

**IMPLEMENTASI MODEL *HYBRID CONVOLUTIONAL NEURAL NETWORK* DAN *BIDIRECTIONAL LONG SHORT-TERM MEMORY* UNTUK ANALISIS TINGKAT KEPUASAN PENGGUNA APLIKASI HALODOC**

**(Skripsi)**

**Oleh**

**ARIF SU'ADMAJI**



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

## ABSTRAK

### IMPLEMENTASI MODEL *HYBRID CONVOLUTIONAL NEURAL NETWORK* DAN *BIDIRECTIONAL LONG SHORT-TERM MEMORY* UNTUK ANALISIS TINGKAT KEPUASAN PENGGUNA APLIKASI HALODOC

Oleh

ARIF SU'ADMAJI

Perkembangan teknologi mengakibatkan terjadinya kemajuan teknologi sehingga memunculkan berbagai jenis layanan yang memudahkan akses terhadap fasilitas pelayanan termasuk pelayanan kesehatan dan mendorong pengembangan berbagai jenis aplikasi. Aplikasi penyedia layanan kesehatan pilihan masyarakat salah satunya yaitu aplikasi Halodoc. Ulasan dan rating dapat menjadi informasi bagi pengguna lainnya mengenai kualitas pelayanan dari aplikasi Halodoc. Oleh karena itu, diperlukan cara yang tepat untuk menganalisis kepuasan pelanggan terhadap pelayanan aplikasi Halodoc. Metode Deep learning yang dapat digunakan untuk melakukan klasifikasi data teks adalah Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), dan Bidirectional Long Short-Term Memory (BiLSTM). Penelitian ini menerapkan resampling data menggunakan random oversampling, word embedding menggunakan Word2Vec, dan hyperparameter tuning menggunakan Grid Search. Setelah dilakukan pengujian hyperparameter tuning didapatkan nilai learning rate dan nilai batch size terbaik, yaitu batch size dengan nilai 64 dan learning rate dengan nilai  $1 \times 10^{-3}$ . Secara umum arsitektur model hybrid CNN-BiLSTM yang dibangun dalam penelitian ini yaitu lapisan embedding, lapisan konvolusi, lapisan pooling, lapisan BiLSTM, flatten, lapisan fully connected, dan lapisan output. Hasil perhitungan menunjukkan bahwa tingkat accuracy dari model hybrid CNN-BiLSTM yang telah dibangun mampu mengklasifikasikan rating dari ulasan pengguna aplikasi Halodoc dengan keakuratan mencapai 98,71% saat COVID-19 dan 98,16% setelah COVID-19.

**Kata Kunci:** *Hybrid CNN-BiLSTM, Deep Learning, Text Mining, Halodoc.*

## **ABSTRACT**

### **IMPLEMENTATION OF A HYBRID CONVOLUTIONAL NEURAL NETWORK AND BIDIRECTIONAL LONG SHORT-TERM MEMORY MODEL FOR ANALYZING THE LEVEL OF USER SATISFACTION OF THE HALODOC APPLICATION**

**By**

**ARIF SU'ADMAJI**

Technological developments have resulted in technological advances that have led to various types of services that facilitate access to service facilities, including health services, and encourage the development of various types of applications. One of the health service provider applications chosen by the community is the Halodoc application. Reviews and ratings can be information for other users regarding the quality of service from the Halodoc application. Therefore, an appropriate way is needed to analyze customer satisfaction with Halodoc application services. Deep learning methods that can be used to classify text data are Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BiLSTM). This research applies data resampling using random oversampling, word embedding using Word2Vec, and hyperparameter tuning using Grid Search. After testing the hyperparameter tuning, the best learning rate and batch size value are obtained, namely batch size with a value of 64 and learning rate with a value of  $1 \times 10^{-3}$ . In general, the architecture of the CNN-BiLSTM hybrid model built in this study is the embedding layer, convolutional layer, pooling layer, BiLSTM layer, flatten, fully connected layer, and output layer. The calculation results show that the accuracy level of the CNN-BiLSTM hybrid model that has been built is able to classify ratings from Halodoc application user reviews with accuracy reaching 98.71% during COVID-19 and 98.16% after COVID-19.

**Keywords:** Hybrid CNN-BiLSTM, Deep Learning, Text Mining, Halodoc.

**IMPLEMENTASI MODEL *HYBRID CONVOLUTIONAL NEURAL NETWORK* DAN *BIDIRECTIONAL LONG SHORT-TERM MEMORY* UNTUK ANALISIS TINGKAT KEPUASAN PENGGUNA APLIKASI HALODOC**

Oleh  
**ARIF SU'ADMAJI**  
**2017031039**

**Skripsi**

Sebagai Salah Satu Syarat untuk Mencapai Gelar  
**SARJANA MATEMATIKA**

Pada

Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Lampung



**JURUSAN MATEMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS LAMPUNG**  
**BANDAR LAMPUNG**  
**2024**

Judul Skripsi : **IMPLEMENTASI MODEL *HYBRID CONVOLUTIONAL NEURAL NETWORK* DAN *BIDIRECTIONAL LONG SHORT-TERM MEMORY* UNTUK ANALISIS TINGKAT KEPUASAN PENGGUNA APLIKASI HALODOC**

Nama Mahasiswa : **Arif Su'admaji**

Nomor Pokok Mahasiswa : **2017031039**

Jurusan : **Matematika**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

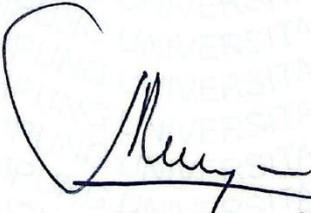


1. **Komisi Pembimbing**

  
**Dr. Dian Kurniasari, S.Si., M.Sc.**  
NIP. 196903051996032001

  
**Favorisen R. Lumbanraja, M.Si., Ph.D.**  
NIP. 198301102008121002

2. **Ketua Jurusan Matematika**

  
**Dr. Aang Nuryaman, S.Si., M.Si.**  
NIP. 197403162005011001

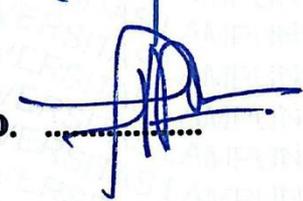
**MENGESAHKAN**

**1. Tim Penguji**

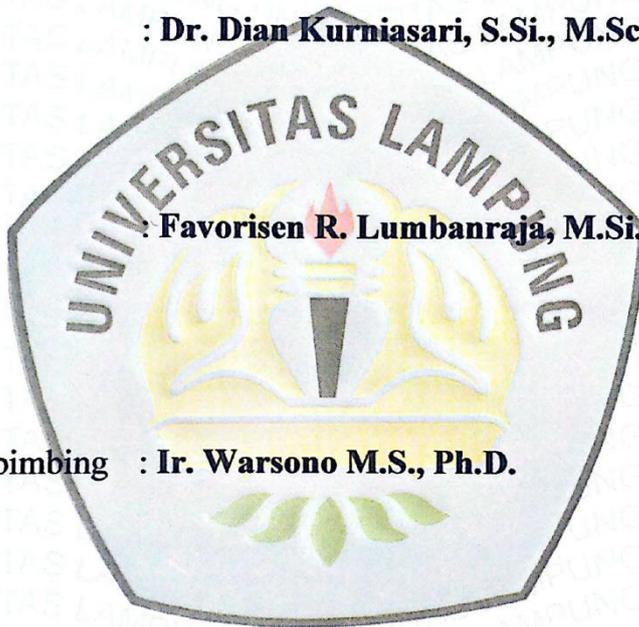
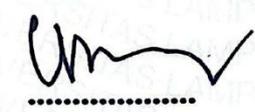
**Ketua : Dr. Dian Kurniasari, S.Si., M.Sc.**



**Sekretaris : Favorisen R. Lumbanraja, M.Si., Ph.D.**



**Penguji  
Bukan Pembimbing : Ir. Warsono M.S., Ph.D.**



**2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam**



**Dr. Eng. Heri Satria, S.Si., M.Si.**  
NIP. 197110012005011002

**Tanggal Lulus Ujian Skripsi: 14 Mei 2024**

## PERNYATAAN SKRIPSI MAHASISWA

Saya yang bertanda tangan di bawah ini:

Nama : **Arif Su'admaji**

Nomor Pokok Mahasiswa : **2017031039**

Jurusan : **Matematika**

Judul Skripsi : **IMPLEMENTASI MODEL *HYBRID*  
*CONVOLUTIONAL NEURAL NETWORK* DAN  
*BIDIRECTIONAL LONG SHORT-TERM*  
*MEMORY* UNTUK ANALISIS TINGKAT  
KEPUASAN PENGGUNA APLIKASI  
**HALODOC****

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri dan semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah karya penulisan ilmiah Universitas Lampung.

