode LSTM.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Penelitian yang membahas penggunaan metode LSTM dilakukan oleh (Ihsan, Negara, & Agustian, 2022) yang melakukan klasifikasi sentiment vaksin Covid-19 di Twitter dengan menggunakan 3 kelas sentiment yaitu positif, netral dan negative. Penelitian tersebut menggunakan metode Word2Vec sebagai input, untuk menguji penggunaan pretrained model Bahasa Indonesia dari korpus Wikipedia dan dilatih pada dataset yang digunakan. Data yang digunakan 2563 data latih, 778 data validasi dan 400 data uji dengan 1802 data netral, 1066 data negatif dan 566 data positif. Hasil terbaik dari berbagai proses optimasi parameter memberikan nilai F1-Score 54% dengan akurasi 66%. Hasil dari penelitian tersebut menghasilkan model yang dapat mengklasifikasi sentimen dengan kalimat yang baru.

Penelitian lain yang melakukan analisis sentimen dilakukan oleh (Fadly, Marlina, Kurniawan, Zakaria, & Farahnasifah, 2022) terhadap produk perawatan kulit alami menggunakan 5 metode klasifikasi yaitu *naïve bayes*, KNN, SVM, *decision tree* dan *deep learning*. Dataset yang digunakan didapatkan dari *web Sephora.com* dengan jumlah 10.000 data. Hasil penelitian menunjukkan bahwa *Deep Learning* dan *Decision Tree* merupakan pengklasifikasi dalam sentimen analisis dengan akurasi hampir 80% dan pengukuran F1 60%.

Analisis sentiment pada produk kecantikan dilakukan oleh (Astuti & Astuti, 2022) terhadap review produk skincare pada female daily menggunakan metode *naïve bayes* berbasis *Particle Swarm Optimization* (PSO). Metode PSO digunakan untuk meningkatkan akurasi pengklasifikasi *Naïve Bayes*. Dataset yang digunakan sebanyak 800 data *review* dan diujikan menggunakan *10-Fold Cross Validation*. Hasil penelitian menunjukkan peningkatan akurasi dari 77.96% menjadi 79.85%.

Penelitian analisis sentimen oleh (Widayat, 2021) dilakukan terhadap review movie menggunakan metode Word2Vec dan metode LSTM. Dataset yang digunakan adalah dataset movie review yang terdiri dari 25.000 dokumen review, dengan panjang rata-rata per review adalah 233 kata. Penelitian menggunakan metode CBOV dan Skip-Gram pada word2vec untuk membentuk vektor representasi dari setiap kata (word vector) di corpus. Ukuran dimensi word vector yang digunakan yaitu dimensi 50, 60, 100, 150, 200 dan 500, untuk mengetahui pengaruhnya terhadap akurasi yang dihasilkan. Akurasi terbaik diperoleh pada ukuran dimensi word vector 100 sebesar 88.17% dan akurasi terendah sebesar 85.86% pada ukuran dimensi word vector 500.

Penelitian analisis sentiment terhadap ulasan TripAdvisor menggunakan metode LSTM pernah dilakukan oleh (Nurvania, Jondri, & Lhaksamana, 2021). Penelitian bertujuan untuk mengklasifikasikan ulasan pengunjung tentang pengaruh COVID-19 terhadap tempat wisata di Bali dari Tripadvisor. Setiap teks pada ulasan akan divektorisasi dengan word2vec, kemudian dilakukan analisis sentiment pada metode LSTM. Hasil pengujian pada model yang dibangun didapatkan nilai akurasi sebesar 71,67%.

Penelitian analisis sentiment pada review skincare female daily dilakukan oleh (Pratiwi, et al., 2021) menggunakan metode *Support Vector Machine* (SVM). Dataset penelitian didapatkan dari *website female daily*. Hasil akurasi menggunakan dataset 80% data *training* dan 20% data *testing* mendapatkan akurasi sebesar 87% dengan *recall* sebesar 90%, *precision* sebesar 84,90%, dan *f1 score* sebesar 87,37%.

Penelitian analisis ulasan pada hotel Arab berdasarkan LSTM dilakukan oleh (Nejjari & Meziane, 2020). Ulasan tersebut ditulis menggunakan Bahasa Arab Standar Modern (MSA) yang diimplementasikan kedalam *Latent Semantic Analysis* (LSA) dan *Chi-Square*. Hasil penelitian menunjukkan nilai akurasi 83,6% pada metode LSA dan Chi-Square dan 92% pada Model klasifikasi LSTM.

Penelitian deteksi infeksi demam berdarah/flu pada Tweet menggunakan LSTM dan word embedding dilakukan oleh (Amin, et al., 2020). Metode *word embedding* yang digunakan yaitu Word2Vec dengan Skip-gram (SG) dan Word2Vec dengan *Continuous-bag-of-words* (CBOW). Hasil penelitian menunjukkan bahwa LSTM Word2Vec dengan CBOW lebih baik dibandingkan dengan LSTM dengan teknik penyisipan fitur Word2Vec SG. Nilai akurasi yang didapatkan yaitu 94% sehingga LSTM berkinerja lebih baik daripada metode terkemuka lainnya dalam mendeteksi orang yang terinfeksi penyakit di tweet.

Penelitian analisis sentimen pada produk kosmetik menggunakan metode *naïve bayes* dilakukan oleh (Ardian & Kosasi, 2019). Dalam penelitian tersebut untuk meningkatkan kinerja dari metode *naïve bayes* menggunakan TF-IDF. Hasil dari penelitian yaitu mengembangkan sistem yang dapat mengklasifikasikan *review*

menjadi positif atau negatif menggunakan metode *naïve bayes* dengan mendapatkan akurasi pada *confusion matrix* sebesar 69% - 82%.

Penelitian lain yang menggunakan metode LSTM dilakukan oleh (Nurrohmat & SN, 2019) untuk analisis review pada novel. Review yang digunakan akan diklasifikasi menjadi 3 jenis yaitu positif, netral dan negatif. Dataset yang digunakan didapatkan dari situs *goodreads.com*. Dalam penelitian ini membandingkan hasil akurasi metode LSTM dengan *naïve bayes* untuk mengetahui akurasi terbaik dalam analisis *review* novel. Hasil penelitian menunjukkan bahwa metode *Long Short-Term Memory* memiliki hasil akurasi yang lebih baik dibandingkan dengan metode *naïve bayes* dengan nilai akurasi 72.85%, *precision* 73%, *recall* 72%, dan *f-measure* 72% dibandingkan dengan hasil akurasi metode *Naïve Bayes* dengan nilai akurasi 67.88%, *precision* 69%, *recall* 68%, dan *f-measure* 68%.

Penelitian analisis sentiment ulasan aplikasi Pospay dilakukan oleh (Putri & Ridwan, 2023). Permasalahan yang melatarbelakangi penelitian ini adalah banyaknya ulasan pengguna membuat semakin sulit dan lama waktu yang dibutuhkan bagi pengelola aplikasi untuk menyimpulkan informasi ulasan tersebut. Sehingga diperlukan analisis sentiment pada aplikasi Pospay menggunakan SVM. Penelitian ini menggunakan proporsi 90:10 yang menghasilkan akurasi model sebesar 95%, proporsi 80:20 menghasilkan akurasi sebesar 92%, dan 70:30 menghasilkan akurasi model sebesar 89%.



## 2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian Analisis Sentimen pada *Review* Produk *Skincare* Menggunakan *Word Embedding* dan Metode *Long Short-Term Memory (LSTM)*

No	Judul	Peneliti, Media Publikasi dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Analisis Sentimen Masyarakat Terhadap <i>Exchange Tokocrypto</i> Pada Twitter Menggunakan Metode LSTM	Green Arther Sandag dan Jacqueline Waworindeng, (2022)	Melakukan analisis sentimen masyarakat terhadap <i>exchange tokocrypto</i> pada <i>twitter</i> menggunakan metode LSTM.	Penggunaan model <i>Long Short-Term Memory (LSTM)</i> bekerja dengan baik dalam mengklasifikasikan jumlah <i>tweets</i> tentang <i>tokocrypto</i> yang ada, juga menggunakan <i>Lexicon Sentimen Bahasa Indonesia</i> untuk diambil kata positif, negatif, dan netral. Dari 5000 <i>tweets</i> yang ada mengenai keyword <i>tokocrypto</i> dalam bahasa Indonesia, ditemukan sebanyak 2022 kata positif, 1632 kata negatif, dan 1012 kata netral.	Untuk penelitian lebih lanjut di masa mendatang, dapat digunakan berbagai metode klasifikasi dan perbandingan terhadap metode lain, juga algoritma yang lain dalam menemukan algoritma dengan prediksi terbaik terhadap kata-kata positif, negatif, dan netral, serta mengoptimalkan efisiensi dan meminimalkan kesalahan dalam proses pemodelan atau <i>trainingsuatu data</i> .	Pada penelitian ini peneliti masih belum menggunakan <i>word embedding</i> , selain itu peneliti juga tidak menjelaskan terkait dengan detail dari hasil akurasi yang diperoleh. Sedangkan pada penelitian yang akan dilakukan peneliti akan menambahkan <i>word embedding Word2Vec</i> sebelum melakukan klasifikasi menggunakan LSTM serta melakukan evaluasi terhadap hasil dari klasifikasi yang diperoleh.



Tabel 2.1 (Lanjutan)

2	<p>Analisis Pendapat Masyarakat terhadap Berita Kesehatan Indonesia menggunakan Pemodelan Kalimat berbasis LSTM</p>	<p>Esther Irawati Setiawan, dkk., (2020)</p>	<p>melakukan analisis terhadap pendapat masyarakat terhadap berita kesehatan indonesia menggunakan pemodelan kalimat berbasis LSTM</p>	<p>hasil uji coba dengan dataset bahasa Indonesia bertopik kesehatan yang telah dibangun untuk makalah ini, model analisis <i>stance</i> yang diusulkan mampu meraih rata-rata nilai F1 71%, dengan rincian label mendukung 69%, menentang 70%, dan netral 74%.</p>	<p>Bagianbagian dari arsitektur LSTM ini masih dapat dimodifikasi dengan berbagai metode pembelajaran mendalam lainnya. Upaya lainnya adalah menggunakan model representasi kata lainnya seperti FastText untuk analisis.</p>	<p>Penelitian ini melakukan pemodelan kalimat dengan menggunakan metode LSTM terhadap Berita Kesehatan Indonesia dengan untuk makalah ini, model analisis <i>stance</i> yang diusulkan mampu meraih rata-rata nilai F1 71%, dengan rincian label mendukung 69%, menentang 70%, dan netral 74%. sedangkan penelitian yang akan dilakukan, peneliti akan melakukan analisis sentimen terhadap <i>review</i> produk <i>skincare</i> dengan menggunakan Word2Vec dan metode LSTM <i>Deep Learning</i> pada <i>review</i> produk <i>skincare</i> menggunakan <i>word embedding</i> dan metode Long Short-Term Memory (LSTM)</p>
---	---	--	--	---	---	--

Tabel 2.1 (Lanjutan)

3	<p>Analisis Sentimen Review Produk Skincare Dengan <i>Naïve Bayes Classifier</i> Berbasis <i>Particle Swarm Optimization (PSO)</i></p>	Tri Astuti, Yuli Astuti, (2022)	<p>Menganalisis sentimen menggunakan dataset <i>review</i> dari website <i>female daily</i>.</p>	<p>Nilai akurasi dari algoritma <i>Naïve Bayes</i> sebesar 77,96% dan <i>Naïve Bayes-PSO</i> sebesar 78,85% hasil validasi menggunakan metode 10 <i>k-fold cross validation</i>.</p>	<p>Terdapat perbedaan hasil dari analisis sentimen karena PSO bekerja mencari nilai parameter terbaik dengan cara beradaptasi terhadap <i>local best</i> dan beradaptasi terhadap letak partikel terbaik pada seluruh kelompok (<i>global best</i>).</p>	<p>Penelitian ini melakukan perbandingan antara <i>naïve bayes</i> dengan PSO dan tanpa PSO. Dari hasil penelitian ini <i>Naïve Bayes-PSO</i> memperoleh hasil akurasi sebesar 78,85%. sedangkan penelitian yang akan dilakukan akan menambahkan <i>word embedding Word2Vec</i> sebelum melakukan klasifikasi menggunakan LSTM.</p>
4	<p>Analisis Sentimen <i>Movie Review</i> menggunakan <i>Word2Vec</i> dan metode LSTM <i>Deep Learning</i></p>	Widi Widayat, (2021)	<p>Menganalisis <i>review</i> dari <i>movie</i> menggunakan <i>Word2Vec</i> dan metode LSTM.</p>	<p>Analisis <i>review movie</i> berhasil dilakukan dengan akurasi terbaik diperoleh pada ukuran dimensi <i>word vector</i> 100 sebesar 88,17% dan akurasi terendah sebesar 85,86% pada ukuran dimensi <i>word vector</i> 500.</p>	<p>Ukuran dimensi yang semakin besar tidak menjamin bisa menghasilkan akurasi yang lebih baik.</p>	<p>Penelitian ini melakukan analisis sentimen terhadap <i>Movie Review</i> menggunakan <i>Word2Vec</i> dan metode LSTM <i>Deep Learning</i> dengan akurasi terbaik diperoleh pada ukuran dimensi <i>word vector</i> 100 sebesar 88,17%. sedangkan penelitian yang akan dilakukan, peneliti akan melakukan analisis sentimen terhadap <i>review</i> produk <i>skincare</i> dengan menggunakan <i>Word2Vec</i> dan metode LSTM <i>Deep Learning</i> pada <i>review</i> produk <i>skincare</i> menggunakan <i>word embedding</i> dan metode Long Short-Term Memory (LSTM)</p>

Tabel 2.1 (Lanjutan)

5	Analisis Sentimen Pada Ulasan di TripAdvisor Menggunakan Metode <i>Long Short-Term Memory</i> (LSTM)	Jovita Nurvania, Jondri, Kemas Muslim Lhaksamana, (2021)	Mengklasifikasikan ulasan-pengunjung tentang pengaruh COVID-19 terhadap tempat wisata di Bali dari TripAdvisor menggunakan metode <i>Long Short-Term Memory</i> (LSTM)	Metode LSTM dengan Word2Vec dapat mengklasifikasi dokumen dengan nilai akurasi 71,67%.	Menggunakan dataset yang banyak untuk proses pelatihan data seperti jumlah dataset yang digunakan pada jurnal acuan. Selain itu, data yang digunakan harus seimbang dan sudah bersih atau melewati tahap <i>preprocessing</i> dengan baik dan benar.	Penelitian ini melakukan analisis sentimen terhadap Ulasan di TripAdvisor menggunakan Metode LSTM dengan Word2Vec dapat mengklasifikasi dokumen dengan nilai akurasi 71,67%. sedangkan penelitian yang akan dilakukan, peneliti analisis sentimen terhadap <i>review</i> produk <i>skincare</i> dengan menggunakan Word2Vec dan metode LSTM <i>Deep Learning</i> pada <i>review</i> produk <i>skincare</i> menggunakan <i>word embedding</i> dan metode <i>Long Short-Term Memory</i> (LSTM)
6	<i>Sentimen Analysis of Novel Review Using Long Short-Term Memory Method</i>	Muh Amin Nurrohmah, Azhari SN, (2019)	Melakukan pengklasifikasian terhadap <i>review novel</i> berbahasa Indonesia berdasarkan sentimen positif, netral dan negatif dengan menggunakan metode <i>Long Short-Term Memory</i> (LSTM)	Metode <i>Long Short-Term Memory</i> memiliki hasil akurasi yang lebih baik dibandingkan dengan metode <i>naïve bayes</i> dengan nilai akurasi 72,85%, dibandingkan dengan hasil akurasi metode <i>Naïve Bayes</i> dengan nilai akurasi 67,88%.	Penelitian ini masih terbatas pada sejumlah kecil data, yang seharusnya menjadi kasus penelitian di masa depan menggunakan jumlah data yang lebih besar.	Penelitian ini melakukan perbandingan antara metode LSTM dengan metode <i>naïve bayes</i> dalam melakukan analisis sentimen terhadap <i>Novel Review</i> . Berdasarkan hasil yang diperoleh Metode <i>Long Short-Term Memory</i> memiliki hasil akurasi yang lebih baik dibandingkan dengan metode <i>naïve bayes</i> . Sedangkan pada penelitian yang akan dilakukan peneliti hanya menggunakan metode <i>Long Short-Term Memory</i> (LSTM) dengan menggunakan <i>word embedding</i> .