Bandar Lampung, 17 Mei 2024  
Penulis



**Arif Su'admaji**  
**NPM. 2017031039**

## **RIWAYAT HIDUP**

**Arif Su'admaji** lahir di Way Kanan, Lampung pada 17 Desember 2001. Anak kedua dari empat bersaudara dari pasangan Bapak Semanudi dan Wanti.

Menempuh pendidikan pertama di Taman Kanak-Kanak Pelangi Desa Tanjung Kurung pada tahun 2007-2008 dan melanjutkan pendidikan dasar di Sekolah Dasar (SD) Negeri 1 Beringin Jaya, Desa Beringin Jaya pada tahun 2008-2014. Selanjutnya, pendidikan menengah dilanjutkan di Sekolah Menengah Pertama (SMP) Negeri 1 Rebang Tangkas pada tahun 2014-2017. Kemudian, Sekolah Menengah Atas (SMA) Negeri 1 Kasui pada tahun 2017-2020. Setelah itu, diterima sebagai mahasiswa Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) pada tahun 2020.

Selama menjadi mahasiswa, penulis aktif di beberapa kegiatan mahasiswa di antaranya sebagai berikut:

1. Pada tahun 2020 penulis aktif menjadi anggota magang Unit Kegiatan Mahasiswa (UKM) Sains dan Teknologi (Saintek), UKMF Natural, dan Himpunan Mahasiswa Matematika (HIMATIKA) FMIPA Unila
2. Pada Tahun 2021 penulis menjadi pengurus HIMATIKA Unila dan menjabat sebagai anggota bidang Keilmuan.
3. Pada Tahun 2021-2022 penulis menjadi pengurus UKMF Natural FMIPA Unila dan menjabat sebagai Videografer bidang Multimedia.

Selain aktif dalam kegiatan mahasiswa, penulis juga aktif dalam kegiatan kampus. Pada tahun 2022 Penulis mengikuti Kompetisi Satria Data dan memperoleh skor

akurasi 99% dengan hasil karya yaitu “Klasifikasi Status Pulang Peserta Berdasarkan Data FKTP BPJS Kesehatan dengan Menggunakan Metode *Artificial Neural Network* (ANN) dan *Extreme Gradient Boosting* (XGBOOST)”. Kemudian kembali mengikuti kompetisi Satria Data pada tahun 2023 dengan hasil karya yaitu “*License Plate Recognition using Optical Character Recognition* (OCR)”. Pada Tahun 2023 penulis melakukan Kerja Praktik (KP) di Dinas Sosial Provinsi Lampung dan Kuliah Kerja Nyata (KKN) di Desa padang Manis, Kecamatan Way Lima, Kabupaten Pesawaran selama 40 hari.

## **KATA INSPIRASI**

*"Sesungguhnya, setelah kesulitan itu ada kemudahan."*

**(Q.S. Al-Insyirah: 5-6)**

*"Sesungguhnya, Allah tidak akan mengubah keadaan suatu kaum hingga mereka mengubah keadaan yang ada pada diri mereka sendiri."*

**(Q.S. Ar-Ra'd: 11)**

*"Sesungguhnya salatku, ibadahku, hidupku, dan matiku hanyalah untuk Allah,  
Tuhan semesta alam."*

**(Q.S. Al-An'am : 162)**

*"Meskipun memiliki seribu alasan untuk menyerah pada suatu tujuan, setidaknya carilah satu saja alasan untuk tetap bertahan."*

**(Penulis)**

*"Sukses bukan tentang kenyamanan tapi tentang ketidaknyamanan yang harus dilakukan, maka nyamanlah dengan ketidaknyamanan."*

**(Penulis)**

## **PERSEMBAHAN**

Alhamdulillah, puji dan syukur kepada **Allah SWT** atas nikmat serta hidayahnya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya.

Oleh karena itu, dengan rasa syukur dan bahagia saya persembahkan rasa terimakasih saya kepada :

### **Bapak Semanudi dan Ibu Wanti**

Tidak ada satupun kata yang dapat mewakili perasaanku pada kalian. Terima kasih kepada kedua orang tuaku atas segala pengorbanan, cinta, dan kasih sayang yang kalian berikan kepadaku. Berkat do'a dan sujud kalian, Allah kuatkan serta mudahkan setiap jalan hidupku. Terima kasih untuk setiap keikhlasan, kesabaran, dan keringat jerih payah yang kalian lakukan. Maafkan anakmu ini yang tak jarang membuat kalian bersusah payah menjatuhkan keringat dan menangis dalam do'a. Terimakasih untuk pelukan hangat yang kalian berikan untukku. Tetaplah bersamaku dan bimbing aku. Aku membutuhkan tangan mu untuk menghapus air mataku. Seandainya kalian dapat membaca ini, kalian adalah ayah dan ibu terbaik yang Allah kirimkan untukku. Aku tau bahwa saat ini kalian sedang berjuang dan berusaha untuk anakmu ini.

Teruntuk ibuku, di saat jauh denganmu aku rindu tidur di pangkuanmu dengan belaian lembut mu. Teruntuk ayahku maafkan aku jika keberadaanku belum membanggakan mu. Untuk itu, terimalah hadiah kecil ini yang kupersembahkan untukmu ayah dan ibu.

### **Dosen Pembimbing dan Pembahas**

Terima kasih kepada dosen pembimbing dan pembahas yang sudah sangat membantu, memberikan motivasi, memberikan masukan, dan ilmu yang berharga. Terkhusus kepada Ibu Dr. Dian Kurniasari S.Si., M.Sc. selaku pembimbing dalam penulisan penelitian ini. Terimakasih telah banyak memberikan kesempatan kepada penulis untuk selalu berkembang, layaknya seorang ibu yang selalu mendidik dan menguatkan anak-anaknya.

Terimakasih ibu.

**Almamater Tercinta Universitas Lampung**

## SANWACANA

*Assalamu'alaikum warahmatullahi wabarakatuh*

Puji syukur saya haturkan kehadiran Allah SWT yang telah melimpahkan nikmat, rahmat, hidayah, serta pertolongan-Nya kepada penulis sehingga skripsi ini dapat diselesaikan.

Skripsi dengan judul “**Implementasi Model *Hybrid Convolutional Neural Network* dan *Bidirectional Long Short-Term Memory* untuk Analisis Tingkat Kepuasan Pengguna Aplikasi Halodoc**” dibuat sebagai bentuk pertanggungjawaban penulis selama menempuh pendidikan S1 dan merupakan salah satu syarat untuk memperoleh gelar Sarjana Matematika (S.Mat.) di Universitas Lampung.

Penulis menyadari bahwa selama proses penulisan skripsi ini masih jauh dari kata sempurna. Proses penyusunan skripsi ini tentu tidak luput dari pengarahan, kritik, saran, dukungan, serta bimbingan dari berbagai pihak sehingga dapat terselesaikan pada waktu yang tepat. Dalam kesempatan ini, penulis menyampaikan rasa hormat dan ucapan terima kasih kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc. selaku dosen pembimbing I yang senantiasa membimbing, memberi masukan serta saran serta mendukung penulis dalam menyelesaikan skripsi ini. Terima kasih telah memberikan banyak kesempatan bagi penulis untuk berkembang, selalu sabar mendidik penulis, dan tulus memberikan ilmu serta wawasan bagi penulis.

2. Bapak Favorisen R. Lumbanraja, S.Kom., M.Si., Ph.D. selaku dosen pembimbing II yang telah memberikan bimbingan, pengarahan, serta saran sehingga penulis dapat menyelesaikan skripsi ini.
3. Bapak Ir. Warsono, M.S., Ph.D. selaku dosen penguji yang telah memberikan kritik dan saran yang membangun sehingga skripsi ini dapat diselesaikan.
4. Bapak Tiryono, M.Sc., Ph.D. selaku dosen pembimbing akademik yang telah memberikan bimbingan dan arahan selama masa perkuliahan.
5. Seluruh Dosen Jurusan Matematika FMIPA Universitas Lampung yang telah memberikan ilmu yang bermanfaat selama di bangku perkuliahan.
6. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
7. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
8. Bapak, Ibu, adik-adik, beserta keluarga besar yang selalu memberikan semangat dukungan, dan do'a kepada penulis.
9. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M. selaku rektor Universitas Lampung.
10. Kedua orang tuaku bapak Semanudi dan ibu Wanti yang selalu mendukung di setiap langkahku.
11. Saudara-saudaraku, kakak Nur Husna, adik-adikku Aan Nataputra dan Aldy Kurniawan yang telah memberikan semangat bagi penulis.
12. Seluruh keluarga besar yang tidak dapat disebutkan satu-persatu.
13. Teman-teman saat kuliah (Muhtarom, Ziyad, Rahmat, Hanafi, Wira, Nurul, Agis, Sephira, Prisca, Mawar, Devanisa, Atip) dan semuanya yang tidak dapat disebutkan satu-persatu.
14. Terkhusus untuk teman seperjuangan Muhtarom yang selalu menjadi sahabat bagi penulis dalam menjalani proses perkuliahan.
15. UKM HIMATIKA Unila khususnya teman-teman bidang keilmuan (Faris, Abdi, Nanda, Bidari, Chintya, Micelle, Adel, Fegy, Yulian, Deta, dan semuanya) yang telah memberikan pengalaman terbaik selama penulis menjalani perkuliahan.

16. UKMF Natural FMIPA Unila yang telah menjadi tempat untuk berkembang di bidang non-akademik bagi penulis.
17. Teman-teman Kelas C Matematika Angkatan 2020 yang menjadi rekan seperjuangan selama perkuliahan.
18. Teman-teman seperjuangan skripsi (Ziyad, Muhtarom, Tama, Nanda, Harum, Yazid, dan Wais) yang menjadi rekan seperjuangan bersama penulis.
19. Teman-teman KKN Desa Padang Manis (Dani, Dini, Nova, Suci, Willis, dan Vanny) yang telah menjadi keluarga, sahabat, dan saudara untuk seterusnya.
20. Seluruh Warga Desa Padang Manis, Bapak Hendri Kurniawan selaku kepala desa dan seluruh perangkat desa, karang taruna, serta anak-anak murid di SD N 9 Way Lima di Desa Padang Manis yang telah memberikan semangat dan dukungan bagi penulis.
21. Wira Adiguna Sutawa dan Ditha Ayu Pratiwi sebagai teman seperjuangan selama pelaksanaan kerja praktik.
22. Teman-teman seperjuangan Matematika Angkatan 2020 yang namanya tidak bisa disebutkan satu per satu, terimakasih untuk rasa kekeluargaan yang terjalin selama ini.
23. Seluruh pihak terkait yang telah banyak membantu dan tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini. Oleh karena itu, penulis mengharapkan masukan serta saran yang akan sangat bermanfaat menjadi pembelajaran kedepannya.

Bandar Lampung, 17 Mei 2024

Penulis

**Arif Su'admaji**

**NPM. 2017031039**

## DAFTAR ISI

halaman

<b>DAFTAR TABEL</b> .....	<b>xix</b>
---------------------------	------------

<b>DAFTAR GAMBAR</b> .....	<b>xx</b>
----------------------------	-----------

<b>I. PENDAHULUAN</b> .....	<b>1</b>
-----------------------------	----------

1.1 Latar Belakang dan Masalah .....	1
--------------------------------------	---

1.2 Rumusan Masalah .....	3
---------------------------	---

1.3 Tujuan Penelitian .....	4
-----------------------------	---

1.4 Manfaat Penelitian .....	4
------------------------------	---

<b>II. TINJAUAN PUSTAKA</b> .....	<b>5</b>
-----------------------------------	----------

2.1 Penelitian Terkait.....	5
-----------------------------	---

2.2 <i>Sentiment Analysis</i> .....	8
-------------------------------------	---

2.3 <i>Natural Language Processing (NLP)</i> .....	8
--	---

2.4 <i>Text Mining</i> .....	9
------------------------------	---

2.5 Klasifikasi Teks .....	10
----------------------------	----

2.6 <i>Resampling Data</i> .....	11
----------------------------------	----

2.6.1 <i>Random Oversampling</i> .....	11
--	----

2.6.2 <i>Random Undersampling</i> .....	12
---	----

2.7 <i>Word Embedding</i> .....	12
---------------------------------	----

2.8 Fungsi Aktivasi.....	14
--------------------------	----

2.8.1 <i>Sigmoid</i> .....	15
----------------------------	----

2.8.2 <i>Tangen Hiperbolik (tanh)</i> .....	16
---	----

2.8.3 <i>Rectified Linear Unit (ReLU)</i> .....	16
---	----

2.8.4 <i>Softmax</i> .....	17
----------------------------	----

2.9	<i>Hyperparameter Tuning</i> .....	18
2.10	<i>Deep Learning</i> .....	20
2.11	<i>Convolutional Neural Network (CNN)</i> .....	21
2.12	<i>Long Short-Term Memory</i> .....	24
2.13	<i>Bidirectional Long Short-Term Memory</i> .....	28
2.14	<i>Hybrid CNN-BiLSTM</i> .....	29
2.15	Evaluasi Model .....	30
2.15.1	<i>Accuracy</i> .....	32
2.15.2	<i>Precision</i> .....	32
2.15.3	<i>Recall</i> .....	32
2.15.4	<i>F1-score</i> .....	33
<b>III. METODOLOGI PENELITIAN.....</b>		<b>34</b>
3.1	Waktu dan Tempat Penelitian .....	34
3.2	Data dan Alat Penelitian .....	35
3.3	Alur Kerja Penelitian .....	38
<b>IV. HASIL DAN PEMBAHASAN.....</b>		<b>42</b>
4.1	<i>Input Data</i> .....	42
4.2	<i>Exploratory Data Analysis</i> .....	42
4.3	<i>Preprocessing Data</i> .....	48
4.3.1	Menghapus Data Duplikat.....	48
4.3.2	<i>Case Folding</i> .....	49
4.3.3	<i>Cleaning</i> .....	50
4.3.4	<i>Stopword Removal</i> .....	50
4.3.5	<i>Categorical Encoding</i> .....	51
4.3.6	<i>Resampling Data</i> .....	52
4.3.7	<i>Tokenization</i> .....	54
4.4	<i>Word Embedding</i> .....	55
4.5	Pembagian Data .....	56
4.6	<i>Hyperparameter Tuning</i> .....	58
4.7	Membangun Model <i>Hybrid CNN-BiLSTM</i> .....	59
4.8	Validasi Data .....	62
4.9	Evaluasi Kinerja Model <i>CNN-BiLSTM</i> .....	64

<b>V. PENUTUP .....</b>	<b>42</b>
5.1 Kesimpulan.....	42
5.2 Saran .....	43
 <b>DAFTAR PUSTAKA .....</b>	 <b>44</b>
 <b>LAMPIRAN</b>	

## DAFTAR TABEL

Tabel	halaman
1. Penelitian Terkait Implementasi Model <i>Hybrid</i> CNN-BiLSTM.....	5
2. Data Ulasan Pengguna Aplikasi Halodoc Saat COVID-19 .....	35
3. Data Ulasan Pengguna Aplikasi Halodoc Setelah COVID-19.....	36
4. Perbandingan Proporsi Rating Saat COVID-19 dan Setelah COVID-19 .....	47
5. Hasil Proses <i>Remove Duplicated</i> .....	48
6. Hasil Proses <i>Case Folding</i> .....	49
7. Hasil Proses <i>Cleaning</i> .....	50
8. Hasil Proses <i>Stopword Removal</i> .....	51
9. Hasil Proses <i>Categorical Encoding</i> .....	52
10. Hasil <i>Resampling</i> Data .....	53
11. Hasil Proses <i>Tokenization</i> .....	55
12. Hasil <i>Word Embedding</i> dengan <i>Word2Vec</i> .....	56
13. Skenario Pembagian Data. ....	57
14. <i>Hyperparameter</i> yang Ditentukan.....	58
15. Hasil <i>Hyperparameter Tuning</i> Data Saat COVID-19 .....	59
16. Hasil <i>Hyperparameter Tuning</i> Data Setelah COVID-19 .....	59
17. Hasil <i>Accuracy</i> Model.....	65
18. <i>Classification Report</i> Data Saat COVID-19 .....	66
19. <i>Classification Report</i> Data Setelah COVID-19 .....	66
20. Perbandingan Nilai <i>Accuracy</i> Data Saat COVID-19. ....	69
21. Perbandingan Nilai <i>Accuracy</i> Data Setelah COVID-19. ....	69