Tabel 2.1 (Lanjutan)

7	Analisis Sentimen Twitter Terhadap Kebijakan Ppkm Di Tengah Pandemi Menggunakan Model LSTM	Adi Yahyadi, Fitri Latifah, (2022)	Untuk mengetahui pendapat pengguna twitter di Indonesia tentang penerapan PPKM dengan menggunakan metode <i>Long Short-Term Memory (LSTM)</i>	Hasil pengujian analisis sentimen tweet dengan menggunakan model LSTM dari algoritma RNN memiliki tingkat keefektifan yang cukup bagus dalam penelitian ini dengan tingkat akurasi sebesar 70% serta memiliki kriteria dengan nilai loss yang rendah	Untuk meningkatkan kinerja atau kemampuan yang lebih baik lagi penelitian lebih lanjut dapat menggunakan model yang lain sehingga dapat dilakukan training ulang atau mengganti model yang baru dalam melakukan analisis sentimen.	Pada penelitian ini peneliti masih belum menggunakan <i>word embedding</i> , selain itu peneliti juga tidak menjelaskan terkait dengan detail dari hasil akurasi yang diperoleh. Sedangkan pada penelitian yang akan dilakukan peneliti akan menambahkan <i>word embedding Word2Vec</i> sebelum melakukan klasifikasi menggunakan LSTM serta melakukan evaluasi terhadap hasil dari klasifikasi yang diperoleh.
8	Analisis Sentimen Pada Twitter Terhadap Program Kartu Pra Kerja Dengan <i>Recurrent Neural Network</i>	Rosit Samusi, Femi Dwi Astuti dan Indra Yatin Buryadi, (2021)	Untuk menganalisa <i>tweet</i> berbahasa Indonesia yang membicarakan mengenai program kartu prakerja yang ditandai dengan kata kunci prakerja	Penelitian menghasilkan sebuah sistem yang mampu melakukan klasifikasi sentimen (positif, netral dan negatif) terhadap sebuah <i>tweet</i> . Tingkat akurasi dari proses <i>training</i> yang didapat sebesar 95,66% serta tingkat akurasi dari proses testing sebesar 64,48%	Dataset kurang seimbang dimana dari 4122 dataset, 2460 termasuk label netral, 689 termasuk label positif dan 973 termasuk label negatif	Pada penelitian ini peneliti masih belum menggunakan <i>word embedding</i> , hasil tingkat akurasi yang diperoleh dari proses <i>training</i> yang didapat sebesar 95,66% serta tingkat akurasi dari proses testing sebesar 64,48%. Sedangkan pada penelitian yang akan dilakukan peneliti akan menambahkan <i>word embedding Word2Vec</i> sebelum melakukan klasifikasi menggunakan LSTM serta melakukan evaluasi terhadap hasil dari klasifikasi yang diperoleh.



Tabel 2.1 (Lanjutan)

9.	<i>Sentiment Analysis of US Airlines Tweets using LSTM/RNN</i>	R.Monika, S.Deivalakshm, B.Janet (2019)	Menyelidiki analisis sentimen menggunakan model <i>Recurrent Neural Network (RNN)</i> bersama dengan unit jaringan <i>Long Short-Term Memory (LSTM)</i> dapat menangani dependensi jangka panjang dengan memperkenalkan memori dalam model jaringan untuk prediksi dan visualisas	Makalah ini mengusulkan analisis sentimen menggunakan deep teknik pembelajaran dengan model jaringan LSTM. Kami menganalisis data tweet untuk industri penerbangan di media sosial yang menunjukkan kinerja yang lebih baik dalam rangkaian pelatihan.	Akurasi dapat ditingkatkan menggunakan Jaringan LSTM dua arah (Bi-LSTM)	Penelitian ini melakukan analisis sentimen terhadap <i>US Airlines</i> menggunakan Metode LSTM dengan menggunakan <i>word embedding</i> , sedangkan penelitian yang akan dilakukan, peneliti analisis sentimen terhadap <i>review</i> produk <i>skincare</i> dengan menggunakan Word2Vec dan metode LSTM <i>Deep Learning</i> pada <i>review</i> produk <i>skincare</i> menggunakan <i>word embedding</i> dan metode <i>Long Short-Term Memory (LSTM)</i>
10	<i>Sentimen Analysis of IMDb Movie Reviews Using Long Short-Term Memory</i>	Karmanova E.V., Dokolin A.S., Chernova E. V.	Untuk melakukan analisis sentimen terhadap Ulasan Film IMDb menggunakan metode <i>Long Short-Term Memory (LSTM)</i>	Kinerja klasifikasi dari <i>Long Short-Term Memory (LSTM)</i> menunjukkan akurasi klasifikasi terbaik sebesar 89,9%	Total 10k ulasan dipertimbangkan, 5k untuk sentimen positif dan 5k untuk sentimen negatif. Hasil penelitian telah menyimpulkan bahwa akurasi tertinggi yang dicapai oleh pendekatan yang dirancang adalah dari 89.9%.	Penelitian ini melakukan analisis sentimen terhadap Ulasan pada <i>IMDb Movie Reviews</i> menggunakan Metode LSTM dengan Doc2Vec dapat mengklasifikasi dokumen dengan nilai akurasi 71,67%. sedangkan penelitian yang akan dilakukan, peneliti analisis sentimen terhadap <i>review</i> produk <i>skincare</i> dengan menggunakan Word2Vec dan metode LSTM <i>Deep Learning</i> pada <i>review</i> produk <i>skincare</i> menggunakan <i>word embedding</i> dan metode <i>Long Short-Term Memory (LSTM)</i>



Tabel 2.1 (Lanjutan)

11	Analisis Sentimen Ulasan Aplikasi Pospay dengan Algoritma Support Vector Machine	Dea Safryda Putri, Taufik Ridwan	Untuk melakukan analisis sentiment terhadap ulasan aplikasi Pospay menggunakan SVM	Penelitian ini menggunakan proporsi 90:10 yang menghasilkan akurasi model sebesar 95%, proporsi 80:20 menghasilkan akurasi sebesar 92%, dan 70:30 menghasilkan akurasi model sebesar 89%.	Mengimplementasikan algoritma maupun metode yang berbeda dengan yang telah diterapkan dalam penelitian ini. Selain itu, model analisis sentimen ulasan aplikasi Pospay diharapkan dapat diimplementasikan ke dalam sebuah sistem.	Penelitian ini melakukan analisis sentimen terhadap Ulasan aplikasi Pospay menggunakan Metode SVM. sedangkan penelitian yang akan dilakukan, peneliti analisis sentimen terhadap review produk <i>skincare</i> dengan menggunakan Word2Vec dan metode LSTM <i>Deep Learning</i> pada review produk <i>skincare</i> menggunakan <i>word embedding</i> dan metode <i>Long Short-Term Memory (LSTM)</i>
----	--	----------------------------------	--	---	---	--

## 2.3. Landasan Teori

### 2.3.1 Analisis Sentimen

Analisis sentimen merupakan salah satu cabang dari penelitian *text mining* yang memiliki tujuan untuk menentukan persepsi publik terhadap topik pembahasan, kejadian maupun permasalahan serta mengklasifikasikan persepsi tersebut kedalam label positif atau negatif (Rachman & Pramana, 2020).

Analisis sentimen merupakan bidang studi yang mempelajari terkait opini, sentimen, evaluasi, sikap dan emosi reviewer yang ditumpahkan kedalam bahasa tulisan. Adapun cakupan pada analisis sentimen yaitu *emotion analysis, opinion extraction, sentimen mining, subjectivity analysis, affected analysis, emotion analysis, review mining*. Tujuan dari analisis sentimen yaitu cara agar komputer bisa mengenali tingkat kelas emosi positif atau negatif dari sampel teks. Terdapat polaritas sentimen fungsinya untuk mengenali ciri khusus dari teks tersebut. Sebuah teks secara umum diklasifikasikan ke polaritas sentimen positif dan sentimen negatif (Pratiwi, et al., 2021).

### 2.3.2 Preprocessing

*Text preprocessing* berfungsi untuk mengubah data teks yang tidak terstruktur atau sembarang menjadi data yang terstruktur. *Preprocessing* dilakukan untuk menghindari dataset yang kurang sempurna, terdapat *noise* pada dataset, data-data yang tidak konsisten dan mempercepat pemrosesan terhadap dokumen (Astuti & Astuti, 2022).

*Pre-processing* merupakan sebuah proses awal klasifikasi dokumen dengan tujuan menyiapkan data agar menjadi terstruktur. *Pre-processing* merupakan salah satu langkah yang penting dalam proses analisis sentimen. Tahapan-tahapan dalam *pre-processing* diantaranya ialah sebagai berikut (Musafar & Y. Yatini, 2021):

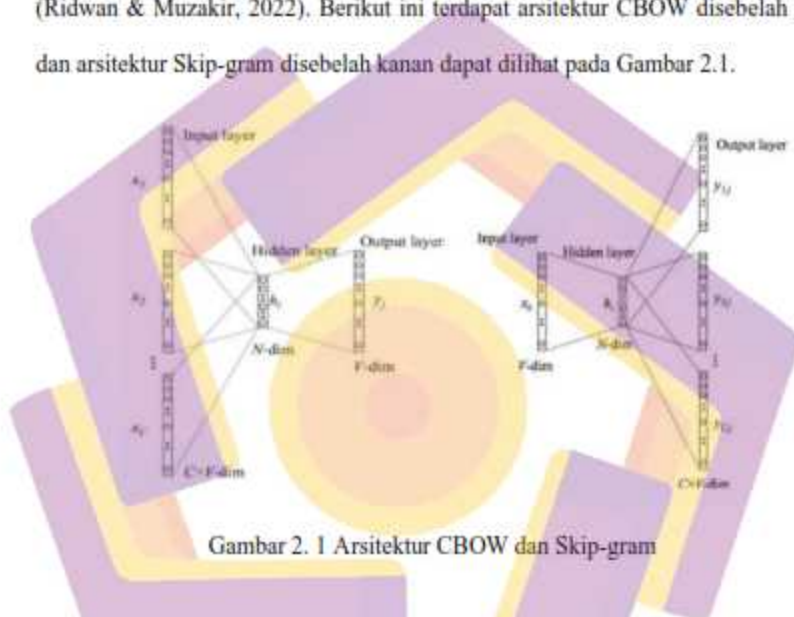
1. Tahapan *Case folding*, merupakan proses dimana pada sebuah teks akan dilakukan proses perubahan dari huruf besar menjadi huruf kecil dan menghilangkan seluruh tanda baca pada kalimat.
2. Tahapan *Tokenizing*, merupakan proses dimana setiap kata akan dipisahkan berdasarkan spasi yang ditemukan.
3. Tahapan *Filtering*, yaitu merupakan sebuah proses pembuangan/penyaringan kata-kata yang tidak penting dari hasil token.
4. Tahapan *Stemming/Lemmatization*, yaitu merupakan proses pengubahan kata berimbuhan menjadi sebuah kata dasar.

### 2.3.3 *Word2Vec*

*Word2Vec* merupakan sebuah *word embedding model* yang dapat digunakan untuk mengubah kata sehingga menjadi representasi sebuah vektor dengan panjang  $N$ , yang dimana vektor tersebut tidak hanya di representasikan secara sintaksis, tetapi kata juga diwakili secara semantik. *Word2Vec* bekerja dengan *neural network* yang dimana arsitekturnya hanya terdiri dari *layer input*, *projection (hidden layer)* dan *outoput* pada rancangan arsitekturnya (Ridwan & Muzakir, 2022).

Model *Word2Vec* terdiri dari model Skip-gram dan CBOW. Model Skip-gram lebih dikenal sebagai cara yang efisien untuk memeriksa seberapa besar

representasi vektor dalam teks yang tidak terstruktur. Arsitektur *word embedding* model Skip-gram bekerja dengan mencoba membuat prediksi pada rentang sesudah atau sebelum kata saat ini yang inputnya juga berasal dari kata saat ini, sedangkan model CBOW memprediksi kata-kata saat ini hanya berdasarkan konteks kata (Ridwan & Muzakir, 2022). Berikut ini terdapat arsitektur CBOW disebelah kiri dan arsitektur Skip-gram disebelah kanan dapat dilihat pada Gambar 2.1.



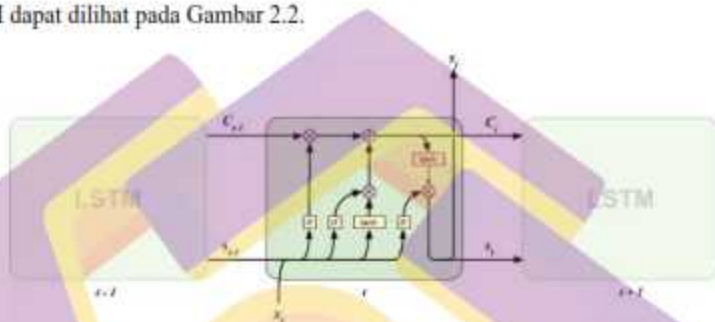
Gambar 2. 1 Arsitektur CBOW dan Skip-gram

#### 2.3.4 Metode *Long Short-Term Memory* (LSTM)

Metode *deep learning* LSTM (*Long Short-Term Memory*) dipilih karena beberapa pertimbangan, diantaranya adalah LSTM memiliki akurasi yang baik untuk data berupa teks serta LSTM merupakan pengembangan dari metode *deep learning* RNN yang memiliki kelebihan mampu memproses data yang relatif panjang (*long-term dependency*) (Widayat, 2021).

LSTM memiliki tiga jenis *gates* yaitu, *forget gate* (it), *input gate* (ft), dan *output gate* (ot). *Input gate* berfungsi untuk memutuskan nilai dari input untuk

diperbaharui pada *state memory*. *Output gate* berfungsi untuk memutuskan nilai dari input untuk diperbaharui pada *state memory*. *Output gate* berfungsi untuk memutuskan apakah yang dihasilkan *output* sesuai dengan input dan memori pada *cell* (Af'ida, Dairoh, Handayani, & Pratiwi, 2021). Berikut merupakan arsitektur LSTM dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Desain Algoritma *Long Short-Term Memory*

Pada gambar 2.2 diatas menjelaskan bagaimana alur kerja memory cells pada setiap *neurons* LSTM bekerja. Terdapat empat proses fungsi aktivasi pada setiap masukan pada *neurons* yang selanjutnya disebut sebagai gates units. Gates units tersebut ialah *forget gates*, *input gates*, *cell gates*, dan *output gates*. Pada *forget gates* informasi pada setiap data masukan diolah dan dipilih data mana saja yang akan disimpan atau dibuang pada *memory cells*. Fungsi aktivasi yang digunakan pada *forget gates* ini adalah fungsi aktivasi sigmoid. Dimana hasil keluarannya antara 0 dan 1. Jika keluarannya adalah 1 maka semua data akan disimpan dan sebaliknya jika keluarannya 0 maka semua data akan dibuang (Riyantoko, Fahrudin, Maulida, Hindrayani, & Safitr, 2020).

Lapisan tersembunyi LSTM terdiri dari beberapa proses untuk mengakomodasi lapisan *input*, lapisan *output*, proses sebelumnya dari keadaan



tersembunyi, dan keluaran dari keadaan tersembunyi. Kemampuan LSTM untuk mendeteksi aspek berurutan menyebabkan beberapa proses komputasi penting dalam keadaan tersembunyi. Setiap variabel memperoleh *output* karena proses pembelajaran berdasarkan beberapa perhitungan yang melibatkan beberapa aspek (Hanafi & Aboob, 2021).

$$i = \sigma(x_t U^i + S_{t-1} W^i) \quad (2.1)$$

$$f = \sigma(x_t U^f + S_{t-1} W^f) \quad (2.2)$$

$$o = \sigma(x_t U^o + S_{t-1} W^o) \quad (2.3)$$

$$g = \tanh(x_t U^g + S_{t-1} W^g) \quad (2.4)$$

$$c_t = c_t - Of + gOi \quad (2.5)$$

$$s_t = \tanh(c_t) - O_o g Oi \quad (2.6)$$

Dimana:

$i, f, o$  :  $i$  mewakili input,  $f$  mewakili *forget*, dan  $o$  mewakili gerbang *output*.

Semuanya memiliki persamaan yang sama dan hanya memiliki matriks parameter yang berbeda. Ini dikenal sebagai gerbang karena aturan sigmoid yang menentukan nilai sebagai 0 atau 1.

$g$  : Ini mewakili keadaan tersembunyi, yang dihitung berdasarkan input yang ada dan keadaan tersembunyi masa lalu.

$c_t$  : Ini mewakili memori internal dari keadaan tersembunyi. Ini adalah kombinasi dari memori sebelumnya  $c_{t-1}$  yang dikalikan dengan gerbang lupa dan keadaan tersembunyi baru  $g$  yang dikalikan dengan gerbang input.

$s_t$  : Ini mewakili memori keadaan tersembunyi. *Output* yang dihitung dari status tersembunyi pada HT dikalikan dengan output GA

### 2.3.5 Confussion Matrix

*Confussion matrix* adalah sebuah tabel atau matriks yang berisikan empat nilai yang merupakan pengukuran performa dari masalah klasifikasi yang telah dilakukan. Ada empat nilai atau point yang ada pada *confussion matrix* yaitu *True Positive*, *True Negative*, *False Positive* dan *False Negative* (Nurvania, Jondri, & Lhaksamana, 2021).

Tabel 2. 1 *Confussion Matrix*

		<i>True Values</i>	
		<i>Positive</i>	<i>Negative</i>
<i>Predictions</i>	<i>Positive</i>	<i>True Positive</i>	<i>False Positive</i>
	<i>Negative</i>	<i>False Negative</i>	<i>True Negative</i>

Keterangan

- True positive* (TP): Prediksi yang bernilai *positive* dan benar sesuai target.
- True negative* (TN): Prediksi yang bernilai *negative* dan benar sesuai target.
- False positive* (FP): Prediksi yang bernilai *positive* dan salah tidak sesuai target.
- False negative* (FN): Prediksi yang bernilai *negative* dan salah tidak sesuai target.

Perhitungan *confussion matrix* menghasilkan *accuracy*, *precision*, *recall* dan *f1-score*. Berikut merupakan persamaannya:

a. *Accuracy*

*Accuracy* adalah perhitungan seberapa tepat klasifikasi yang telah dibangun, sesuai dengan target yang ada.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \% \quad (2.7)$$

b. *Precision*

*Precision* adalah perhitungan keakuratan antara data target dengan hasil prediksi dari model.

$$Precision = \frac{TP}{TP + FP} \times 100 \% \quad (2.7)$$

c. *Recall*

*Recall* adalah perhitungan yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi.

$$Recall = \frac{TP}{TP + FN} \times 100 \% \quad (2.8)$$

d. *F1-Score*

*F1-score* adalah perhitungan yang menggambarkan perbandingan antara *precision* dan *recall*. Jika nilai FN dan FP tidak mendekati sebaiknya digunakan *f1-score* dibandingkan nilai akurasi

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100 \% \quad (2.9)$$

## BAB III METODE PENELITIAN

### 3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian ini adalah penelitian eksperimental, dimana penelitian ini melakukan pengukuran tingkat akurasi yang dihasilkan dari metode LSTM dengan menggunakan *word embedding word2vec*.

Sifat penelitian ini adalah deskriptif, dimana pada penelitian ini menggambarkan suatu objek yang akan diteliti dan menjabarkan hasil dari analisis sentimen dan pengujian yang dilakukan.

Pendekatan penelitian ini adalah menggunakan metode kuantitatif, dimana nantinya akan menghasilkan angka dan grafik dari algoritma LSTM.

### 3.2. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini adalah teknik *scrapping*. Teknik *web scrapping* data dilakukan menggunakan bahasa pemrograman *python* dengan *library selenium* dengan menggunakan kata kunci "*skincare*" dan data tersebut disimpan kedalam format *.csv*. Berikut merupakan *source code* dari proses *scrapping* yang dilakukan.

```
import requests
from bs4 import BeautifulSoup
import time

#the website URL
list_data = []
max_page = 1000

product_name = "C White Night Moisturizer"
link = "https://reviews.femaledaily.com/products/moisturizer/night-cream-
```

```

16/azarine-cosmetics/c-white-night-
moisturizer?cat=&cat_id=&age_range=&skin_type=&skin_tone=&skin_undertone=&hair_texture=&hair_type=&order=newest&page="

total_data = 0
total_limit_year = 0
# limit_year = "2021"
is_done = False
total_not_found = 0
for i in range(max_page):
    time.sleep(0.5)
    if total_not_found > 6:
        break

    page = i+1
    url_link = link+str(page)
    result = requests.get(url_link).text
    doc = BeautifulSoup(result, "html.parser")
    rev = doc.find('div', class_="list-reviews")
    print('Scraping page', page, end='... ')

    if is_done:
        break

    if rev is not None:
        print('Found it', end='...')
        list_item = rev.find_all('div', class_='item')
        for j in list_item:
            username = j.find('p', class_='profile-username').text
            tanggal = j.find('p', class_='review-date').text
            review_text = j.find('p', class_='text-content').text
            stars = len(j.find_all('i', class_='icon-
is_big_star_full'))

            tmp_data = [product_name, username, tanggal, stars,
review_text]
            total_data+=1
            # Limit berdasarkan min tanggal yang ditentukan
            # is_tanggal = tanggal[-4:]
            # if is_tanggal == limit_year:
            #     total_limit_year+=1
            #     print('Total limit Year', total_limit_year)

            # if total_limit_year > 1000:
            #     print('Total lebih dari 1000')
            #     is_done = True
            #     break
            # try:
            #     if int(is_tanggal) < int(limit_year):
            #         is_done = True
            #         break
            # except:
            #     pass
            list_data.append(tmp_data)
            print('Done!!')
        else:
            total_not_found+=1
            print('Not found')

```

```
import pandas as pd
```



```
df_data = pd.DataFrame(list_data, columns=['Product', 'Username',
'Tanggal', 'Rating', 'Review'])
df_data
```

```
from datetime import datetime, timedelta

new_data = df_data
today = datetime.today()
# d = today - timedelta(days=1)
# print(d.strftime("%d %b %Y"))

for i, r in df_data.iterrows():
    tanggal = df_data['Tanggal'][i]

    spl = tanggal.split('/')
    tgl = ''
    if spl[1] == 'minute' or spl[1] == 'hour' or spl[1] == 'hours':
        tgl = today.strftime("%d %b %Y")
    elif spl[1] == 'day':
        d = today - timedelta(days=1)
        tgl = d.strftime("%d %b %Y")
    elif spl[1] == 'days':
        d = today - timedelta(days=int(spl[0]))
        tgl = d.strftime("%d %b %Y")
    if tgl != '':
        print(tgl)
        new_data.at[i, 'Tanggal'] = tgl
```

```
new_data.to_csv('datasets/'+product name+'.csv')
```

Adapun sampel data yang diperoleh dapat dilihat pada Tabel 3.1 berikut. Pada tabel tersebut terdiri dari *product* yaitu nama produk, *username* yaitu nama pengguna, *tanggal* yaitu tanggal pemberian ulasan, *rating* yaitu peringkat yang diberikan pada produk tersebut, dan *review* berisi komentar yang diberikan pengguna.

Tabel 3. 1 Sampel Data

	Product	...	...	Rating	Review
0	Bright Beauty Triple Glow Serum	...	...	5	setelah pakai rutin tiap hari, ponds ini bener2 menarik sih, salah satu hasil yang paling kelihatan adalah bekas jerawat di pipi yang awalnya gelap dan ganggu banget, sekarang jadi lebih pudar. tenang aja, ini teksturnya gampang meresap di kulit dan enggak lengket. ada sedikit aroma wangi, tapi gak ganggu. kalau ini habis, kayaknya bakal beli besok xixi.

Tabel 3.1 (Lanjutan)

1	Bright Beauty Triple Glow Serum	...	...	4	Kesan pertama pake ini dia cukup melembabkan jadi buat yang kulitnya kering bisa banget pake ini. Mudah di blend juga karena teksturnya cair. Namun mungkin bagi yang kurang suka skincare beraroma lumut, ini kurang cocok sih. Tapi, kalo untuk aku masih oke. Benefit selanjutnya yang aku rasain adalah setelah pemakaian beberapa minggu, efek mencerahkannya mulai terlihat dan bikin kulit jadi gak kusam. Di kulitku selama pemakaian tidak menimbulkan reaksi negatif. Untuk menyamakan flek hitam aku belum bisa ngasih tau karena kebetulan mukaku belum ada flek hitam.
2	Bright Beauty Triple Glow Serum	...	...	5	Aku nuka dan ga ada permasalahan dengan produk satu ini. Bentuknya droplet jadi jangan sampe pipetnya nyentuh kulit. Raiknya ditetesin ke tangan aja baru diratain ke muka. Atau beri jarak bbrp cm dari wajah. Serumannya juga cepet meresap, tapi untuk hasil yg terlihat belum terlalu signifikan karna baru coba 1. Mungkin kalo udh rutin pake 3-4 akan keliatan signifikannya.
3	Bright Beauty Triple Glow Serum	...	...	4	Aku dapet produk ini dari id try n review; setelah sekian lama banget ga coba produk ponds, kali ini aku berkesempatan buat cobain lagi! Packing nya botol kaca dengan warna botol pink dan aroma yg khas banget ponds, teksturnya menurutku agak thick ya jadi saat dipencet pipetnya agak susah keluar, tapi saat di apply ke wajah menurutku cepet meresap dan lumayan ngasih kelembaban di kulit, warnanya putih agak kuning dan so far cocok di aku, surprisingly karena dulu coba ponds selalu failed alias ga cocok :( udah dipakai 2 minggu dan yang sampai saat ini terasa kulit lebih lembut dan keliatan lebih sehat, aku pakai di pagi hari barengi dgn sunscreen, senengnya ga keliatan bikin kecut atau kusam. Bakal terus dipakai sampai habis biar lebih terlihat hasilnya, thx yaa FD
4	Bright Beauty Triple Glow Serum	...	...	5	Another produk yang gak disangka-sangka bagusnya. Pertama kali cobain yang serum kemasan sachet karena aku gak cocok. Lalu dari pemakaian rutin satu minggu langsung terlihat glowing, bekas2 jerawat memudar, kulit juga jadi lebih cerah jadinya beli yang full size. Asli raket banget.
5	Fresh Glow Multifunction Gel Aloe Bright UV	...	...	4	sudah pasarkan sejak lama sama moisturizer gel citra varian ini dan baru kesampean beli waktu surabaya x beauty. karena bentuknya gel tentunya moisturizer ini sangat ringan dan mudah menyerap. untuk aromanya sendiri tidak terlalu menyengat dan sangat soft. sebagai pemilik kulit kering tapi sangat benci body lotion yang sangat thick ini sih bisa jadi pengganti pelembab badan ya.
6	Fresh Glow Multifunction Gel Aloe Bright UV	...	...	5	aduuh, beneran ini ga lebay.. tapi mung ngaruh banget! pemakaian 1 bulan ga di skip samsek. gw pake sebelum pake pelembab siang atau sebelum makan pelembab mlm. bahkan gw pake ini doang abtm tidur, di jaduin face mask gitu. cepet bgt meresap, ga lengket dan pemakaian teratur selama 14 hari, lu org bisa terlihat wajah jd lebih cerah. oh iya, buat kalian yg ada masalah kemerahan atau sun burn, mending pake ini. kulit langsung lbh calming.

Tabel 3.1 (Lanjutan)

7	Fresh Glow Multifunction Gel Aloe Bright UV	...	...	5	terima kasih female daily sudah memberiku kesempatan lagi untuk mengikuti female daily try n review. untuk produknya aku dapat citra fresh glow multifunction gel aloe bright UV. tapi sayang karena aku ga pake ini jadi aku hibah ke pacarku xixi overall baguss
8	C White Night Moisturizer	...	...	4	Sama seperti day cream nya, meski namanya cream tp teksturnya lebih ke gel ya. ringan dan cepat meresap ke kulit. Pas bangun tidur pagi, kulit muka jd lembab dan fresh. Harga jg affordable. Ya utk basic skincare, ini cukup oke kok, yuk cobain jg
9	C White Night Moisturizer	...	...	5	vitamin c sekarang bukan cuman buat diminum, tapi skincare juga harus. Tekstur dari azarine night cream ini gel, jadi ringan banget diwajah & ga bikin berminyak. Aku tiap hari rutin pake ini selama sebulan hasilnya bagus banget. Udah ga muncul tuh yang namanya jerawat & beruntusan

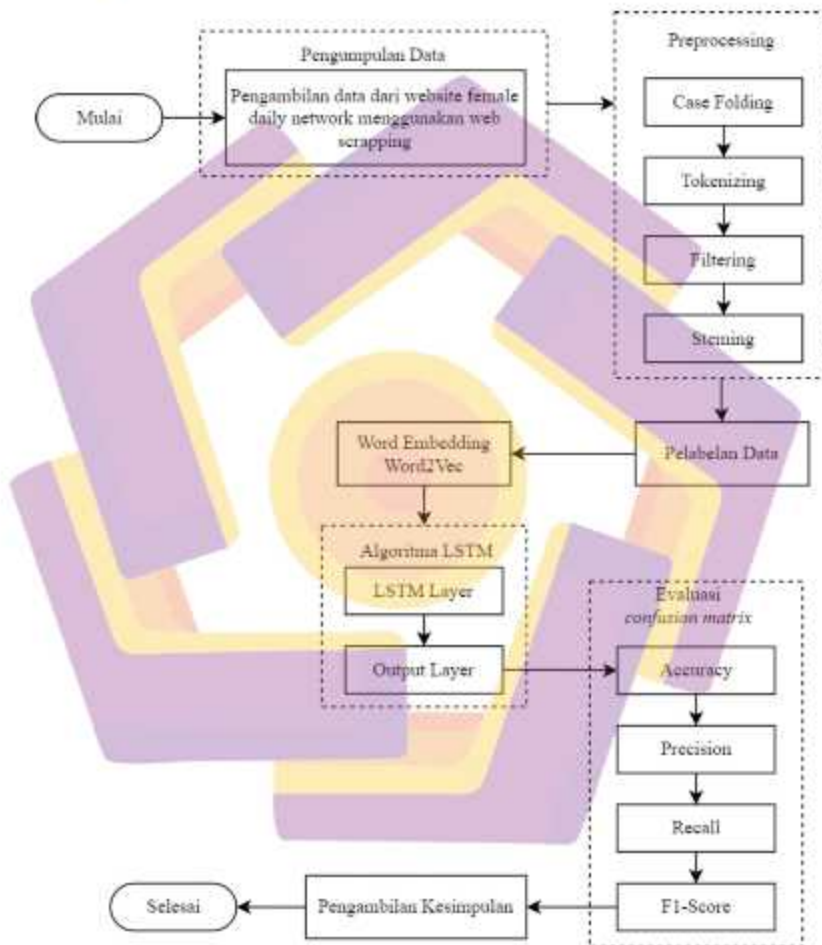
### 3.3. Metode Analisis Data

Dalam melakukan analisis data pada penelitian ini, data yang telah diperoleh dari *website female daily network* menggunakan *web scrapping* akan dilakukan pelabelan dan kemudian akan dilakukan proses *preprocessing data*, dimana pada proses ini data yang terdiri dari kata atau simbol yang tidak penting, sehingga data lebih mudah untuk diolah.

Setelah melakukan *preprocessing*, selanjutnya dilakukan proses *Word Embedding Word2Vec* dan akan ditambahkan sebelum menjadi input pada tahap *training data* dengan metode LSTM, hasil akhir dari proses *training* menggunakan metode LSTM ini berupa model yang dapat mengklasifikasikan data sebagai sentimen positif, negatif atau netral. Setelah itu hasil dari algoritma *Long Short-Term Memory (LSTM)* akan di evaluasi dengan menggunakan *Confusion Matrix* untuk mengetahui hasil tingkat akurasi, presisi, *recall* dan *f1-score*.

### 3.4. Alur Penelitian

Alur penelitian yang digunakan pada penelitian ini diilustrasikan pada gambar 3.1 dibawah ini.



Gambar 3. 1 Alur Penelitian

Berdasarkan Gambar 3.1 alur penelitian diatas, pada penelitian ini dimulai tahapan pengumpulan data dengan menggunakan teknik *web scrapping* data dengan

menggunakan kata kunci “*skincare*”. Kemudian data tersebut akan dilakukan pelabelan untuk membedakan setiap kelasnya.

Tahapan berikutnya adalah tahap *preprocessing* dilakukan karena memiliki pengaruh yang cukup baik dalam meningkatkan kinerja sistem dimana dari hasil pengujian yang melibatkan teknik *casefolding*, *tokenizing*, *filtering*, dan *stemming* yang memberikan pengaruh yang cukup baik dalam meningkatkan kinerja analisis sentimen (Khairunnisa, Adiwijaya, & Faraby, 2021).

*Dataset* yang telah melalui tahapan *preprocessing* akan dilakukan pelabelan menggunakan *corpus*, selanjutnya dilakukan ekstraksi fitur dengan menggunakan metode *word2vec*. *Word2vec* didasarkan pada ide *deep learning* di mana kata direpresentasikan dalam *vector*, selain itu klasifikasi dengan menggunakan *word2vec* sebagai ekstraksi fitur menghasilkan akurasi yang lebih baik.