## DAFTAR GAMBAR

Gambar	halaman
1. <i>Random Oversampling</i> .....	11
2. <i>Random Undersampling</i> .....	12
3. Representasi <i>Word2Vec</i> .....	13
4. Arsitektur <i>Word2Vec</i> CBOW dan <i>Skip Gram</i> .....	13
5. Fungsi Aktivasi <i>Sigmoid</i> .....	15
6. Fungsi Aktivasi <i>tanh</i> .....	16
7. Fungsi Aktivasi <i>ReLU</i> .....	17
8. Arsitektur Jaringan CNN .....	22
9. Operasi Konvolusi .....	23
10. Setelah Operasi Konvolusi .....	23
11. Tiga Metode Lapisan <i>Pooling</i> .....	24
12. Arsitektur Jaringan LSTM .....	25
13. Arsitektur Jaringan BiLSTM .....	28
14. Arsitektur Jaringan CNN-BiLSTM .....	30
15. <i>Confusion Matrix Dasar</i> .....	31
16. <i>Confusion Matrix Multiclass</i> .....	31
17. <i>Flowchart</i> Metodologi Penelitian Model <i>Hybrid CNN-BiLSTM</i> .....	38
18. <i>Wordcloud</i> Rating 1 Data saat COVID-19 .....	43
19. <i>Wordcloud</i> Rating 1 Data Setelah COVID-19 .....	43
20. <i>Wordcloud</i> Rating 2 Data saat COVID-19 .....	43
21. <i>Wordcloud</i> Rating 2 Data Setelah COVID-19 .....	43
22. <i>Wordcloud</i> Rating 3 Data saat COVID-19 .....	44
23. <i>Wordcloud</i> Rating 3 Data Setelah COVID-19 .....	44
24. <i>Wordcloud</i> Rating 4 Data saat COVID-19 .....	44

25. <i>Wordcloud</i> Rating 4 Data Setelah COVID-19 .....	44
26. <i>Wordcloud</i> Rating 5 Data saat COVID-19.....	45
27. <i>Wordcloud</i> Rating 5 Data Setelah COVID-19 .....	45
28. Perbandingan Proporsi Rating Saat dan Setelah COVID-19. ....	46
29. Hasil <i>Resampling</i> untuk Data Saat COVID-19.....	54
30. Hasil <i>Resampling</i> untuk Data Setelah COVID-19. ....	54
31. Arsitektur CNN-BiLSTM yang Dibangun.....	60
32. <i>Loss</i> dan Validasi <i>Loss</i> Saat COVID-19.....	63
33. <i>Accuracy</i> dan Validasi <i>Accuracy</i> Saat COVID-19.....	63
34. <i>Loss</i> dan Validasi <i>Loss</i> Setelah COVID-19. ....	63
35. <i>Accuracy</i> dan Validasi <i>Accuracy</i> Setelah COVID-19.....	63
36. <i>Confusion Matrix</i> Model Saat COVID-19. ....	64
37. <i>Confusion Matrix</i> Model Setelah COVID-19. ....	65

## I. PENDAHULUAN

### 1.1 Latar Belakang dan Masalah

Perkembangan teknologi mengakibatkan terjadinya kemajuan teknologi sehingga memunculkan berbagai jenis layanan yang memudahkan akses terhadap fasilitas pelayanan termasuk pelayanan kesehatan dan mendorong pengembangan berbagai jenis aplikasi. *Telemedicine* merupakan salah satu inovasi dari perkembangan teknologi yang membantu dalam memudahkan akses pelayanan kesehatan akibat keterbatasan jarak antara pasien dan dokter (Izzati dan Firmanto, 2021).

Berbagai pihak telah meluncurkan berbagai aplikasi yang mendukung akses fasilitas pelayanan kesehatan. Aplikasi penyedia layanan kesehatan pilihan masyarakat salah satunya yaitu aplikasi Halodoc. Halodoc adalah aplikasi yang menyediakan beberapa fasilitas untuk para penggunanya secara *online* seperti pembelian obat, fasilitas konsultasi dengan dokter, dan beberapa fasilitas pendukung kesehatan lainnya (Izzati dan Firmanto, 2021). Selain itu, Halodoc juga menyediakan fitur untuk penggunanya agar dapat memberikan ulasan dan rating berupa bintang mengenai pelayanan yang diberikan. Ulasan dan rating tersebut dapat menjadi informasi bagi pengguna lainnya mengenai kualitas pelayanan dari aplikasi Halodoc. Semakin tinggi rating diberikan yang berarti semakin baik kualitas pelayanan yang diberikan. Sehingga, rating menjadi salah satu pertimbangan bagi para pengguna seberapa baik layanan aplikasi Halodoc mampu memenuhi kebutuhan penggunanya. Oleh karena itu, diperlukan cara yang tepat untuk menganalisis kepuasan pengguna terhadap pelayanan aplikasi Halodoc. Salah satu cara yang dapat digunakan melalui klasifikasi rating dari ulasan yang

diberikan oleh pengguna dan membandingkan proporsi rating untuk melihat bagaimana peranan aplikasi halodoc pada saat dan setelah adanya COVID-19.

Klasifikasi dapat dilakukan dengan bantuan *Machine Learning* (ML), tetapi memiliki keterbatasan utama yaitu tidak dapat mengekstraksi kumpulan fitur tertentu pada teks dan tidak dapat mengekstraksi fitur pembeda dari kumpulan data. Kelemahan ini dapat diperbaiki dengan menggunakan algoritma *Deep Learning* (DL) yang mampu melakukan ekstraksi fitur secara otomatis (Alzubaidi dkk., 2021).

Metode DL yang dapat digunakan untuk melakukan klasifikasi data teks adalah *Convolutional Neural Network* (CNN), *Long Short-Term Memory* (LSTM), dan *Bidirectional Long Short-Term Memory* (BiLSTM). Metode CNN adalah jenis khusus jaringan saraf *multilayer* yang terinspirasi oleh sistem visual makhluk hidup (Ghosh dkk., 2019). Pada konteks data teks, CNN mampu mengekstraksi fitur-fitur penting dari teks (Krishna dkk., 2018). Kemudian, LSTM merupakan jenis struktur *Recurrent Neural Networks* (RNN) khusus yang mengatasi kelemahan RNN dalam menyimpan memori jangka panjang (Fan dkk., 2020). Metode LSTM dapat mempelajari dependensi jangka panjang menggunakan mekanisme yang disebut *gates*. *Gates* ini dapat mempelajari informasi apa saja yang penting untuk disimpan atau dibuang (Mousavi dkk., 2021). Namun, jaringan LSTM tunggal yang hanya dapat menangkap informasi keadaan sebelumnya tetapi tidak dapat menangkap informasi keadaan masa depan. Hal tersebut dapat di atasi dengan menggunakan model BiLSTM. Model BiLSTM adalah tipe jaringan saraf yang mampu memahami konteks jangka panjang dalam urutan data dan dapat mengakses konteks dari masa lalu dan masa depan pada suatu kata atau entitas di dalam kalimat (Shan dkk., 2021). Sehingga, implementasi penggabungan model CNN dan BiLSTM sebagai model *hybrid* akan meningkatkan tingkat akurasi dalam melakukan klasifikasi rating ulasan aplikasi Halodoc.

Berbagai penelitian mengenai model *hybrid* CNN dan BiLSTM juga telah dilakukan dan memperoleh hasil yang baik, salah satunya adalah penelitian yang dilakukan oleh Rhanoui dkk., (2019), yaitu penelitian mengenai CNN-BiLSTM untuk analisis sentimen pada artikel berbahasa Prancis yang mendapatkan hasil akhir dengan tingkat akurasi sebesar 90% yang menunjukkan bahwa hasil analisis sentimennya baik. Contoh penelitian lain yaitu dilakukan oleh Xiaoyan dkk., (2022), yaitu melakukan penelitian tentang model CNN-BiLSTM untuk analisis sentimen pada teks ulasan mengenai COVID-19 yang memperoleh hasil akhir akurasi sebesar 95%. Berdasarkan tingkat akurasi yang didapatkan menunjukkan bahwa hasil analisis sentiment menggunakan model CNN-BiLSTM sangat baik.

Berdasarkan uraian di atas, peneliti akan mencoba mengimplementasikan model *hybrid* yang menggabungkan model CNN dan BiLSTM untuk klasifikasi rating ulasan pengguna aplikasi layanan kesehatan Halodoc. Kemudian, dilakukan prediksi berdasarkan hasil klasifikasi dengan menentukan parameter terbaik untuk mendapatkan tingkat akurasi yang baik.