Proses selanjutnya adalah melakukan pembagian komposisi *dataset* untuk *training* dan *testing*, kemudian melakukan klasifikasi menggunakan metode LSTM, dimana pada tahap ini setiap kata yang sudah direpresentasikan ke dalam bentuk vektor (*word vector*) dimasukkan ke dalam unit LSTM satu per satu sesuai dengan urutan dari *sequence file review* (Widayat, 2021). Setelah memperoleh hasil dari LSTM, tahap berikutnya adalah melakukan evaluasi dengan menggunakan *Confusion Matrix* untuk mengetahui hasil tingkat akurasi, presisi, *recall*, dan *F1-Score*. Setelah diperoleh hasil akurasi yang selanjutnya akan dianalisis dan ditarik kesimpulan.



## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data dengan menggunakan teknik *web scrapping* dan menggunakan bahasa pemrograman *Python*. Hasil dari proses *web scrapping* dapat dilihat pada gambar 4.1 dibawah ini.

	Product	Username	Tanggal	Rating	Review
0	C White Night Moisturizer	Ittepings	10 Nov 2022	4	Sama seperti itu dalam nya enak namanya cream...
1	C White Night Moisturizer	Heedaya	03 Nov 2022	5	terima kasih barang bukan suram buat dirumah t...
2	C White Night Moisturizer	RizkaKhan	01 Nov 2022	5	Tekstur ringan dan gampang menyerap ke kulit...
3	C White Night Moisturizer	Sarah_Amelia	12 Sep 2022	4	Tekstur enak sangat bagus dan ringan, sangat c...
4	C White Night Moisturizer	fyatiana	02 Sep 2022	4	Warna kulit jadi lebih merata dan lebih halus...
...	...	...	...	...	...
1763	C White Night Moisturizer	amely	21 Mar 2020	4	Hi aku pakai salah satunya, awalnya sih g...
1764	C White Night Moisturizer	jimi_	16 Mar 2020	4	SIANTUL BANDET LAN ANJIR & Oh, Di rumah gue sa...
1766	C White Night Moisturizer	Ranyush	13 Mar 2020	4	Walaupun namanya night moisturizer tapi ts sk...
1766	C White Night Moisturizer	immumpi	11 Mar 2020	4	Awal2 pemakaian muncul kayak komedo putih glu...
1767	C White Night Moisturizer	Thalitaque	07 Mar 2020	5	pertama kali tanggal aku coba produk skincare d...

Gambar 4. 1 *Web Scrapping* dari Femaledaily.com

#### 4.2. Preprocessing

Tahap selanjutnya adalah tahap *preprocessing*, dimana beberapa proses seperti *case folding*, *tokenizing*, *filtering*, dan *stemming* akan dilakukan pada data yang ada saat ini. *Case folding* adalah langkah pertama dalam prosedur *preprocessing*. Semua huruf besar yang ada di dalam dokumen akan dihapus, angka, dll. Proses penggantian baris baru dengan spasi juga dilakukan, menghapus tanda baca, mengubah semua huruf menjadi huruf kecil, memisahkan *string*



Dari data tersebut kemudian dilihat total label yang didapatkan seperti pada Gambar 4.3.

```
Total Label :
Label
Negative    1072
Neutral     117
Positive    1488
```

Gambar 4.3 Total Label

Dari Gambar 4.3 tersebut didapatkan hasil pelabelan negative sejumlah 1072, neutral dengan total 117, dan positive dengan total 1488. Selanjutnya, dari data tersebut, dapat melihat berapa data yang Non-Null pada setiap row. Jika jumlahnya lebih sedikit maka terdapat null value di dalamnya. Yang dapat dilihat pada Gambar 4.4 dibawah ini.

```
<class 'pandas.core.frame.DataFrame'>
Index: 2677 entries, destiana_nrb to DitaIndahSarii
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Product         2677 non-null   object
1   Rating          2677 non-null   float64
2   tweet           2677 non-null   object
3   casefolding     2677 non-null   object
4   tokenizing      2677 non-null   object
5   #filtering      2677 non-null   object
6   stemming        2677 non-null   object
7   Label           2677 non-null   object
dtypes: float64(1), object(7)
memory usage: 188.2+ KB
```

Gambar 4.4 Null Value

### 4.3. Algoritma LSTM Dengan Word2Vec

Tahap selanjutnya adalah melakukan *splitting data*. Tujuan *splitting data* adalah untuk menghindari *overfitting*, yaitu kondisi dimana model terlalu mempelajari data training sehingga tidak mampu memprediksi data yang belum pernah dilihat sebelumnya dengan baik. Pada proses *splitting* ini, digunakan metode

memisahkan dataset berdasarkan labelnya, kemudian menggabungkan kembali menjadi data *train* dan data *test*. Pada penelitian ini, peneliti menggunakan perbandingan 70% data *training* dan 30% data *testing*, 80% data *training* dan 20% data *testing* dan 90% data *training* dan 10% data *testing*. Penggunaan proporsi tersebut dilakukan karena berdasarkan penelitian yang dilakukan oleh (Fitriani, Yasin, & Tarno, 2021) menyebutkan bahwa proporsi data pelatihan harus lebih besar dibandingkan data uji. Selain itu, berdasarkan hasil penelitian yang dilakukan oleh (Tilasefana & Putra, 2023) menyatakan bahwa semakin besar data training maka semakin tinggi juga ketepatan yang diperoleh. Adapun hasil yang diperoleh dari setiap *splitting data* adalah sebagai berikut:

#### 4.5.1 Splitting Data 70:30

Pada *splitting data* 70:30 dilakukan perbandingan data *training* sebesar 70% dan data *test* sebesar 30%, dimana pada label *negative*, nilai *train negative* adalah 750 dan *test negative* adalah 322. Kemudian label *neutral*, nilai *train negative* adalah 81 dan *test negative* adalah 36. Terakhir label *positive*, nilai *train negative* adalah 1041 dan *test negative* adalah 447. Tahap selanjutnya adalah menggabungkan kembali setiap data untuk data *train & test* yang dapat dilihat pada Gambar 4.5 dibawah ini.

Label	Train	Test
negative	750	322
neutral	81	36
positive	1041	447
<b>Total</b>	<b>1872</b>	<b>805</b>

Gambar 4.5 Proses Penggabungan Data *Train*



Dari proses diatas, didapatkan data total *label train* adalah nilai *negative* sebanyak 750 label, kemudian *neutral* adalah 81 dan *positive* adalah sebesar 1041.

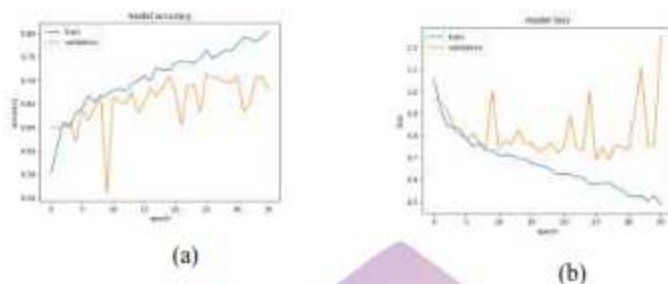
Selanjutnya adalah melakukan penggabungan data *test*, sehingga didapatkan data total label test adalah nilai *negative* sebanyak 322, nilai *neutral* adalah 36 dan *positive* adalah sebanyak 447. Setelah data gabungan tahap selanjutnya adalah *Generate Word2Vec Model*. Tahap ini akan dilatih model *Word2Vec* menggunakan data training untuk menghasilkan vektor representasi kata dengan menggunakan parameter seperti dimensi *embedding*, *window size*, *min\_count*, *negative*, *workers*, *seed*, dan metode pembelajaran (skip-gram atau CBOW). Tahap pertama adalah mencari panjang maksimum dari teks dalam variabel *x\_train*. Langkahnya adalah pertama menginisialisasi variabel *max\_len* dengan nilai 0 untuk menyimpan panjang maksimum teks. Selanjutnya dilakukan iterasi pada setiap teks (*r*) dalam *X\_train* dan menggunakan metode *split (' ')* untuk membagi teks menjadi kata-kata. Kemudian, dilakukan pengecekan apakah panjang kata-kata tersebut lebih besar dari nilai *max\_len* saat ini. Jika ya, maka nilai *max\_len* diperbarui dengan panjang kata-kata tersebut. Setelah iterasi selesai, nilai *max\_len* akan berisi panjang maksimum dari teks dalam *x\_train*. Dalam kasus ini output yang diperoleh adalah nilai 348, yang menunjukkan bahwa teks terpanjang dalam *X\_train* memiliki panjang 348 kata. Tahap kedua adalah dilakukan vektorisasi menggunakan TensorFlow. Langkahnya adalah dilakukan pengaturan *inpur\_length* sebagai *max\_len* yang telah dihitung sebelumnya. Jumlah maksimum kata yang akan disimpan ditentukan sebagai *vocab\_length*. Kemudian dilakukan pembuatan tokenizer menggunakan *Tokenizer* dari TensorFlow dengan parameter



`num_words` dan `oov_token`. Kosakata internal tokenizer diperbarui berdasarkan teks pada `X_train`. Terakhir dilakukan perubahan teks menjadi urutan bilangan bulat menggunakan metode `texts_to_sequences`, dan dilakukan konversi daftar menjadi array Numpy 2D menggunakan `pad_sequences`. Selanjutnya, dilakukan `pad_sequences` untuk mengubah teks menjadi urutan bilangan bulat dengan menggunakan tokenizer. Kemudian, dilakukan `print` untuk menampilkan bentuk (`shape`) dari `X_train_data` dan `X_test_data`. Selanjutnya, label diubah menjadi bentuk `categorical` menggunakan `to_categorical` karena jumlah label lebih dari 2. Pada bagian tersebut, dilakukan `print` untuk menampilkan bentuk (`shape`) dari variabel `X_train` dan `X_test`. Bentuk (`shape`) `X_train` adalah (1872, 348), artinya `X_train` memiliki 1872 baris dan 348 kolom. Bentuk (`shape`) `X_test` adalah (805, 348), artinya `X_test` memiliki 805 baris dan 348 kolom. Selanjutnya adalah membuat matriks `embedding` yang menghubungkan setiap token dalam kosakata dengan representasi vektornya. Matriks `embedding` ini memiliki ukuran (10000, 1000), yang berarti terdapat 10000 kata dalam kosakata yang direpresentasikan dalam vektor berdimensi 1000. Kami menggunakan model `Word2Vec` untuk mendapatkan vektor yang sesuai dengan setiap kata dalam kosakata.

Tahap selanjutnya adalah LSTM Model. Dimana pertama adalah membuat arsitektur LSTM. Arsitektur dari LSTM dapat dilihat pada Gambar 4.6 dibawah ini.





Gambar 4. 8 Akurasi Train dan Validation (a), Loss Train dan Validation

#### 4.5.2 Splitting Data 80:20

Pada *splitting* data 80:20 dilakukan perbandingan data *training* sebesar 80% dan data *test* sebesar 20%, dimana pada label negative, nilai train negative adalah 857 dan test negative adalah 215. Kemudian label neutral, nilai train negative adalah 93 dan test negative adalah 24. Terakhir label positive, nilai train negative adalah 11190 dan test negative adalah 298.

Tahap selanjutnya adalah menggabungkan kembali setiap data untuk data train & test. Dari proses tersebut, didapatkan data total label train adalah nilai negative sebanyak 857 label, kemudian neutral adalah 93 dan positive adalah sebesar 1190.

Selanjutnya adalah penggabungan data test yang dapat dilihat pada Gambar 4.9 dibawah ini.

Group	Label	Count
0	negative	215
0	neutral	24
0	positive	298

Gambar 4. 9 Proses Penggabungan Data Test

Dari proses diatas, didapatkan data total label test adalah nilai negative sebanyak 215, nilai neutral adalah 24 dan positive adalah sebanyak 298.

Setelah data gabungan tahap selanjutnya adalah Generate Word2Vec Model. Tahap ini akan dilatih model Word2Vec menggunakan data training untuk menghasilkan vektor representasi kata dengan menggunakan parameter seperti dimensi *embedding*, *window size*, *min count*, *negative*, *workers*, *seed* dan metode pembelajaran (*skip-gram* atau *CBOW*). Tahap pertama adalah mencari panjang maksimum dari teks dalam variabel `x_train`. Langkahnya adalah pertama menginisialisasi variabel `max_len` dengan nilai 0 untuk menyimpan panjang maksimum teks. Selanjutnya dilakukan iterasi pada setiap teks (`r`) dalam `X_train` dan menggunakan metode `split (' ')` untuk membagi teks menjadi kata-kata. Kemudian, dilakukan pengecekan apakah panjang kata-kata tersebut lebih besar dari nilai `max_len` saat ini. Jika ya, maka nilai `max_len` diperbarui dengan panjang kata-kata tersebut. Setelah iterasi selesai, nilai `max_len` akan berisi panjang maksimum dari teks dalam `x_train`. Dalam kasus ini output yang diperoleh adalah nilai 348, yang menunjukkan bahwa teks terpanjang dalam `X_train` memiliki

panjang 348 kata. Tahap kedua adalah dilakukan vektorisasi menggunakan TensorFlow. Langkahnya adalah dilakukan pengaturan `inpur_length` sebagai `max_len` yang telah dihitung sebelumnya. Jumlah maksimum kata yang akan disimpan ditentukan sebagai `vocab_length`. Kemudian dilakukan pembuatan tokenizer menggunakan `Tokenizer` dari TensorFlow dengan parameter `num_words` dan `oov_token`. Kosakata internal tokenizer diperbarui berdasarkan teks pada `X_train`. Terakhir dilakukan perubahan teks menjadi urutan bilangan bulat menggunakan metode `texts_to_sequences`, dan dilakukan konversi daftar menjadi array Numpy 2D menggunakan `pad_sequences`. Selanjutnya, dilakukan `pad_sequences` untuk mengubah teks menjadi urutan bilangan bulat dengan menggunakan tokenizer. Kemudian, dilakukan print untuk menampilkan bentuk (`shape`) dari `X_train_data` dan `X_test_data`. Selanjutnya, label diubah menjadi bentuk `categorical` menggunakan `to_categorical` karena jumlah label lebih dari 2. Pada bagian tersebut, dilakukan print untuk menampilkan bentuk (`shape`) dari variabel `X_train` dan `X_test`. Bentuk (`shape`) `X_train` adalah (2140, 348), artinya `X_train` memiliki 2140 baris dan 348 kolom. Bentuk (`shape`) `X_test` adalah (537, 348), artinya `X_test` memiliki 537 baris dan 348 kolom. Selanjutnya adalah membuat matriks `embedding` yang menghubungkan setiap token dalam kosakata dengan representasi vektornya. Matriks `embedding` ini memiliki ukuran (10000, 1000), yang berarti terdapat 10000 kata dalam kosakata yang direpresentasikan dalam vektor berdimensi 1000. Kami menggunakan model `Word2Vec` untuk mendapatkan vektor yang sesuai dengan setiap kata dalam kosakata.



Tahap selanjutnya adalah LSTM Model. Dimana pertama adalah membuat arsitektur LSTM Arsitektur dari LSTM dapat dilihat pada Gambar 4.10 dibawah ini.

```
model: "sequential"
```

Layer (Type)	Output Shape	Param #
embedding (Embedding)	(None, 144, 1000)	1000000
lstm (LSTM)	(None, 144, 128)	172048
dropout (Dropout)	(None, 144, 128)	0
lstm_1 (LSTM)	(None, 128)	131584
batch_normalization (Batch Normalization)	(None, 128)	112
dense (Dense)	(None, 64)	1296
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 7)	105

---

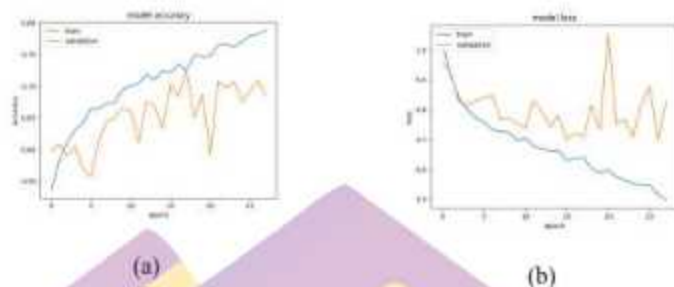
```
Total params: 1871656 (49.89 MB)
Trainable params: 710220 (2.24 MB)
Non-trainable params: 10000256 (30.13 MB)
```

Gambar 4. 10 Total Parameter

Kemudian model tersebut dilatih dengan data training dan selanjutnya dievaluasi dengan data validasi. Adapun ukuran batch (*batch\_size*) yang digunakan sebanyak 32 dengan epoch sebanyak 100. Setelah dikompilasi, diperoleh nilai *loss* mencapai 0.4968 dengan nilai akurasi sebesar 0.7888. Epoch berhenti secara otomatis di epoch 28 karena menggunakan fungsi *early stopping* guna memberhentikan proses pelatihan data ketika terjadi overfit. Adapun proses epoch dapat dilihat pada Gambar 4.11.

```
Epoch 28/100
28/28 [=====] - 12s 4s - loss: 0.4968 - accuracy: 0.7888
Epoch 29: val_loss: 0.5071 - val_acc: 0.7888
Epoch 30: val_loss: 0.5174 - val_acc: 0.7888
Epoch 31: val_loss: 0.5277 - val_acc: 0.7888
Epoch 32: val_loss: 0.5380 - val_acc: 0.7888
Epoch 33: val_loss: 0.5483 - val_acc: 0.7888
Epoch 34: val_loss: 0.5586 - val_acc: 0.7888
Epoch 35: val_loss: 0.5689 - val_acc: 0.7888
Epoch 36: val_loss: 0.5792 - val_acc: 0.7888
Epoch 37: val_loss: 0.5895 - val_acc: 0.7888
Epoch 38: val_loss: 0.5998 - val_acc: 0.7888
Epoch 39: val_loss: 0.6101 - val_acc: 0.7888
Epoch 40: val_loss: 0.6204 - val_acc: 0.7888
Epoch 41: val_loss: 0.6307 - val_acc: 0.7888
Epoch 42: val_loss: 0.6410 - val_acc: 0.7888
Epoch 43: val_loss: 0.6513 - val_acc: 0.7888
Epoch 44: val_loss: 0.6616 - val_acc: 0.7888
Epoch 45: val_loss: 0.6719 - val_acc: 0.7888
Epoch 46: val_loss: 0.6822 - val_acc: 0.7888
Epoch 47: val_loss: 0.6925 - val_acc: 0.7888
Epoch 48: val_loss: 0.7028 - val_acc: 0.7888
Epoch 49: val_loss: 0.7131 - val_acc: 0.7888
Epoch 50: val_loss: 0.7234 - val_acc: 0.7888
Epoch 51: val_loss: 0.7337 - val_acc: 0.7888
Epoch 52: val_loss: 0.7440 - val_acc: 0.7888
Epoch 53: val_loss: 0.7543 - val_acc: 0.7888
Epoch 54: val_loss: 0.7646 - val_acc: 0.7888
Epoch 55: val_loss: 0.7749 - val_acc: 0.7888
Epoch 56: val_loss: 0.7852 - val_acc: 0.7888
Epoch 57: val_loss: 0.7955 - val_acc: 0.7888
Epoch 58: val_loss: 0.8058 - val_acc: 0.7888
Epoch 59: val_loss: 0.8161 - val_acc: 0.7888
Epoch 60: val_loss: 0.8264 - val_acc: 0.7888
Epoch 61: val_loss: 0.8367 - val_acc: 0.7888
Epoch 62: val_loss: 0.8470 - val_acc: 0.7888
Epoch 63: val_loss: 0.8573 - val_acc: 0.7888
Epoch 64: val_loss: 0.8676 - val_acc: 0.7888
Epoch 65: val_loss: 0.8779 - val_acc: 0.7888
Epoch 66: val_loss: 0.8882 - val_acc: 0.7888
Epoch 67: val_loss: 0.8985 - val_acc: 0.7888
Epoch 68: val_loss: 0.9088 - val_acc: 0.7888
Epoch 69: val_loss: 0.9191 - val_acc: 0.7888
Epoch 70: val_loss: 0.9294 - val_acc: 0.7888
Epoch 71: val_loss: 0.9397 - val_acc: 0.7888
Epoch 72: val_loss: 0.9500 - val_acc: 0.7888
Epoch 73: val_loss: 0.9603 - val_acc: 0.7888
Epoch 74: val_loss: 0.9706 - val_acc: 0.7888
Epoch 75: val_loss: 0.9809 - val_acc: 0.7888
Epoch 76: val_loss: 0.9912 - val_acc: 0.7888
Epoch 77: val_loss: 1.0015 - val_acc: 0.7888
Epoch 78: val_loss: 1.0118 - val_acc: 0.7888
Epoch 79: val_loss: 1.0221 - val_acc: 0.7888
Epoch 80: val_loss: 1.0324 - val_acc: 0.7888
Epoch 81: val_loss: 1.0427 - val_acc: 0.7888
Epoch 82: val_loss: 1.0530 - val_acc: 0.7888
Epoch 83: val_loss: 1.0633 - val_acc: 0.7888
Epoch 84: val_loss: 1.0736 - val_acc: 0.7888
Epoch 85: val_loss: 1.0839 - val_acc: 0.7888
Epoch 86: val_loss: 1.0942 - val_acc: 0.7888
Epoch 87: val_loss: 1.1045 - val_acc: 0.7888
Epoch 88: val_loss: 1.1148 - val_acc: 0.7888
Epoch 89: val_loss: 1.1251 - val_acc: 0.7888
Epoch 90: val_loss: 1.1354 - val_acc: 0.7888
Epoch 91: val_loss: 1.1457 - val_acc: 0.7888
Epoch 92: val_loss: 1.1560 - val_acc: 0.7888
Epoch 93: val_loss: 1.1663 - val_acc: 0.7888
Epoch 94: val_loss: 1.1766 - val_acc: 0.7888
Epoch 95: val_loss: 1.1869 - val_acc: 0.7888
Epoch 96: val_loss: 1.1972 - val_acc: 0.7888
Epoch 97: val_loss: 1.2075 - val_acc: 0.7888
Epoch 98: val_loss: 1.2178 - val_acc: 0.7888
Epoch 99: val_loss: 1.2281 - val_acc: 0.7888
Epoch 100: val_loss: 1.2384 - val_acc: 0.7888
```

Gambar 4. 11 Proses Epoch



Gambar 4. 12 Akurasi Train dan Validation (a), Loss Train dan Validation

#### 4.5.3 Splitting Data 90:10

Pada *splitting* data 90:10 dilakukan perbandingan data *training* sebesar 90% dan data *test* sebesar 10%, dimana Pada label negative, nilai train negative adalah 964 dan test negative adalah 108. Kemudian label neutral, nilai train negative adalah 105 dan test negative adalah 12. Terakhir label positive, nilai train negative adalah 1339 dan test negative adalah 149.

Tahap selanjutnya adalah menggabungkan kembali setiap data untuk data train & test yang dapat dilihat pada gambar Gambar 4.13 dibawah.

nama	model	jenis	jumlah	jumlah	jumlah	jumlah	jumlah
training	positive	1339	negative	964	neutral	105	total
test	positive	149	negative	108	neutral	12	total
total	positive	1488	negative	1072	neutral	117	total
total	positive	1339	negative	964	neutral	105	total
total	positive	149	negative	108	neutral	12	total

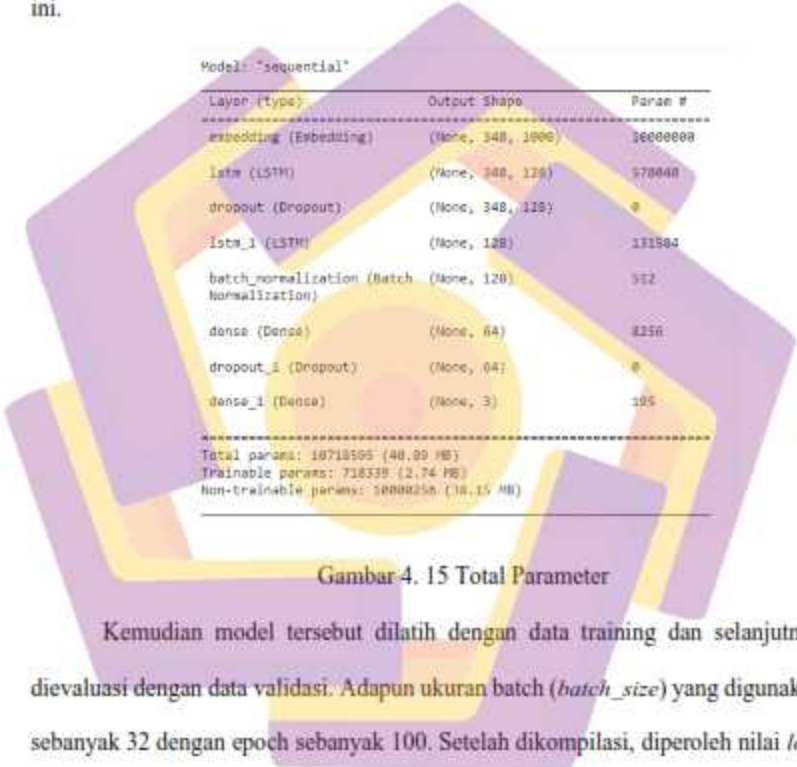
Gambar 4. 13 Proses Penggabungan Data Train



kata-kata tersebut. Setelah iterasi selesai, nilai `max_len` akan berisi panjang maksimum dari teks dalam `x_train`. Dalam kasus ini output yang diperoleh adalah nilai 348, yang menunjukkan bahwa teks terpanjang dalam `X_train` memiliki panjang 348 kata. Tahap kedua adalah dilakukan vektorisasi menggunakan TensorFlow. Langkahnya adalah dilakukan pengaturan `input_length` sebagai `max_len` yang telah dihitung sebelumnya. Jumlah maksimum kata yang akan disimpan ditentukan sebagai `vocab_length`. Kemudian dilakukan pembuatan tokenizer menggunakan `Tokenizer` dari TensorFlow dengan parameter `num_words` dan `oov_token`. Kosakata internal tokenizer diperbarui berdasarkan teks pada `X_train`. Terakhir dilakukan perubahan teks menjadi urutan bilangan bulat menggunakan metode `texts_to_sequences`, dan dilakukan konversi daftar menjadi array Numpy 2D menggunakan `pad_sequences`. Selanjutnya, dilakukan `pad_sequences` untuk mengubah teks menjadi urutan bilangan bulat dengan menggunakan tokenizer. Kemudian, dilakukan print untuk menampilkan bentuk (`shape`) dari `X_train_data` dan `X_test_data`. Selanjutnya, label diubah menjadi bentuk `categorical` menggunakan `to_categorical` karena jumlah label lebih dari 2. Pada bagian tersebut, dilakukan print untuk menampilkan bentuk (`shape`) dari variabel `X_train` dan `X_test`. Bentuk (`shape`) `X_train` adalah (2408, 348), artinya `X_train` memiliki 2408 baris dan 348 kolom. Bentuk (`shape`) `X_test` adalah (269, 348), artinya `X_test` memiliki 269 baris dan 348 kolom. Selanjutnya adalah membuat matriks embedding yang menghubungkan setiap token dalam kosakata dengan representasi vektornya. Matriks embedding ini memiliki ukuran (10000, 1000), yang berarti terdapat 10000 kata dalam kosakata yang direpresentasikan

dalam vektor berdimensi 1000. Tahap ini menggunakan model Word2Vec untuk mendapatkan vektor yang sesuai dengan setiap kata dalam kosakata.

Tahap selanjutnya adalah LSTM Model. Dimana pertama adalah membuat arsitektur LSTM Arsitektur dari LSTM dapat dilihat pada Gambar 4.15 dibawah ini.



```

Model: "sequential"
-----
Layer (type)                Output Shape         Param #
-----
embedding (Embedding)      (None, 348, 1000)   36000000
lstm (LSTM)                 (None, 348, 128)    570048
dropout (Dropout)          (None, 348, 128)    0
lstm_1 (LSTM)              (None, 128)         131584
batch_normalization (Batch (None, 128)         512
Normalization)
dense (Dense)              (None, 64)          8256
dropout_1 (Dropout)        (None, 64)          0
dense_1 (Dense)            (None, 3)           195
-----
Total params: 16718595 (48.00 MB)
Trainable params: 716339 (2.74 MB)
Non-trainable params: 16002256 (18.15 MB)

```

Gambar 4. 15 Total Parameter

Kemudian model tersebut dilatih dengan data training dan selanjutnya dievaluasi dengan data validasi. Adapun ukuran batch (*batch\_size*) yang digunakan sebanyak 32 dengan epoch sebanyak 100. Setelah dikompilasi, diperoleh nilai *loss* mencapai 0.4432 dengan nilai akurasi sebesar 0.8248. Epoch berhenti secara otomatis di epoch 31 karena menggunakan fungsi early stopping guna memberhentikan proses pelatihan data ketika terjadi overfit. Adapun proses epoch dapat dilihat pada Gambar 4.16.

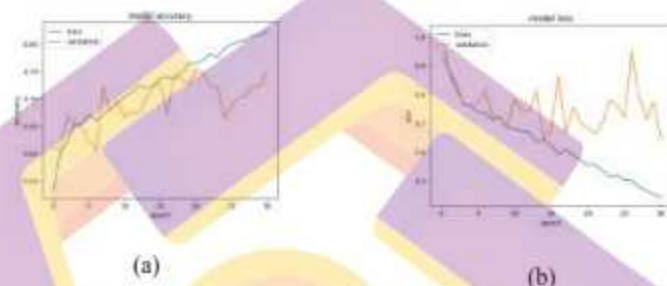


```

Epoch 16: val_accuracy did not improve from 0.7549
75/76 [====...] - 127s 24/step - loss: 0.4758 - accuracy: 0.8027 - val_loss: 0.7554 - val_accuracy: 0.7602
Epoch 16: 100%
75/76 [====...] - 174s 40 - loss: 0.4078 - accuracy: 0.8008
Epoch 16: val_accuracy did not improve from 0.7549
75/76 [====...] - 127s 24/step - loss: 0.4976 - accuracy: 0.8206 - val_loss: 0.7547 - val_accuracy: 0.7540
Epoch 16: 100%
75/76 [====...] - 172s 36 - loss: 0.4034 - accuracy: 0.8034
Epoch 16: val_accuracy did not improve from 0.7549
75/76 [====...] - 127s 24/step - loss: 0.4938 - accuracy: 0.8168 - val_loss: 0.7527 - val_accuracy: 0.7539
Epoch 16: 100%
75/76 [====...] - 173s 36 - loss: 0.4032 - accuracy: 0.8044
Epoch 16: val_accuracy did not improve from 0.7549
75/76 [====...] - 127s 24/step - loss: 0.4882 - accuracy: 0.8168 - val_loss: 0.8002 - val_accuracy: 0.7549
Epoch 16: early stopping

```

Gambar 4. 16 Proses Epoch

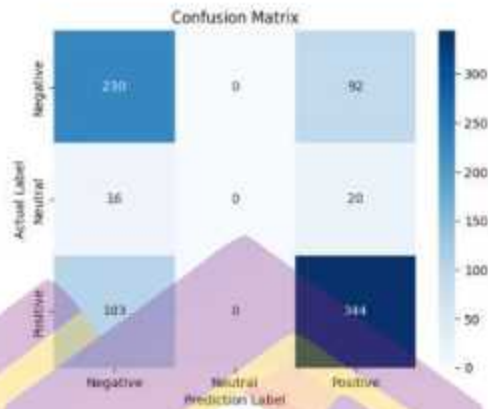


Gambar 4. 17 Akurasi Train dan Validation (a), Loss Train dan Validation

#### 4.4. Evaluasi *Confusion Matrix* Dengan *Word2Vec*

##### 4.4.1 Hasil Evaluasi LSTM dengan *Splitting Data 70:30*

Hasil evaluasi *confusion matrix* yang diperoleh dari model LSTM dengan *Splitting Data 70:30* dapat dilihat pada Gambar 4.18 dibawah ini.



Gambar 4. 18 *Confusion Matrix*

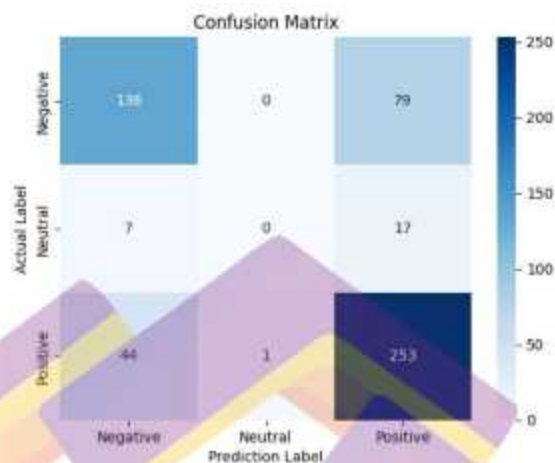
Berdasarkan proses diatas, didapatkan nilai *accuracy* pada data train dan validation adalah sebesar 71%. Selain itu juga menunjukkan bahwa label 0 (*Negative*) mendapatkan nilai *precision* sebesar 66 %, *recall* sebesar 71 %, *f1-score* sebesar 69 %. Sedangkan pada label 2 (*Positive*) menghasilkan nilai *precision* sebesar 75 %, *recall* sebesar 77 %, *f1-score* sebesar 76 %. Hasil dari proses tersebut dapat dilihat pada Gambar 4.19 dibawah ini.