## **1.2 Rumusan Masalah**

Rumusan masalah dalam penelitian ini diantaranya:

1. Bagaimana peranan dari aplikasi Halodoc pada saat COVID-19 berlangsung dan setelah COVID-19 selesai?.
2. Bagaimana menentukan parameter terbaik yang akan digunakan untuk membangun model CNN-BiLSTM agar mendapatkan hasil dengan tingkat akurasi yang baik?.
3. Bagaimana menggabungkan dua struktur model *deep learning* sekaligus, yaitu CNN dan BiLSTM untuk analisis tingkat kepuasan pengguna aplikasi Halodoc?.

### 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menganalisis jumlah proporsi rating untuk melihat bagaimana peranan aplikasi halodoc pada saat COVID-19 berlangsung dan setelah COVID-19 selesai.
2. Mengembangkan model *hybrid* yang menggabungkan model CNN dan BiLSTM untuk klasifikasi rating ulasan aplikasi Halodoc.
3. Menganalisis hasil klasifikasi rating ulasan aplikasi Halodoc menggunakan model *metric* evaluasi *accuracy*, *precision*, *recall*, dan *f1-score*.

### 1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Sebagai wawasan peneliti selanjutnya dalam mengaplikasikan model *hybrid* CNN-BiLSTM.
2. Sebagai bahan pertimbangan perusahaan, instansi kesehatan, masyarakat maupun pihak-pihak lain yang memiliki kepentingan terhadap aplikasi Halodoc.
3. Sebagai bahan rujukan bagi pengembang aplikasi Halodoc agar dapat meningkatkan kualitas pelayanan dan pengambilan keputusan.

## II. TINJAUAN PUSTAKA

### 2.1 Penelitian Terkait

Penelitian terkait dalam penelitian ini digunakan sebagai bahan acuan atau rujukan untuk hasil analisis. Topik penelitian yang menjadi pembanding ialah analisis *sentiment*, model *hybrid CNN-BiLSTM*, dan *metric* evaluasi yang digunakan mulai dari *accuracy (acc)*, *precision (prec)*, *recall (rec)*, dan *f1-score (f1)*. Secara umum mengenai beberapa penelitian yang digunakan dalam penelitian ini dirangkum dalam Tabel 1.

Tabel 1. Penelitian Terkait Implementasi Model *Hybrid CNN-BiLSTM*

No	Penelitian	Data	Metode	Hasil (%)			
				<i>acc</i>	<i>prec</i>	<i>rec</i>	<i>f1</i>
1	<i>GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews</i> (Xiaoyan dkk., 2022)	<b>COVID-19</b> Jumlah: 81696 Positif : 35093 Negatif: 31060 Netral : 15543  Sumber: Twitter	<i>Embedding:</i> <i>Glove</i> Model: CNN-BiLSTM	95,6	-	-	95,5

2.	<i>Social Media Sentiment Analysis Using CNN-BiLSTM Mechanism</i> (Bharal dan Krishna, 2021)	<b>Social Media</b> Jumlah: 1600000 Positif: 800000 Negatif: 800000  Sumber: Twitter	<i>Embedding: Word2Vec</i> Model: CNN-BiLSTM	79	79	79	79
3.	<i>A CNN-BiLSTM Model for Document-Level Sentiment Analysis</i> (Rhanoui dkk., 2019)	<b>French articles</b> Jumlah: 2003 Positif : 474 Negatif: 1247 Netral : 282  Sumber: Surat kabar nasional dan internasional	<i>Embedding: Doc2Vec</i> Model: CNN-BiLSTM	90,6	-	-	-

Berikut adalah resume dari penelitian pada Tabel 1.

a. Penelitian Pertama (Xiaoyan dkk., 2022)

Penelitian ini dilakukan oleh Xiaoyan dkk., (2022) yang menerapkan model CNN-BiLSTM untuk analisis sentimen pada teks ulasan mengenai COVID-19. Data yang digunakan dalam penelitian ini adalah data komentar tentang COVID-19 dan diperoleh dari Twitter melalui metode *web crawler*. Data tersebut memiliki total 81696 baris yang meliputi 35093 komentar positif, 31060 komentar negatif, dan 15543 komentar netral.

Pada penelitian ini dilakukan *text preprocessing* yang meliputi *stopword removal*, *lemmatization*, dan *tokenization*. Selanjutnya, menggunakan *word embedidng GloVe* dengan representasi kata yang memuat informasi semantik dan tata bahasa teks sebanyak mungkin. Kemudian, data dibagi menjadi data pelatihan dan data

*test*, dengan rasio 80:20. Selanjutnya, dibangun model CNN-BiLSTM untuk pelatihan. Hasil penelitian dengan menggunakan model CNN-BiLSTM diperoleh nilai *accuracy* sebesar 95,6% dan *f1-score* sebesar 95,5%.

b. Penelitian Kedua (Bharal dan Krishna, 2021)

Penelitian yang dilakukan oleh Bharal dan Krishna (2021) adalah penelitian mengenai analisis sentimen menggunakan model CNN-BiLSTM. Penelitian ini menggunakan data tentang berbagai *platform* media sosial yang diperoleh dari *Real-time Twitter API*. Data yang digunakan berjumlah 1.600.000 *tweet* yang mencakup 800.000 *tweet* positif dan negatif.

Tahapan pertama yang dilakukan dalam penelitian ini adalah menghapus tanda baca dan emoji. Selanjutnya, digunakan *word embedding*, yaitu *Word2Vec* sebagai representasi kata. Kemudian, data dalam penelitian ini dibagi menjadi 80% untuk data pelatihan dan 20% untuk data pengujian. Selanjutnya, dibangun model CNN-BiLSTM dengan lapisan *embedding* yaitu *Word2Vec* untuk melakukan pelatihan. Hasil pengujian model diperoleh nilai *accuracy* lebih rendah dari penelitian sebelumnya yaitu sebesar 79%.

c. Penelitian Ketiga (Rhanoui dkk., 2019)

Penelitian lainnya dilakukan oleh Rhanoui dkk., (2019), yaitu penelitian tentang analisis sentimen pada artikel berbahasa Prancis menggunakan model CNN-BiLSTM. Data yang digunakan dalam penelitian ini diperoleh dari Surat kabar nasional dan internasional dengan jumlah 2003 artikel yang terdiri dari 1247 artikel netral, 474 artikel positif, dan 282 artikel negatif.

Pada penelitian ini dilakukan *cleaning* data. Kemudian, melakukan representasi kata dengan menggunakan *Doc2Vec*. Data dalam penelitian ini dibagi menjadi tiga bagian yaitu sebanyak 1502 data untuk data pelatihan, 375 data untuk data validasi dan 126 data untuk data pengujian. Selanjutnya, dibangun model CNN-BiLSTM dan diperoleh hasil akhir nilai *accuracy* sebesar 90,6%.

## 2.2 *Sentiment Analysis*

Menurut Xiaoyan dkk., (2022), *sentiment analysis* atau penggalian opini adalah tugas penting dalam pemrosesan bahasa alami (*natural language processing*). *Sentiment analysis* teks adalah proses mengekstraksi opini, emosi, sikap, dan perasaan seseorang tentang suatu topik atau situasi dari sejumlah besar data yang tidak terstruktur. Perkembangan internet yang semakin pesat membuat masyarakat semakin sering mengutarakan pendapatnya mengenai suatu objek atau peristiwa pada platform sosial seperti barang yang dibeli di toko *online*, film yang baru dirilis, perkembangan politik, dan beberapa peristiwa hangat lainnya. Komentar masyarakat mengenai hal ini seringkali mengandung sentimen dan kecenderungan sentimental. Jika komentar-komentar tersebut dikumpulkan dan dianalisis bagaimana kecenderungan sentimental pengguna, maka dapat dipahami informasi dari mulut ke mulut pengguna dan memberikan kontribusi kuat untuk pengambilan keputusan selanjutnya.

## 2.3 *Natural Language Processing*

*Natural Language Processing* (NLP) adalah salah satu bidang ilmu dari *Artificial Intelligence* yang berhubungan dengan interaksi antara komputer dan bahasa alami manusia (Jatnika dkk., 2019). Mesin mempelajari sintaks dan makna bahasa manusia, memprosesnya, dan memberikan *output* kepada pengguna. Bidang NLP melibatkan sistem komputer untuk melakukan tugas-tugas dengan bahasa alami dan dapat dimengerti manusia. Tahapan dalam NLP meliputi *text preprocessing*. Tahap *text preprocessing* tergantung pada kebutuhan dan karakteristik dari tugas pemrosesan bahasa alami.