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
0	0.66	0.71	0.69	322
1	0.00	0.00	0.00	36
2	0.75	0.77	0.76	447
<i>accuracy</i>			0.71	805
<i>macro avg</i>	0.47	0.49	0.48	805
<i>weighted avg</i>	0.68	0.71	0.70	805

Gambar 4. 19 Hasil Perhitungan *Confusion Matrix*

Dari hasil perhitungan diatas, didapatkan hasil evaluasi negative, didapatkan akurasi sebesar 71.43% yang dapat dilihat pada Gambar 4.20 dibawah.





Gambar 4. 22 *Confusion Matrix*

Berdasarkan proses diatas, didapatkan nilai *accuracy* pada data *train* dan *validation* adalah sebesar 72%. Selain itu juga menunjukkan bahwa label 0 (*Negative*) mendapatkan nilai *precision* sebesar 73 %, *recall* sebesar 63 %, *f1-score* sebesar 68 %. Sedangkan pada label 2 (*Positive*) menghasilkan nilai *precision* sebesar 72 %, *recall* sebesar 85 %, *f1-score* sebesar 78 %. Hasil dari proses tersebut dapat dilihat pada Gambar 4.23 dibawah ini.

	precision	recall	f1-score	support
0	0.73	0.63	0.68	215
1	0.00	0.00	0.00	24
2	0.72	0.85	0.78	298
accuracy			0.72	537
macro avg	0.48	0.49	0.49	537
weighted avg	0.69	0.72	0.70	537

Gambar 4. 23 Hasil Perhitungan *Confusion Matrix*

Dari hasil perhitungan diatas, didapatkan hasil evaluasi *negative*, didapatkan akurasi sebesar 63.26% yang dapat dilihat pada Gambar 4.24 dibawah.

	Actual = 0	Actual = 1	Total
Predicted = 0	10	5	15
Predicted = 1	5	10	15
Total	15	15	30

Gambar 4. 24 Hasil Evaluasi Label Negative

Dan hasil evaluasi *positive* mendapatkan hasil akurasi sebesar 84.9% yang dapat dilihat pada Gambar 4.25 dibawah ini.

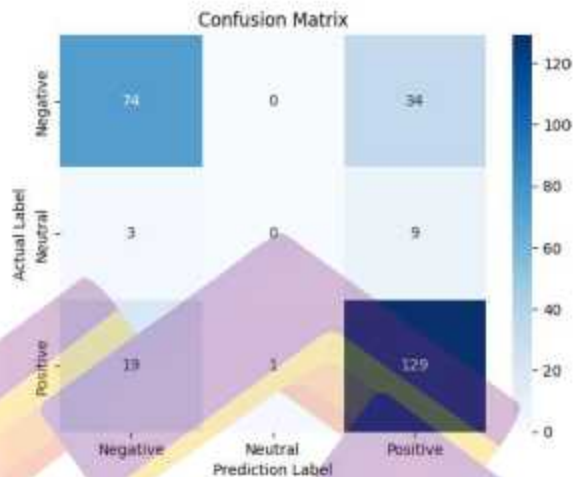
	Actual = 0	Actual = 1	Total
Predicted = 0	10	2	12
Predicted = 1	2	10	12
Total	12	12	24

Gambar 4. 25 Hasil Evaluasi Label Positive

#### 4.4.3 Hasil Evaluasi LSTM dengan Splitting Data 90:10

Hasil evaluasi *confusion matrix* yang diperoleh dari model LSTM dengan *Splitting* Data 90:10 dapat dilihat pada Gambar 4.26.





Gambar 4. 26 *Confusion Matrix*

Berdasarkan proses diatas, didapatkan nilai *accuracy* pada data train dan validation adalah sebesar 75%. Selain itu juga menunjukkan bahwa label 0 (*Negative*) mendapatkan nilai *precision* sebesar 77%, *recall* sebesar 69%, *f1-score* sebesar 73%. Sedangkan pada label 2 (*Positive*) menghasilkan nilai *precision* sebesar 75%, *recall* sebesar 87%, *f1-score* sebesar 80%. Hasil dari proses tersebut dapat dilihat pada Gambar 4.27 dibawah ini.

	precision	recall	f1-score	support
0	0.77	0.69	0.73	108
1	0.00	0.00	0.00	12
2	0.75	0.87	0.80	149
accuracy			0.75	269
macro avg	0.51	0.52	0.51	269
weighted avg	0.72	0.75	0.74	269

Gambar 4. 27 Hasil Perhitungan *Confusion Matrix*



dataset berdasarkan labelnya, kemudian menggabungkan kembali menjadi data *train* dan data *test*. Pada penelitian ini, peneliti menggunakan perbandingan 70% data *training* dan 30% data *testing*, 80% data *training* dan 20% data *testing* dan 90% data *training* dan 10% data *testing*. Adapun hasil yang diperoleh dari setiap *splitting data* adalah sebagai berikut:

#### 4.5.1 Splitting Data 70:30

Pada *splitting data* 70:30 dilakukan perbandingan data *training* sebesar 70% dan data *test* sebesar 30%, dimana pada label *negative*, nilai *train negative* adalah 750 dan *test negative* adalah 322. Kemudian label *neutral*, nilai *train neutral* adalah 81 dan *test neutral* adalah 36. Terakhir label *positive*, nilai *train negative* adalah 1041 dan *test negative* adalah 447. Tahap selanjutnya adalah menggabungkan kembali setiap data untuk data *train* & *test* yang dapat dilihat pada Gambar 4.30 dibawah ini.

id	text	sentiment	label
1	... ..	...	...
2	... ..	...	...
3	... ..	...	...
4	... ..	...	...
5	... ..	...	...
6	... ..	...	...
7	... ..	...	...
8	... ..	...	...
9	... ..	...	...
10	... ..	...	...

Gambar 4. 30 Proses Penggabungan Data *Train*

Dari proses diatas, didapatkan data total label *train* adalah nilai *negative* sebanyak 750 label, kemudian *neutral* adalah 81 dan *positive* adalah sebesar 1041.

id	nama	jenis	kegiatan	tempat	waktu	status
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	4	4	4	4	4	4
5	5	5	5	5	5	5
6	6	6	6	6	6	6
7	7	7	7	7	7	7
8	8	8	8	8	8	8
9	9	9	9	9	9	9
10	10	10	10	10	10	10
11	11	11	11	11	11	11
12	12	12	12	12	12	12
13	13	13	13	13	13	13
14	14	14	14	14	14	14
15	15	15	15	15	15	15
16	16	16	16	16	16	16
17	17	17	17	17	17	17
18	18	18	18	18	18	18
19	19	19	19	19	19	19
20	20	20	20	20	20	20

Gambar 4. 31 Proses Penggabungan Data *Test*

Selanjutnya adalah melakukan penggabungan data *test*, sehingga didapatkan data total label test adalah nilai *negative* sebanyak 322, nilai *neutral* adalah 36 dan *positive* adalah sebanyak 447.

Tahap selanjutnya adalah LSTM Model. Dimana pertama adalah membuat arsitektur LSTM, Arsitektur dari LSTM dapat dilihat pada Gambar 4.32 dibawah ini.

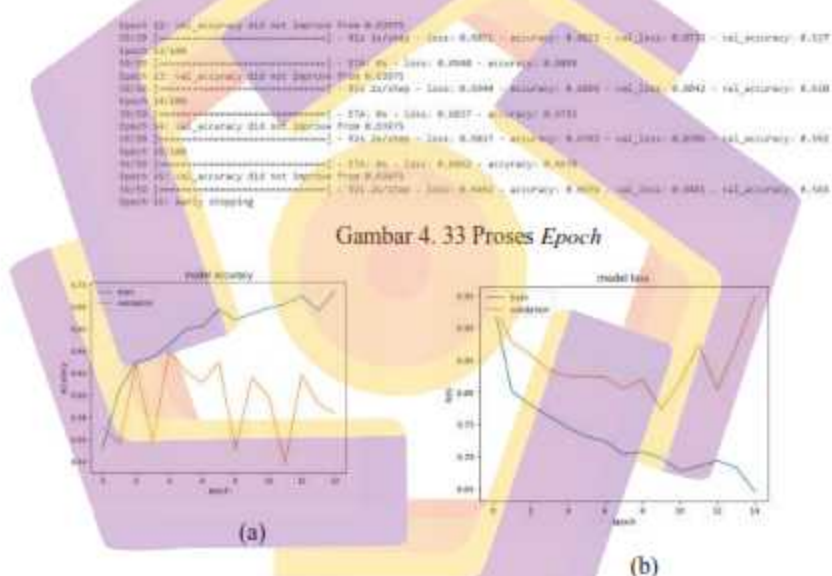
```

Model: "sequential"
Layer (Type)          Output Shape         Param #
-----
embedding (Embedding) (None, 348, 64)      22080
lstm (LSTM)           (None, 348, 128)    98514
dropout (Dropout)     (None, 348, 128)    0
lstm_1 (LSTM)         (None, 128)         131584
batch_normalization (Batch Normalization) (None, 128)         512
dense (Dense)         (None, 64)          8256
dropout_1 (Dropout)   (None, 64)          0
dense_1 (Dense)       (None, 3)           195
-----
Total params: 87930 (3.35 MB)
Trainable params: 22080 (0.84 MB)
Non-trainable params: 64255 (2.44 MB)

```

Gambar 4. 32 Total Parameter

Kemudian model tersebut dilatih dengan data training dan selanjutnya dievaluasi dengan data validasi. Adapun ukuran batch (*batch\_size*) yang digunakan sebanyak 32 dengan epoch sebanyak 100. Setelah dikompilasi, diperoleh nilai *loss* mencapai 0.6462 dengan nilai akurasi sebesar 0.6939. Epoch berhenti secara otomatis di epoch 15 karena menggunakan fungsi *early stopping* guna memberhentikan proses pelatihan data ketika terjadi overfit. Adapun proses epoch dapat dilihat pada Gambar 4.33.



Gambar 4.34 Akurasi Train dan Validation (a), Loss Train dan Validation

#### 4.5.2 Splitting Data 80:20

Pada *splitting* data 80:20 dilakukan perbandingan data *training* sebesar 80% dan data *test* sebesar 20%, dimana pada label negative, nilai train negative adalah 574 dan test negative adalah 144. Kemudian label neutral, nilai train negative adalah 156 dan test negative adalah 39. Terakhir label positive, nilai train negative adalah



469 dan test negative adalah 118. Tahap selanjutnya adalah menggabungkan kembali setiap data untuk data *train* & *test* yang dapat dilihat pada Gambar 4.35 dibawah ini.



nama	jumlah_label	label	jumlah_label	jumlah_label	jumlah_label	jumlah_label
gabungan	1190	positive	1190	1190	1190	1190
gabungan	857	negative	857	857	857	857
gabungan	93	neutral	93	93	93	93

Gambar 4. 35 Proses Penggabungan Data *Train*

Dari proses diatas, didapatkan data total label *train* adalah nilai *negative* sebanyak 857 label, kemudian *neutral* adalah 93 dan *positive* adalah sebesar 1190.



nama	jumlah_label	label	jumlah_label	jumlah_label	jumlah_label	jumlah_label
gabungan	298	positive	298	298	298	298
gabungan	215	negative	215	215	215	215
gabungan	24	neutral	24	24	24	24

Gambar 4. 36 Proses Penggabungan Data *Test*

Selanjutnya adalah melakukan penggabungan data *test*, sehingga didapatkan data total label test adalah nilai *negative* sebanyak 215, nilai *neutral* adalah 24 dan positive adalah sebanyak 298.

Tahap selanjutnya adalah LSTM Model. Dimana pertama adalah membuat arsitektur LSTM, Arsitektur dari LSTM dapat dilihat pada Gambar 4.37 dibawah ini.

Model: "sequential\_2"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 348, 64)	640000
lstm_2 (LSTM)	(None, 348, 128)	98816
dropout_2 (Dropout)	(None, 348, 128)	0
lstm_3 (LSTM)	(None, 128)	131584
batch_normalization_1 (Batch Normalization)	(None, 128)	512
dense_2 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 3)	195

---

Total params: 879363 (3.35 MB)  
 Trainable params: 239107 (934.01 KB)  
 Non-trainable params: 640256 (2.44 MB)

Gambar 4. 37 Total Parameter

Kemudian model tersebut dilatih dengan data training dan selanjutnya dievaluasi dengan data validasi. Adapun ukuran batch (*batch\_size*) yang digunakan sebanyak 32 dengan epoch sebanyak 100. Setelah dikompilasi, diperoleh nilai *loss* mencapai 0.6074 dengan nilai akurasi sebesar 0.7210. Epoch berhenti secara otomatis di epoch 19 karena menggunakan fungsi *early stopping* guna memberhentikan proses pelatihan data ketika terjadi overfit. Adapun proses epoch dapat dilihat pada Gambar 4.38.

```

22/00 [INFO] - val_accuracy: 0.68 | loss: 0.6074 | Time: 0.0277
22/00 [INFO] - 91% 10/100 - loss: 0.6407 - accuracy: 0.7000 - val_loss: 0.6002 - val_accuracy: 0.6910
Epoch 17/100
22/00 [INFO] - 97% 0/1 - loss: 0.6074 - accuracy: 0.7110
22/00 [INFO] - 98% 10/100 - loss: 0.6219 - accuracy: 0.7110 - val_loss: 0.6100 - val_accuracy: 0.6910
Epoch 18/100
22/00 [INFO] - 97% 0/1 - loss: 0.6460 - accuracy: 0.7000
22/00 [INFO] - val_accuracy: 0.68 | loss: 0.6074 | Time: 0.0277
22/00 [INFO] - 91% 10/100 - loss: 0.6409 - accuracy: 0.7000 - val_loss: 0.6011 - val_accuracy: 0.6910
Epoch 19/100
22/00 [INFO] - 97% 0/1 - loss: 0.6074 - accuracy: 0.7110
22/00 [INFO] - 98% 10/100 - loss: 0.6219 - accuracy: 0.7110 - val_loss: 0.6007 - val_accuracy: 0.6910
Epoch 20: early stopping

```

Gambar 4. 38 Proses Epoch



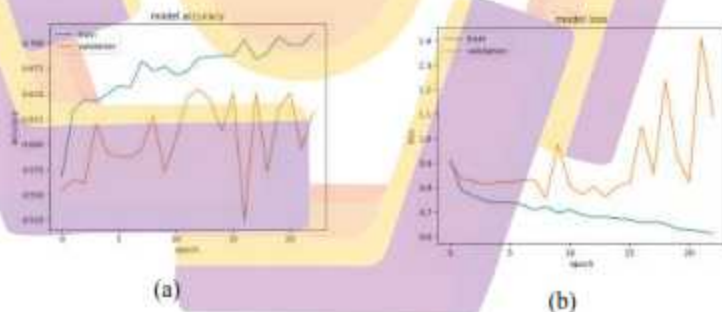


Kemudian model tersebut dilatih dengan data training dan selanjutnya dievaluasi dengan data validasi. Adapun ukuran batch (*batch\_size*) yang digunakan sebanyak 32 dengan epoch sebanyak 100. Setelah dikompilasi, diperoleh nilai *loss* mencapai 0.6130 dengan nilai akurasi sebesar 0.7097. Epoch berhenti secara otomatis di epoch 23 karena menggunakan fungsi *early stopping* guna memberhentikan proses pelatihan data ketika terjadi *overfit*. Adapun proses epoch dapat dilihat pada Gambar 4.43.

```

Epoch 00: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 01: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 02: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 03: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 04: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 05: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 06: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 07: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 08: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 09: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 10: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 11: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 12: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 13: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 14: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 15: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 16: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 17: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 18: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 19: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 20: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 21: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 22: val_accuracy: 0.6130 - loss: 0.6130 -> 100% 0.000s - accuracy: 0.7097 - val_loss: 0.6130 - val_accuracy: 0.6130
Epoch 23: Early stopping
  
```

Gambar 4. 43 Proses Epoch



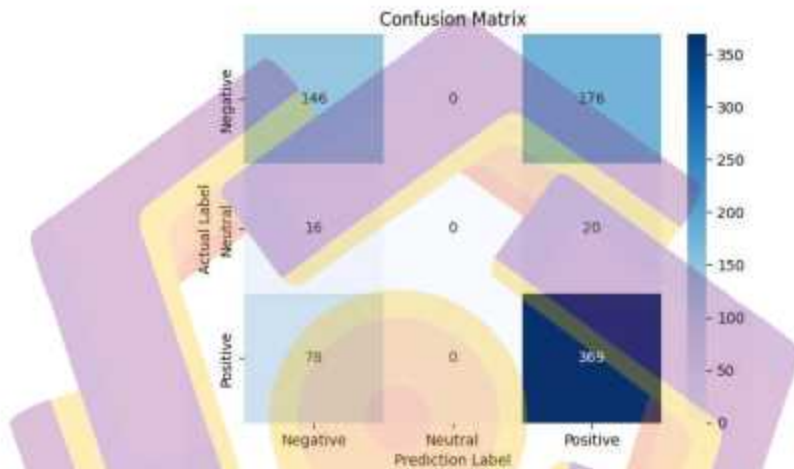
Gambar 4. 44 Akurasi Train dan Validation (a), Loss Train dan Validation



#### 4.6. Evaluasi *Confussion Matrix* Tanpa Word2Vec

##### 4.4.4 Hasil Evaluasi LSTM dengan Splitting Data 70:30

Hasil evaluasi confusion matrix yang diperoleh dari model LSTM dengan *Splitting* Data 70:30 dapat dilihat pada Gambar 4.45.



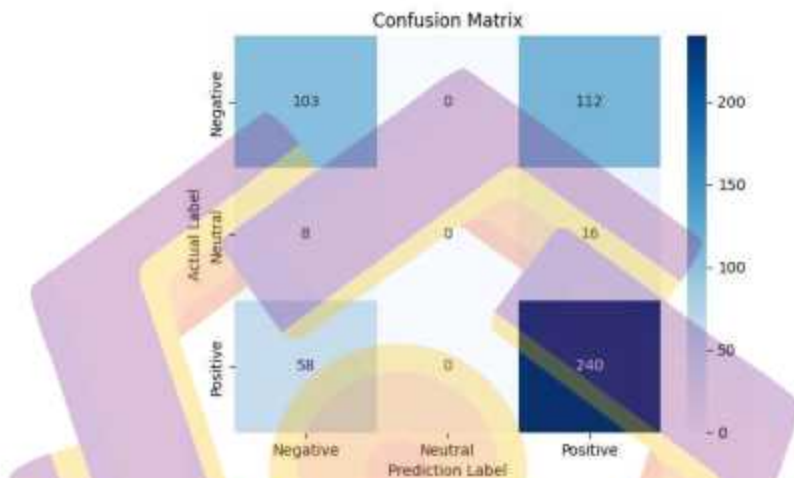
Gambar 4. 45 *Confussion Matrix*.

Berdasarkan proses diatas, didapatkan nilai *accuracy* pada data train dan validation adalah sebesar 64%. Selain itu juga menunjukkan bahwa label 0 (*Negative*) mendapatkan nilai *precision* sebesar 61 %, *recall* sebesar 45 %, *f1-score* sebesar 52 %. Sedangkan pada label 2 (*Positive*) menghasilkan nilai *precision* sebesar 65 %, *recall* sebesar 83 %, *f1-score* sebesar 73 %. Hasil dari proses tersebut dapat dilihat pada Gambar 4.46 dibawah ini.



#### 4.4.5 Hasil Evaluasi LSTM dengan Splitting Data 80:20

Hasil evaluasi confusion matrix yang diperoleh dari model LSTM dengan *Splitting* Data 80:20 dapat dilihat pada Gambar 4.49.



Gambar 4. 49 *Confusion Matrix*

Berdasarkan proses diatas, didapatkan nilai accuracy pada data train dan validation adalah sebesar 64%. Selain itu juga menunjukkan bahwa label 0 (*Negative*) mendapatkan nilai *precision* sebesar 61 %, *recall* sebesar 48 %, *f1-score* sebesar 54 %. Sedangkan pada label 2 (*Positive*) menghasilkan nilai *precision* sebesar 65 %, *recall* sebesar 81 %, *f1-score* sebesar 72 %. Hasil dari proses tersebut dapat dilihat pada Gambar 4.50 dibawah ini.

	precision	recall	f1-score	support
0	0.61	0.48	0.54	215
1	0.00	0.00	0.00	24
2	0.65	0.81	0.72	298
accuracy			0.64	537
macro avg	0.42	0.43	0.42	537
weighted avg	0.61	0.64	0.61	537

Gambar 4. 50 Hasil Perhitungan *Confusion Matrix*

Dari hasil perhitungan diatas, didapatkan hasil evaluasi negative, didapatkan akurasi sebesar 47.91% yang dapat dilihat pada Gambar 4.51 dibawah.

id	target	prediction	label	text	sentiment	category	score	label	text
0	0	0	0	...	...	...	...	0	...
1	0	0	0	...	...	...	...	0	...
2	0	0	0	...	...	...	...	0	...
3	0	0	0	...	...	...	...	0	...
4	0	0	0	...	...	...	...	0	...
5	0	0	0	...	...	...	...	0	...
6	0	0	0	...	...	...	...	0	...
7	0	0	0	...	...	...	...	0	...
8	0	0	0	...	...	...	...	0	...
9	0	0	0	...	...	...	...	0	...
10	0	0	0	...	...	...	...	0	...
11	0	0	0	...	...	...	...	0	...
12	0	0	0	...	...	...	...	0	...
13	0	0	0	...	...	...	...	0	...
14	0	0	0	...	...	...	...	0	...
15	0	0	0	...	...	...	...	0	...
16	0	0	0	...	...	...	...	0	...
17	0	0	0	...	...	...	...	0	...
18	0	0	0	...	...	...	...	0	...
19	0	0	0	...	...	...	...	0	...
20	0	0	0	...	...	...	...	0	...
21	0	0	0	...	...	...	...	0	...
22	0	0	0	...	...	...	...	0	...
23	0	0	0	...	...	...	...	0	...
24	0	0	0	...	...	...	...	0	...
25	0	0	0	...	...	...	...	0	...
26	0	0	0	...	...	...	...	0	...
27	0	0	0	...	...	...	...	0	...
28	0	0	0	...	...	...	...	0	...
29	0	0	0	...	...	...	...	0	...
30	0	0	0	...	...	...	...	0	...
31	0	0	0	...	...	...	...	0	...
32	0	0	0	...	...	...	...	0	...
33	0	0	0	...	...	...	...	0	...
34	0	0	0	...	...	...	...	0	...
35	0	0	0	...	...	...	...	0	...
36	0	0	0	...	...	...	...	0	...
37	0	0	0	...	...	...	...	0	...
38	0	0	0	...	...	...	...	0	...
39	0	0	0	...	...	...	...	0	...
40	0	0	0	...	...	...	...	0	...
41	0	0	0	...	...	...	...	0	...
42	0	0	0	...	...	...	...	0	...
43	0	0	0	...	...	...	...	0	...
44	0	0	0	...	...	...	...	0	...
45	0	0	0	...	...	...	...	0	...
46	0	0	0	...	...	...	...	0	...
47	0	0	0	...	...	...	...	0	...
48	0	0	0	...	...	...	...	0	...
49	0	0	0	...	...	...	...	0	...

Gambar 4. 51 Hasil Evaluasi Label Negative

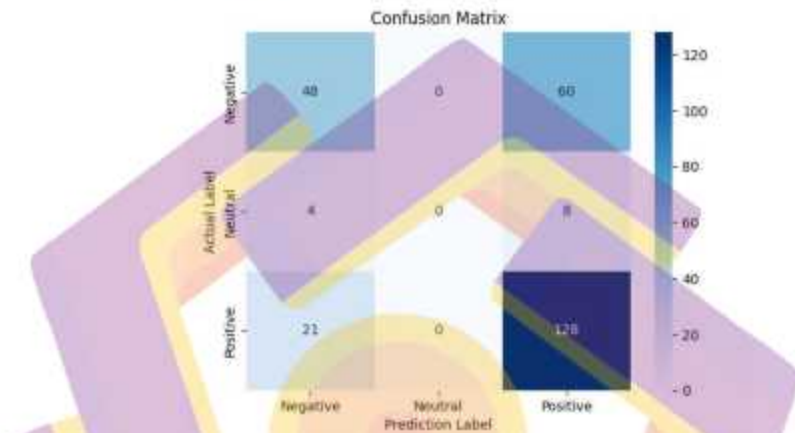
Dan hasil evaluasi positive mendapatkan hasil akurasi sebesar 80.54% yang dapat dilihat pada Gambar 4.52 dibawah ini.

id	target	prediction	label	text	sentiment	category	score	label	text
50	1	1	1	...	...	...	...	1	...
51	1	1	1	...	...	...	...	1	...
52	1	1	1	...	...	...	...	1	...
53	1	1	1	...	...	...	...	1	...
54	1	1	1	...	...	...	...	1	...
55	1	1	1	...	...	...	...	1	...
56	1	1	1	...	...	...	...	1	...
57	1	1	1	...	...	...	...	1	...
58	1	1	1	...	...	...	...	1	...
59	1	1	1	...	...	...	...	1	...
60	1	1	1	...	...	...	...	1	...
61	1	1	1	...	...	...	...	1	...
62	1	1	1	...	...	...	...	1	...
63	1	1	1	...	...	...	...	1	...
64	1	1	1	...	...	...	...	1	...
65	1	1	1	...	...	...	...	1	...
66	1	1	1	...	...	...	...	1	...
67	1	1	1	...	...	...	...	1	...
68	1	1	1	...	...	...	...	1	...
69	1	1	1	...	...	...	...	1	...
70	1	1	1	...	...	...	...	1	...
71	1	1	1	...	...	...	...	1	...
72	1	1	1	...	...	...	...	1	...
73	1	1	1	...	...	...	...	1	...
74	1	1	1	...	...	...	...	1	...
75	1	1	1	...	...	...	...	1	...
76	1	1	1	...	...	...	...	1	...
77	1	1	1	...	...	...	...	1	...
78	1	1	1	...	...	...	...	1	...
79	1	1	1	...	...	...	...	1	...
80	1	1	1	...	...	...	...	1	...
81	1	1	1	...	...	...	...	1	...
82	1	1	1	...	...	...	...	1	...
83	1	1	1	...	...	...	...	1	...
84	1	1	1	...	...	...	...	1	...
85	1	1	1	...	...	...	...	1	...
86	1	1	1	...	...	...	...	1	...
87	1	1	1	...	...	...	...	1	...
88	1	1	1	...	...	...	...	1	...
89	1	1	1	...	...	...	...	1	...
90	1	1	1	...	...	...	...	1	...
91	1	1	1	...	...	...	...	1	...
92	1	1	1	...	...	...	...	1	...
93	1	1	1	...	...	...	...	1	...
94	1	1	1	...	...	...	...	1	...
95	1	1	1	...	...	...	...	1	...
96	1	1	1	...	...	...	...	1	...
97	1	1	1	...	...	...	...	1	...
98	1	1	1	...	...	...	...	1	...
99	1	1	1	...	...	...	...	1	...

Gambar 4. 52 Hasil Evaluasi Label Positive

#### 4.4.6 Hasil Evaluasi LSTM dengan Splitting Data 90:10

Hasil evaluasi *confusion matrix* yang diperoleh dari model LSTM dengan *Splitting* Data 90:10 dapat dilihat pada Gambar 4.53.



Gambar 4. 53 *Confusion Matrix*

Berdasarkan proses diatas, didapatkan nilai *accuracy* pada data *train* dan *validation* adalah sebesar 65%. Selain itu juga menunjukkan bahwa label 0 (*Negative*) mendapatkan nilai *precision* sebesar 66 %, *recall* sebesar 44 %, *f1-score* sebesar 53 %. Sedangkan pada label 2 (*Positive*) menghasilkan nilai *precision* sebesar 65 %, *recall* sebesar 86 %, *f1-score* sebesar 74 %. Hasil dari proses tersebut dapat dilihat pada Gambar 4.54 dibawah ini.

	precision	recall	f1-score	support
0	0.66	0.44	0.53	108
1	0.00	0.00	0.00	12
2	0.65	0.86	0.74	149
accuracy			0.65	269
macro avg	0.44	0.43	0.42	269
weighted avg	0.63	0.65	0.62	269

Gambar 4. 54 Hasil Perhitungan *Confusion Matrix*



Dari hasil perhitungan diatas, didapatkan hasil evaluasi negative, didapatkan akurasi sebesar 44.44% yang dapat dilihat pada Gambar 4.55 dibawah.

Index	Actual Label	Predicted Label	Actual Text	Predicted Text	Actual Label	Predicted Label	Actual Text	Predicted Text	Actual Label	Predicted Label	Actual Text	Predicted Text
0	0	0	...	...	0	0	...	...	0	0	...	...
1	0	1	...	...	0	1	...	...	0	1	...	...
2	0	0	...	...	0	0	...	...	0	0	...	...
3	0	1	...	...	0	1	...	...	0	1	...	...
4	0	0	...	...	0	0	...	...	0	0	...	...
5	0	1	...	...	0	1	...	...	0	1	...	...
6	0	0	...	...	0	0	...	...	0	0	...	...
7	0	1	...	...	0	1	...	...	0	1	...	...
8	0	0	...	...	0	0	...	...	0	0	...	...
9	0	1	...	...	0	1	...	...	0	1	...	...
10	0	0	...	...	0	0	...	...	0	0	...	...
11	0	1	...	...	0	1	...	...	0	1	...	...
12	0	0	...	...	0	0	...	...	0	0	...	...
13	0	1	...	...	0	1	...	...	0	1	...	...
14	0	0	...	...	0	0	...	...	0	0	...	...
15	0	1	...	...	0	1	...	...	0	1	...	...
16	0	0	...	...	0	0	...	...	0	0	...	...
17	0	1	...	...	0	1	...	...	0	1	...	...
18	0	0	...	...	0	0	...	...	0	0	...	...
19	0	1	...	...	0	1	...	...	0	1	...	...

Gambar 4. 55 Hasil Evaluasi Label Negative

Dan hasil evaluasi positive mendapatkan hasil akurasi sebesar 85.91% yang dapat dilihat pada Gambar 4.56 dibawah ini.

Index	Actual Label	Predicted Label	Actual Text	Predicted Text	Actual Label	Predicted Label	Actual Text	Predicted Text	Actual Label	Predicted Label	Actual Text	Predicted Text
0	1	1	...	...	1	1	...	...	1	1	...	...
1	1	1	...	...	1	1	...	...	1	1	...	...
2	1	1	...	...	1	1	...	...	1	1	...	...
3	1	1	...	...	1	1	...	...	1	1	...	...
4	1	1	...	...	1	1	...	...	1	1	...	...
5	1	1	...	...	1	1	...	...	1	1	...	...
6	1	1	...	...	1	1	...	...	1	1	...	...
7	1	1	...	...	1	1	...	...	1	1	...	...
8	1	1	...	...	1	1	...	...	1	1	...	...
9	1	1	...	...	1	1	...	...	1	1	...	...
10	1	1	...	...	1	1	...	...	1	1	...	...
11	1	1	...	...	1	1	...	...	1	1	...	...
12	1	1	...	...	1	1	...	...	1	1	...	...
13	1	1	...	...	1	1	...	...	1	1	...	...
14	1	1	...	...	1	1	...	...	1	1	...	...
15	1	1	...	...	1	1	...	...	1	1	...	...
16	1	1	...	...	1	1	...	...	1	1	...	...
17	1	1	...	...	1	1	...	...	1	1	...	...
18	1	1	...	...	1	1	...	...	1	1	...	...
19	1	1	...	...	1	1	...	...	1	1	...	...

Gambar 4. 56 Hasil Evaluasi Label Positive

#### 4.7. Pembahasan

Dari hasil penelitian yang telah dilakukan, penelitian ini melakukan analisis sentimen pada *review* produk *skincare* menggunakan LSTM dan *word embedding*. *Dataset* yang digunakan pada penelitian ini berasal dari ulasan pengguna pada *skincare* di *female daily network*. Proses analisis yang dilakukan pada penelitian ini setelah pengumpulan data, yaitu dilakukan *preprocessing*, pelabelan menggunakan

*corpus*. Selanjutnya dilakukan proses *word embedding* dan tanpa menggunakan *word embedding*, dan diproses menggunakan algoritma LSTM, dan dilanjutkan dengan evaluasi *confusion matrix*. Berikut pada Tabel 4.1 merupakan hasil akurasi yang didapatkan ketika menggunakan pelabelan berbasis *corpus* dan pelabelan berbasis *rating*.