Menurut Surjandari dkk., (2016), tahap *text preprocessing* dilakukan untuk mengubah data teks yang tidak terstruktur menjadi terstruktur. Secara umum tahapan dalam *text preprocessing* terdiri dari beberapa langkah sebagai berikut:

1. *Case folding*: proses mengubah semua huruf besar menjadi huruf kecil dalam suatu dokumen dan sebaliknya.
2. *Cleaning*: proses menghapus karakter selain alfabet yang terdiri *punctuation* (tanda baca), emoji, dan spasi berlebih.
3. *Stopword removal*: proses menghapus kata-kata yang sangat umum dan nilainya kecil atau tidak memiliki makna berarti. *Stopword* dalam Bahasa Indonesia pada *python* dapat menggunakan library *Sastrawi*. Beberapa contoh kata *stopword* diantaranya “dan”, “aku”, “kamu”, dan lain sebagainya.
4. *Tokenization*: proses memisahkan suatu kalimat menjadi beberapa bagian kata yang disebut token.
5. *Stemming*: merupakan tahapan dalam *text preprocessing* yang bertujuan untuk menemukan kata dasar dari kata asli yang muncul pada sebuah teks (Rifai dan Winarko, 2019). Meskipun pengembangan algoritma untuk *stemming* selalu diperbarui, namun masih terdapat permasalahan yang sering terjadi termasuk *overstemming* dan *understemming*. *Overstemming* yaitu pemotongan kata secara berlebihan setelah dilakukan *stemming*. Sedangkan, *understemming* yaitu pemotongan kata tidak cukup setelah dilakukan *stemming* (Rifai dan Winarko, 2019).

## 2.4 Text Mining

Menurut Salloum dkk., (2018), *text mining* adalah proses memperoleh informasi yang bermakna, seperti pola dan ringkasan dari teks dengan sumber data dokumen yang tidak terstruktur. *Text mining* memiliki kemampuan mengekstraksi indeks numerik yang bermakna dari teks dan kemudian informasi yang terkandung dalam teks tersebut dapat diakses untuk dilakukan analisis (Surjandari dkk., 2016). Mengekstraksi informasi secara manual pada teks dari dokumen yang sangat besar akan memerlukan waktu yang cukup lama dan kurang efektif. Sehingga, dibutuhkan bantuan *software* komputer untuk menerapkan tugas dari *text mining*. Secara khusus, tugas *text mining* diantaranya, yaitu mengklasifikasikan dokumen berdasarkan kata-kata yang terkandung di dalamnya dan menentukan kemiripan

antar dokumen. Aplikasi *text mining* yang paling umum adalah memfilter dokumen spam, menganalisis sentimen, merangkum dokumen, dan lain sebagainya.

## 2.5 Klasifikasi Teks

Klasifikasi merupakan teknik *supervised learning* yang bertujuan untuk menetapkan label berdasarkan dokumen yang telah diberikan label sebelumnya. Secara matematis, klasifikasi adalah proses memetakan suatu fungsi ( $f$ ) dari variabel *input* ( $X$ ) ke variabel *output* ( $Y$ ) (Sarker, 2021). Klasifikasi dibagi menjadi beberapa permasalahan berdasarkan jumlah labelnya, yaitu klasifikasi biner dan *multiclass*. Klasifikasi biner mengacu pada tugas klasifikasi yang memiliki dua label kelas seperti ya dan tidak. Sedangkan, klasifikasi *multiclass* mengacu pada tugas klasifikasi yang memiliki lebih dari dua label kelas (Sarker, 2021). Klasifikasi juga dapat dibagi menjadi beberapa permasalahan berdasarkan jenis variabelnya, yaitu diantaranya klasifikasi nominal dan ordinal. Klasifikasi nominal adalah klasifikasi dengan kelas-kelas atau kategori yang tidak memiliki urutan atau tingkatan tertentu, setiap kelas dianggap setara, dan saling independen. Contoh sederhana dari klasifikasi nominal, yaitu jenis-jenis buah (semangka, melon, dan jeruk). Sedangkan, klasifikasi ordinal adalah klasifikasi dengan kelas-kelas atau kategori yang memiliki urutan atau tingkatan tertentu. Adanya urutan dan tingkatan tertentu memungkinkan data memiliki jumlah kelas yang tidak seimbang. Sehingga, dapat memengaruhi kinerja model dalam mengenali kelas yang memiliki jumlah paling sedikit (Bellmann dkk., 2022). Contoh klasifikasi ordinal, yaitu rating (1, 2, 3, 4, dan 5) yang memiliki tingkatan berupa *sentiment* atau tingkat kepuasan pelanggan.

## 2.6 Resampling Data

Menurut Mohammed dkk., (2020), *resampling* data adalah suatu teknik yang digunakan untuk mengatasi permasalahan data yang tidak seimbang (*imbalance*). *Imbalance* data dalam klasifikasi teks terjadi karena beberapa kelas data memiliki jumlah kelas yang tidak sama. Permasalahan *imbalance* data dapat di atasi menggunakan *resampling* data. Menurut Mohammed dkk., (2020), teknik *resampling* yang biasa digunakan yaitu *oversampling* dan *undersampling*. *Oversampling* digunakan untuk menambahkan jumlah data pada kelas minoritas hingga jumlahnya mencapai jumlah data yang sama dengan jumlah data kelas mayoritas. Sedangkan, *undersampling* digunakan untuk mengeliminasi data pada kelas mayoritas hingga jumlah data pada kelas mayoritas dan kelas minoritas seimbang.

### 2.6.1 Random Oversampling

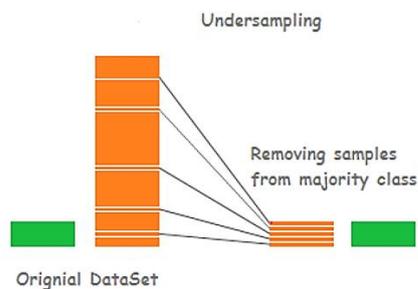
*Random oversampling* merupakan salah satu cara untuk mengatasi *imbalance* data, yaitu menyeimbangkan jumlah data pada kelas minoritas dan kelas mayoritas. Pada proses *random oversampling* akan dilakukan replikasi secara acak pada data minoritas agar jumlah data pada kelas minoritas memiliki jumlah yang sama dengan kelas mayoritas (Wongvorachan dkk., 2023). Berikut ini adalah contoh penggunaan *random oversampling* yang disajikan pada Gambar 1.



Gambar 1. *Random Oversampling* (Mohammed dkk., 2020).

### 2.6.2 *Random Undersampling*

Berbeda dengan *random oversampling* yang secara acak melakukan replikasi data pada kelas minoritas, *random undersampling* melakukan eliminasi secara acak pada data di kelas mayoritas agar jumlah data pada kelas mayoritas memiliki jumlah yang sama dengan kelas minoritas (Wongvorachan dkk., 2023). Berikut ini merupakan contoh penggunaan *random undersampling* yang disajikan pada Gambar 2.

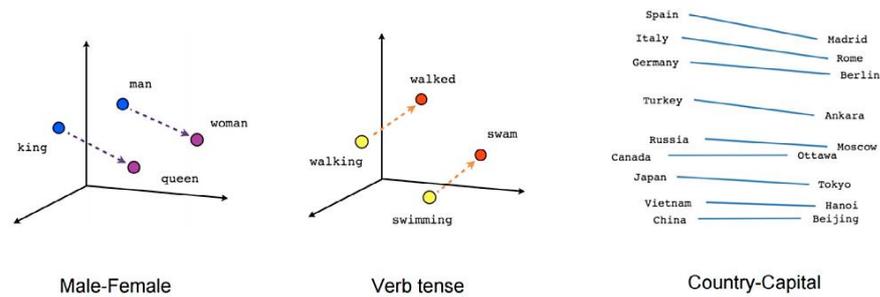


Gambar 2. *Random Undersampling* (Mohammed dkk., 2020).

## 2.7 *Word Embedding*

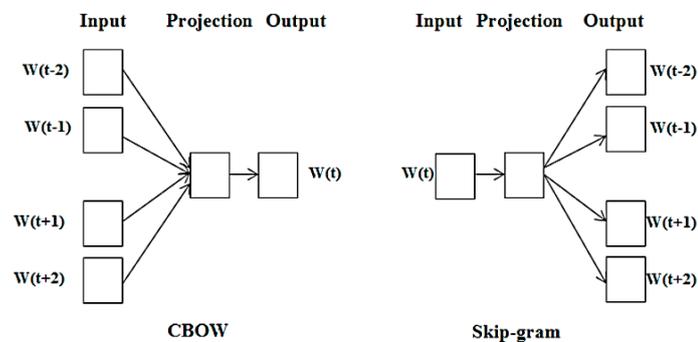
Menurut Jatnika dkk., (2019), *word embedding* adalah teknik pembelajaran dalam *Natural Language Processing* (NLP) yang digunakan untuk merepresentasikan kata atau frasa dalam bentuk vektor bilangan real. Metode *word embedding* yang populer dan sering digunakan salah satunya yaitu *word2vec*.

*Word2Vec* merepresentasikan kata menjadi vektor berdasarkan beberapa fitur yang dimilikinya seperti *windows size* dan *vector dimensions* (Jatnika dkk., 2019). Kata-kata yang mirip cenderung memiliki nilai vektor yang sama dan dikelompokkan dalam satu blok yang sama dapat dilihat pada Gambar 3.



Gambar 3. Representasi *Word2Vec* (Jatnika dkk., 2019).

Oleh karena itu, *Word2Vec* dapat menangkap nilai kesamaan antar kata dari pelatihan korpus yang besar. Nilai kemiripan yang dihasilkan *Word2Vec* berkisar antara -1 hingga 1 sebagai nilai kemiripan tertinggi. *Word2Vec* dapat memberikan implementasi arsitektur *Continuous Bag of Words* (CBOW) dan *Skip Gram* yang efisien untuk menghitung representasi vektor kata, representasi ini dapat digunakan untuk berbagai tugas dalam *text preprocessing* (Jatnika dkk., 2019). Arsitektur CBOW dan *Skip Gram* diilustrasikan pada Gambar 4.



Gambar 4. Arsitektur *Word2Vec* CBOW dan *Skip Gram* (Jatnika dkk., 2019).

a. *Continuous Bag of Words* (CBOW)

Arsitektur CBOW memprediksi kata target ( $t$ ) berdasarkan konteks ( $c$ ). Vektor *input* yang digunakan CBOW adalah rata-rata dari vektor konteks kata-kata yang ditunjukkan oleh Persamaan (2.1) sebagai berikut.

$$c = \frac{1}{2W} \sum_{i=1}^{2W} w_i \quad (2.1)$$

Keterangan:

$w_i$  = Vektor representasi dari kata konteks ke-( i )

Selanjutnya, peluang bersyarat untuk kata target (t) berdasarkan kata konteks (c) dapat dihitung dengan menggunakan Persamaan (2.2).

$$P(t|c) = \frac{e^{v_t v_c}}{\sum_{j=1}^V e^{v_j v_c}} \quad (2.2)$$

Keterangan:

$v_t$  = Vektor kata target yang ingin diprediksi

$V$  = Jumlah total kata dalam korpus

#### b. *Skip Gram*

Arsitektur *skip-gram* memprediksi kata konteks berdasarkan kata target. Peluang bersyarat untuk kata konteks (c) berdasarkan kata target (t) dapat dihitung dengan menggunakan Persamaan (2.3).

$$P(c|t) = \frac{e^{v_c v_t}}{\sum_{j=1}^V e^{v_j v_t}} \quad (2.3)$$

Keterangan:

$v_c$  = Vektor kata konteks yang ingin diprediksi

$v_t$  = Vektor kata target yang diberikan

## 2.8 Fungsi Aktivasi

Menurut S. Sharma dkk., (2020), fungsi aktivasi adalah fungsi untuk mengubah nilai *input* menjadi nilai *output*. Nilai yang diperoleh dari hasil aktivasi digunakan sebagai *input* ke lapisan berikutnya. Tingkat akurasi prediksi model yang digunakan bergantung pada jumlah lapisan yang digunakan dan jenis fungsi

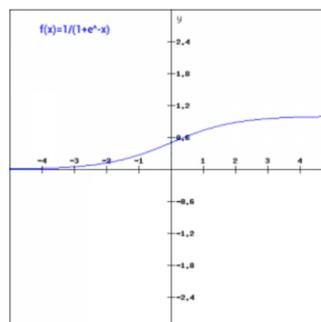
aktivasi yang digunakan. Fungsi aktivasi yang umum digunakan antara lain *sigmoid*, *Tangen Hiperbolik* (TanH), *Rectified Linear Unit* (ReLU), dan *softmax*.

### 2.8.1 Sigmoid

Menurut Nwankpa dkk., (2018), fungsi *sigmoid* adalah fungsi *non-linear* terdiferensiasi terbatas yang didefinisikan untuk semua nilai *input* nyata dan memiliki turunan positif. Fungsi *sigmoid* didefinisikan oleh Persamaan (2.5) sebagai berikut.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

Fungsi aktivasi *sigmoid* memiliki kelemahan utama, yaitu mengikat *input* pada rentang antara 0 dan 1. Berikut bentuk dari fungsi *sigmoid* yang disajikan pada Gambar 5.



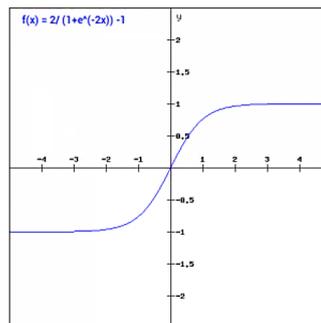
Gambar 5. Fungsi Aktivasi *Sigmoid* (S. Sharma dkk., 2020).

### 2.8.2 Tangen Hiperbolik (tanh)

*Tangen Hiperbolik* (tanh) adalah fungsi aktivasi yang memiliki keunggulan utamanya adalah menghasilkan keluaran terpusat pada nilai nol dengan rentang antara -1 hingga 1 (Nwankpa dkk., 2018). Bentuk fungsi aktivasi *tanh* ditunjukkan oleh Persamaan (2.6).

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.6)$$

Berdasarkan persamaan (2.6), grafik fungsi aktivasi *tanh* dapat dilihat pada Gambar 6 sebagai berikut.



Gambar 6. Fungsi Aktivasi tanh (S. Sharma dkk., 2020).

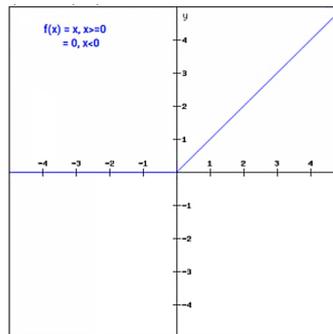
### 2.8.3 Rectified Linear Unit (ReLU)

Menurut Nwankpa dkk., (2018), ReLU adalah fungsi aktivasi yang sering digunakan karena memiliki komputasi yang lebih cepat, kinerja dan generalisasi yang lebih baik dibandingkan dengan fungsi aktivasi *sigmoid* dan *tanh*. Fungsi aktivasi ReLU memperbaiki nilai *input* yang kurang dari nol sehingga memaksa nilai *output* menjadi nol dan setiap nilai *input* yang lebih dari nol akan

dipertahankan nilainya. Bentuk fungsi aktivasi ReLU ditunjukkan oleh Persamaan (2.7) sebagai berikut.

$$f(x) = \max(0, x) \quad (2.7)$$

Persamaan (2.7) menunjukkan bahwa jika  $x > 0$ , maka  $f(x) = x$ , dan jika  $x \leq 0$ . Grafik fungsi aktivasi ReLU dapat dilihat pada Gambar 7.



Gambar 7. Fungsi Aktivasi ReLU (S. Sharma dkk., 2020).

Berdasarkan Gambar 7, dapat interpretasikan bahwa meskipun fungsi aktivasi ReLU *linear* pada  $x > 0$ , namun terdapat nilai nol pada  $x \leq 0$ . Sehingga, dapat dikatakan bahwa fungsi aktivasi ReLU merupakan fungsi *non-linear*.