Tabel 4.1 Perbandingan Hasil Pelabelan

	Proporsi	Pelabelan Corpus (Akurasi)	Pelabelan Rating (Akurasi)
Menggunakan word2vec	90%:10%	75%	91%
	80%:20%	72%	87%
	70%:30%	71%	85%
Tanpa Menggunakan Word2vec	90%:10%	65%	74%
	80%:20%	64%	69%
	70%:30%	64%	73%

Pada Tabel 4.1 menunjukkan perbandingan hasil pengujian yang dilakukan terhadap algoritma LSTM dengan menggunakan word2vec dan tanpa menggunakan word2vec dengan hasil sebagai berikut:

Pada algoritma LSTM menggunakan word2vec dengan proporsi data 90:10 memperoleh hasil akurasi sebesar 75% pada pelabelan menggunakan corpus dan 91% pada pelabelan menggunakan rating. Pada proporsi data 80:20 memperoleh hasil akurasi sebesar 72% pada pelabelan menggunakan corpus dan 87% pada pelabelan menggunakan rating. Sedangkan pada proporsi data 70:30 memperoleh

hasil akurasi sebesar 71% pada pelabelan menggunakan corpus dan 85% pada pelabelan menggunakan rating.

Pada algoritma LSTM tanpa menggunakan word2vec dengan proporsi data 90:10 memperoleh hasil akurasi sebesar 65% pada pelabelan menggunakan corpus dan 74% pada pelabelan menggunakan rating. Pada proporsi data 80:20 memperoleh hasil akurasi sebesar 64% pada pelabelan menggunakan corpus dan 69% pada pelabelan menggunakan rating. Sedangkan pada proporsi data 70:30 memperoleh hasil akurasi sebesar 64% pada pelabelan menggunakan corpus dan 73% pada pelabelan menggunakan rating.

Berdasarkan hasil perbandingan yang telah dilakukan terhadap algoritma LSTM dengan menggunakan word2vec dan tanpa menggunakan word2vec dapat disimpulkan bahwa akurasi terbaik diperoleh pada algoritma LSTM dengan menggunakan word2vec yaitu pada proporsi data 90:10 dengan hasil akurasi sebesar 75% pada pelabelan menggunakan corpus dan 91% pada pelabelan menggunakan rating. Hal tersebut diperoleh karena aspek training dalam LSTM lebih memperhatikan makna konteks dengan urutan kata, dan proses menggunakan word2vec menjadikan pembobotan lebih representative. Apabila dilihat dari hasil yang telah diperoleh bahwasannya proporsi yang menghasilkan akurasi yang lebih baik didapatkan pada 90:10, hal ini didukung berdasarkan penelitian yang dilakukan oleh (Tilasefana & Putra, 2023) menyatakan bahwa semakin besar data training maka semakin tinggi juga ketepatan yang diperoleh.

Selain itu terdapat perbandingan mengenai hasil analisis sentimen dari beberapa penelitian yang menggunakan data yang sama yaitu berasal dari female

daily network dimulai dari tahun 2022. Berikut pada Tabel 4.2 merupakan penjabarannya.

Tabel 4. 2 Perbandingan Hasil Analisis Sentimen

Penulis & Tahun	Hasil dan Fitur yang digunakan dari penelitian sebelumnya	Hasil dan Fitur yang digunakan dari penelitian yang dilakukan
(Hidayat & Handayani, 2022)	Fitur yang digunakan yaitu ulasan produk dan dengan menggunakan fitur ulasan produk penelitian tersebut menghasilkan akurasi sebesar 80,22%. Hasil tersebut diproses menggunakan algoritma <i>ID-Convolutional Neural Network (ID-CNN)</i>	Fitur yang digunakan ada 4 yaitu username, product, rating, dan review. Namun yang digunakan dalam penelitian ini untuk proses labeling menggunakan fitur rating dan fitur review. Hasil akurasi tertinggi menggunakan corpus sebesar 75%, sedangkan menggunakan rating 91%. Hasil tersebut diproses menggunakan algoritma LSTM menggunakan Word2Vec
(Astuti & Astuti, 2022)	Fitur yang digunakan yaitu ada 4 seperti user, date,	Fitur yang digunakan ada 4 yaitu username, product,

	<p>review, dan rating. Namun yang digunakan untuk labeling dalam penelitian tersebut yaitu fitur review dan penelitian tersebut menghasilkan akurasi sebesar 79.85%. Hasil tersebut diproses menggunakan metode <i>naïve bayes classifier</i> berbasis <i>Particle Swarm Optimization (PSO)</i></p>	<p>rating, dan review. Namun yang digunakan dalam penelitian ini untuk proses labeling menggunakan fitur rating dan fitur review. Hasil akurasi tertinggi menggunakan corpus sebesar 75%, sedangkan menggunakan rating 91%. Hasil tersebut diproses menggunakan algoritma LSTM menggunakan Word2Vec</p>
--	---	---



## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan proses implementasi dan pembahasan yang telah dilakukan diatas, dapat disimpulkan sebagai berikut:

1. Analisis sentimen terhadap produk skincare berhasil dilakukan dengan menggunakan algoritma LSTM dan *word embedding* Word2Vec dan dibangun dengan menggunakan bahasa pemrograman *python*.
2. Penggunaan Word2vec pada LSTM memiliki pengaruh terhadap analisis sentimen produk skincare. Hal ini dikarenakan aspek training dalam LSTM lebih memperhatikan makna konteks dengan urutan kata, dan proses menggunakan word2vec menjadikan pembobotan lebih representatif.
3. Berdasarkan hasil pengujian yang telah dilakukan dengan menggunakan *confusion matrix* didapatkan nilai akurasi dari metode algoritma LSTM dengan menggunakan word2vec dengan data training sebesar 90% dan testing sebesar 10% didapatkan nilai akurasi sebesar 75%, pada model LSTM dengan data training sebesar 80% dan testing sebesar 20% didapatkan nilai akurasi sebesar 72% dan pada pada model LSTM dengan data training sebesar 70% dan testing sebesar 30% didapatkan nilai akurasi sebesar 71%. Sedangkan pada metode algoritma LSTM dengan tidak menggunakan word2vec pada data training sebesar 90% dan testing sebesar 10% didapatkan nilai akurasi sebesar 65%, pada Pada model LSTM dengan

data training sebesar 80% dan testing sebesar 20% didapatkan nilai akurasi sebesar 64% dan pada pada model LSTM dengan data training sebesar 70% dan testing sebesar 30% didapatkan nilai akurasi sebesar 64%.

## 5.2. Saran

Dari kesimpulan tersebut, dapat diberikan saran yaitu diharapkan untuk pengembangan penelitian berikutnya dapat melakukan perbandingan dengan metode lain atau meningkatkan akurasi pada algoritma LSTM yang sebelumnya.



## DAFTAR PUSTAKA

- Afida, D. I., Dairoh, Handayani, S. F., & Pratiwi, R. W. (2021). Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen. *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*, 6.
- Amin, S., Uddin, M. I., Zeb, M. A., Alarood, A. A., Mahmoud, M., & Alkinani, M. H. (2020). Detecting Dengue/Flu Infections Based on Tweets Using LSTM and Word Embedding. *IEEE*.
- Ardian, H., & Kosasi, S. (2019). Analisis Sentimen Pada Review Produk Kosmetik Bahasa Indonesia Dengan Metode Naive Bayes. *Jurnal ENTER*.
- Arissinta, I. O., Sulistiyawati, I. D., & Kumianto, D. (2022). Pemodelan Time Series untuk Peramalan Web Traffic Menggunakan Algoritma Arima, LSTM, dan GRU. *Prisma, Prosiding Seminar Nasional Matematika*, 693-700.
- Astuti, T., & Astuti, Y. (2022). Analisis Sentimen Review Produk Skincare Dengan Naive Bayes Classifier Berbasis Particle Swarm Optimization (PSO). *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 1806-1815.
- Astuti, T., & Astuti, Y. (2022). Analisis Sentimen Review Produk Skincare Dengan Naive Bayes Classifier Berbasis Particle Swarm Optimization (PSO). *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 6(4), pp. 1806-1815.
- Darmawan, M. D., & Laily, N. (2022). PENGARUH ONLINE CUSTOMER REVIEW DAN RATING TERHADAP MINAT BELI KONSUMEN MARKETPLACE TOKOPEDIA MELALUI TRUST SEBAGAI VARIABEL INTERVENINGNYA. *Jurnal Ilmu dan Riset Manajemen*, 11(6), 1-17.
- Fadly, Marlina, D., Kurniawan, T. B., Zakaria, M. Z., & Farahnasihah, S. (2022). Sentiment Analysis on Natural Skincare Products. *JOURNAL OF DATA SCIENCE*.
- Female Daily. (2022). *FEMALE DAILY*. Retrieved from FEMALE DAILY: <https://femaledaily.com/>

- Fitriani, R. D., Yasin, H., & Tarno. (2021). PENANGANAN KLASIFIKASI KELAS DATA TIDAK SEIMBANG DENGAN RANDOM OVERSAMPLING PADA NAIVE BAYES (Studi Kasus: Status Peserta KB IUD di Kabupaten Kendal). *JURNAL GAUSSIAN*, 10(1), 11-20.
- Fritama, S. D., Ramadhan, Y. R., & Komara, M. A. (2023). Analisis Sentimen Review Produk Acne Spot Treatment di Female Daily. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 4(1), 134-143.
- Hanafi, & Aboob, a. M. (2021). Word Sequential Using Deep LSTM and Matrix Factorization to Handle Rating Sparse Data for E-Commerce Recommender System. *Hindawi (Computational Intelligence and Neuroscience)*.
- Hendrawan, I. R. (2022). Perbandingan algoritma naive bayes, SVM dan XGBoost dalam Klasifikasi Teks Sentimen Masyarakat Terhadap Produk Lokal di Indonesia. *Jurnal Transformasi (Informasi & Pengembangan IPTEK)*, 18, 1-8.
- Hidayat, E. Y., & Handayani, D. (2022). Penerapan ID-CNN untuk analisis sentimen ulasan produk kosmetik berdasar female daily review. *Jurnal Nasional teknologi dan Sistem Informasi*, 153-163.
- Ihsan, M., Negara, B. S., & Agustian, S. (2022). Metode LSTM (Long short term memory) untuk Klasifikasi Sentimen Vaksin Covid-19 pada Twitter. *Jurnal Teknologi Informasi & Komunikasi Digital Zone*.
- Ikhsania, A. A. (2020, Desember 21). *Skincare Adalah Asupan Nutrisi yang Dapat Menjaga Kesehatan Kulit Menjaga Kesehatan Kulit*. Retrieved from SehatQ: <https://www.sehatq.com/artikel/arti-skincare-serta-manfaat-dan-macam-macamnya-sudah-tahu>
- Kang, H. J., Chen, T., Chandrasekaran, M. K., & Kan, M.-Y. (2017). A Comparison of Word Embeddings for English and Cross-Lingual Chinese Word Sense Disambiguation. *arxiv.org*.
- Khairunnisa, S., Adiwijaya, & Faraby, S. A. (2021). Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19) . *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(2), Pp. 406-414.



- Monika, R., Deivalakshmi, S., & Janet, B. (2019). Sentiment Analysis of US Airlines Tweets using LSTM/RNN. *International Conference on Advanced Computing (IACC)*.
- Musafar, L. O., & Y. Yatini, C. (2021). Pre-Processing Data Magnetometer Satelit LAPAN-A3 (Pre-Processing of LAPAN-A3 Satellite's Magnetometer Data). *Jurnal Sains Dirgantara*, 1-12.
- Nabila, A. (2022). Analisis Sentimen Ulasan Produk Toner Pada Beauty Brand "The Body Shop" Menggunakan Metode Naïve Bayes Classifier Dan Support Vector Machine: Studi Kasus Di Female Daily.
- Nejjari, M., & Meziane, A. (2020). SAHAR-LSTM: An enhanced Model for Sentiment Analysis of Hotels' Arabic Reviews based on LSTM. *IEEE*.
- Nurrohmah, M. A., & Azhari, S. N. (2019). Sentiment Analysis of Novel Review Using Long Short-Term Memory Method. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(3), pp. 209-218.
- Nurrohmah, M. A., & SN, A. (2019). Sentiment Analysis of Novel Review Using Long Short-Term Memory Method. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 209-218.
- Nurvania, J., Jondri, & Lhaksamana, K. M. (2021). Analisis Sentimen Pada Ulasan di TripAdvisor Menggunakan Metode Long Short-Term Memory (LSTM). *e-Proceeding of Engineering*.
- Nurvania, J., Jondri, & Lhaksamana, K. M. (2021). Analisis Sentimen Pada Ulasan di TripAdvisor Menggunakan Metode Long Short-Term Memory (LSTM). *e-Proceeding of Engineering*, 8(4), pp. 4124-4135.
- Pane, S. F., & Ramdan, J. (2022). Pemodelan Machine Learning : analisis sentimen masyarakat terhadap kebijakan PPKM menggunakan data twitter. *Jurnal Sistem Cerdas*, 12-20.
- Pratiwi, R. W., H, S. F., Dairoh, Af'idah, D. I., A, Q. R., & F, A. G. (2021). Analisis Sentimen Pada Review Skincare Female Daily Menggunakan Metode Support Vector Machine (SVM). *Journal of Informatics, Information System, Software Engineering and Applications*.



- Putri, D. S., & Ridwan, T. (2023). Analisis Sentimen Ulasan Aplikasi Pospay dengan Algoritma Support Vector Machine. *Jurnal Ilmiah Informatika*, 11(1), 32-41.
- Rachman, F. F., & Pramana, S. (2020, Desember). Analisis Sentimen Pro dan Kontra Masyarakat Indonesia tentang Vaksin COVID-19 pada Media Sosial Twitter. *Indonesian of Health Information Management Journal*, 8(2), 100-109.
- Rahman, M. Z., & Sari, Y. A. (2021). Analisis Sentimen Tweet COVID-19 menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(11), 5120-5127.
- Ridwan, M., & Muzakir, A. (2022). Model Klasifikasi Ujaran Kebencian pada Data Twitter dengan Menggunakan CNN-LSTM. *TEKNOMATIKA*, 12.
- Riyantoko, P. A., Fahrudin, T. M., Maulida, K., Hindrayani, & Safitr, E. M. (2020). ANALISIS PREDIKSI HARGA SAHAM SEKTOR PERBANKAN MENGGUNAKAN ALGORITMA LONG-SHORT TERMS MEMORY (LSTM) . *Seminar Nasional Informatika 2020 (SEMNASIF 2020)*, 427-435.
- Sandag, G. A., & Waworundeng, J. (2022). Analisis Sentimen Masyarakat Terhadap Exchange Tokocrypto Pada Twitter Menggunakan Metode LSTM. *Cogito Smart Journal*, 8(2), pp. 411-421.
- Sanusi, R., Astuti, F. D., & Buryadi, I. Y. (2021). ANALISIS SENTIMEN PADA TWITTER TERHADAP PROGRAM KARTU PRA KERJA DENGAN RECURRENT NEURAL NETWORK. *JIKO (Jurnal Informatika dan Komputer)*, 5(2), pp. 89-97.
- Satriyo, T., Yuliani, D. T., & Nisa, K. (2021). SENTIMENT EMBEDDINGS DOC2VEC PADA KLASIFIKASI KELUHAN POLUSI UDARA. *SURYA INFORMATIKA*, 9(1), 1-15.
- Setiawan, E. I., Ferdianto, A., Santoso, J., Kristian, Y., Gunawan, Sumpeno, S., & Purnomo, M. H. (2020). Analisis Pendapat Masyarakat terhadap Berita

- Kesehatan Indonesia menggunakan Pemodelan Kalimat berbasis LSTM. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 9(1), pp. 8-17.
- Siswa, T. A., & Prihandoko. (2018). PERBANDINGAN KINERJA ALGORITMA C4.5, NAÏVE BAYES, K-NEAREST NEIGHBOR, LOGISTIC REGRESSION, DAN SUPPORT VECTOR MACHINES UNTUK MENDETEKSI PENYAKIT KANKER PAYUDARA. *Jurnal Teknologi Informasi dan Komunikasi*, 7(2), 1-10.
- Tilasefana, R. A., & Putra, R. E. (2023). Penerapan Metode Deep Learning Menggunakan Algoritma CNN Dengan Arsitektur VGG NET Untuk Pengenalan Cuaca. *Journal of Informatics and Computer Science*, 48-58.
- Widayat, W. (2021). Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5, 1018-1026.
- Widayat, W. (2021). Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(3), pp. 1018-1026.
- Yahyadi, A., & Latifah, F. (2022). ANALISIS SENTIMEN TWITTERTERHADAP KEBIJAKAN PPKM DI TENGAH PANDEMI COVID-19MENGUNAKAN MODE LSTM. *JISAMAR (Journal of Information System, Applied, Management, Accounting and Research.)*, 6(2), pp.464-471.