#### 2.8.4 Softmax

*Softmax* merupakan fungsi aktivasi yang digunakan pada *output* dalam klasifikasi *multiclass*. *Softmax* menghasilkan *output* dengan rentang nilai antara 0 sampai 1 dan jumlah peluangnya sama dengan 1 (Nwankpa dkk., 2018). Bentuk fungsi aktivasi *softmax* ditunjukkan oleh Persamaan (2.8).

$$S_k = \frac{e^{y_k}}{\sum_{k=1}^m e^{y_k}} \quad (2.8)$$

Keterangan:

$S_k$  = Nilai peluang prediksi lapisan *output*

$y_k$  = Sinyal *output* pada lapisan *output*

$e$  = eksponensial (2,7183)

$m$  = Jumlah kelas pada lapisan *output*

$\sum_{k=1}^m e^{y_k}$  = Jumlah eksponensial sinyal *output* pada lapisan *output*

## 2.9 Hyperparameter Tuning

Proses pembuatan model perlu dilakukan *hyperparameter tuning* (inisialisasi parameter) untuk mengetahui parameter terbaik yang digunakan untuk membangun model (Elgeldawi dkk., 2021). Salah satu *hyperparameter* utama yang perlu disesuaikan adalah *learning rate* dan *batch size*. *Learning rate* adalah parameter yang digunakan untuk mengatur kecepatan dalam memperbaharui parameter model (Han dkk., 2023). *Learning rate* biasanya memiliki nilai antara 0 dan 1. Jika nilai *learning rate* terlalu kecil, pembaruan parameter model akan sangat kecil, yang berarti bahwa proses penyesuaian model akan berlangsung sangat lambat. Pada kasus ini, algoritma membutuhkan banyak waktu untuk mencapai konvergensi atau titik di mana model telah belajar dengan baik dari data. Sedangkan, jika nilai *learning rate* terlalu besar, setiap pembaruan parameter model akan signifikan yang dapat menyebabkan algoritma "melompati" wilayah pencarian yang diinginkan. Hal ini mengakibatkan algoritma melewati titik minimum atau wilayah yang berarti bahwa kesalahan model berkurang secara signifikan. Menyebabkan hasil yang diperoleh tidak konvergen pada solusi yang optimal (Han dkk., 2023). Aturan praktis dalam menentukan nilai *learning rate* dapat diatur ke nilai  $1/t$ , di mana  $t$  adalah jumlah *epoch* yang digunakan pada proses pelatihan model (Han dkk., 2023). Kemudian, *batch size* merupakan jumlah sampel yang digunakan dalam setiap *epoch* pada proses pelatihan model (Kandel dan Castelli, 2020). Nilai *batch size* yang kecil membutuhkan waktu komputasi yang lebih cepat dan membutuhkan memory yang lebih sedikit dibandingkan dengan nilai *batch size* yang besar.

Namun, nilai *batch size* yang besar memungkinkan mencapai titik minimum optimal yang lebih baik yang mungkin tidak dapat dicapai oleh nilai *batch size* yang kecil. Nilai *batch size* yang biasanya digunakan diantaranya, yaitu 32 dan 64 (Kandel dan Castelli, 2020). Menurut Elgeldawi dkk., (2021), algoritma *hyperparameter tuning* yang umum digunakan yaitu *grid search*, *random search*, dan *bayesian search*.

a. *Grid Search*

*Grid Search* melatih algoritma model dengan setiap kemungkinan kombinasi dari semua nilai *hyperparameter* yang disediakan. Selanjutnya, menghitung skor setiap model untuk mengevaluasinya, dan kemudian memilih model yang memberikan hasil terbaik.

b. *Random Search*

*Random Search* mengambil sampel ruang pencarian dan mengevaluasi kumpulan dari sebaran peluang yang ditentukan. Singkatnya, teknik ini merupakan kombinasi acak dari *hyperparameter* yang digunakan untuk menemukan solusi terbaik pada model yang sedang dipertimbangkan (Elgeldawi dkk., 2021).

c. *Bayesian Search*

*Bayesian search* adalah algoritma pencarian informasi dengan setiap iterasi dari algoritma ini belajar dari iterasi sebelumnya, dan hasil dari satu iterasi membantu dalam membuat iterasi berikutnya. Algoritma *bayesian search* mirip dengan algoritma *random search* dalam arti bahwa metode ini mengambil sampel subset dari kombinasi *hyperparameter*. Namun, cara pemilihan setiap kombinasinya berbeda. *Bayesian search* memandang proses penyetelan *hyperparameter* sebagai optimalisasi fungsi *black-box*, yaitu fungsi yang hanya memberikan *output* berdasarkan *input*. Fungsi yang akan dioptimalkan adalah skor prediksi akhir model (akurasi) pada set pengujian (Elgeldawi dkk., 2021).

## 2.10 Deep Learning

Menurut Han dkk., (2023), *deep learning* adalah pembelajaran yang didasarkan pada jaringan saraf tiruan. Jaringan saraf adalah sekumpulan unit *input-output* yang terhubung, dengan setiap koneksi memiliki bobot yang terkait dengannya. Selama tahap pembelajaran, jaringan belajar dengan menyesuaikan bobot untuk memprediksi nilai target yang benar.

Menurut Krishna dkk., (2018), *deep learning* adalah subbidang *machine learning*, yang mampu belajar melalui metode komputasinya sendiri. Model *deep learning* mampu mencari informasi dengan struktur yang homogen seperti bagaimana manusia akan membuat keputusan. Hal ini membuat *deep learning* paling mumpuni dibandingkan model *machine learning* standar.

Menurut Alzubaidi dkk., (2021), *deep learning* diklasifikasikan menjadi beberapa kategori, yaitu:

1. *Deep supervised learning*: teknik ini berkaitan dengan data yang memiliki label. Keuntungan utama dari teknik ini adalah kemampuan untuk mengumpulkan data atau menghasilkan *output* data dari pengetahuan sebelumnya. Secara keseluruhan, teknik ini lebih sederhana dari teknik lain dalam cara belajar dengan kinerja tinggi.
2. *Deep semi-supervised learning*: merupakan pembelajaran mesin yang digunakan pada data yang sebelumnya tidak memiliki label. Tujuan dari teknik pembelajaran ini ialah untuk mengelompokkan data yang hampir sama dalam satu area tertentu dan tidak memerlukan target *output*. Teknik ini mampu menemukan struktur atau pola tersembunyi pada data yang tidak memiliki label, biasanya digunakan untuk menemukan klasifikasi pola.
3. *Deep unsupervised learning*: teknik ini memungkinkan untuk mengimplementasikan proses pembelajaran dengan data yang tidak memiliki label. Beberapa teknik *deep unsupervised learning* yang sering dilakukan yaitu *dimensionality reduction* dan *clustering*. Kelemahan utama *unsupervised*

*learning* adalah tidak dapat memberikan informasi yang akurat mengenai pemilahan data dan komputasi yang kompleks.

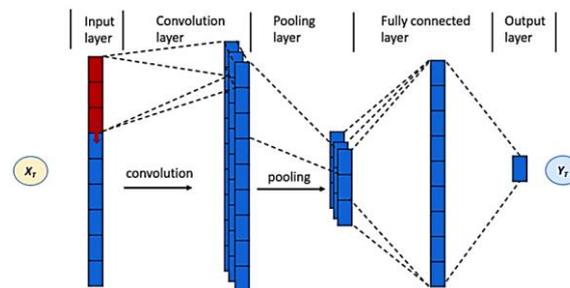
4. *Deep reinforcement learning*: beroperasi pada interaksi dengan lingkungan, sedangkan *supervised learning* beroperasi pada data sampel yang disediakan. Dibandingkan dengan teknik *supervised learning*, melakukan pembelajaran ini jauh lebih sulit, karena tidak ada fungsi kerugian langsung tersedia dalam teknik pembelajaran *reinforcement learning*.

### 2.11 Convolutional Neural Network (CNN)

Menurut Gao dkk., (2022), CNN adalah sejenis jaringan saraf *deep feed-forward* yang khusus memproses data dengan struktur jaringan. Model CNN memiliki kemampuan ekstraksi fitur dan pembagian bobot. Model CNN meliputi CNN 3D, CNN 2D, dan CNN 1D, perbedaannya adalah sebagai berikut:

1. Model CNN 3D dapat secara bersamaan mengekstraksi fitur temporal dan spasial untuk pengenalan perilaku dan pemrosesan video.
2. Model CNN 2D umumnya digunakan dalam bidang computer vision dan pemrosesan gambar.
3. Model CNN 1D umumnya digunakan di bidang model urutan dan *Natural Language Processing* (NLP), seperti analisis opini, review produk, ringkasan teks, dan lain sebagainya.

Model CNN terdiri dari tiga lapisan utama yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Lapisan tersembunyi dari model CNN biasanya terdiri dari tiga lapisan, yaitu lapisan konvolusi, lapisan *pooling* dan lapisan *fully connected*. Lapisan konvolusi menggunakan filter sebagai ekstraktor fitur untuk mengekstraksi fitur tingkat tinggi (Yu dkk., 2021). Berikut merupakan lapisan-lapisan dalam arsitektur CNN beserta fungsinya yang diilustrasikan pada Gambar 8.



Gambar 8. Arsitektur Jaringan CNN (Shan dkk., 2021).

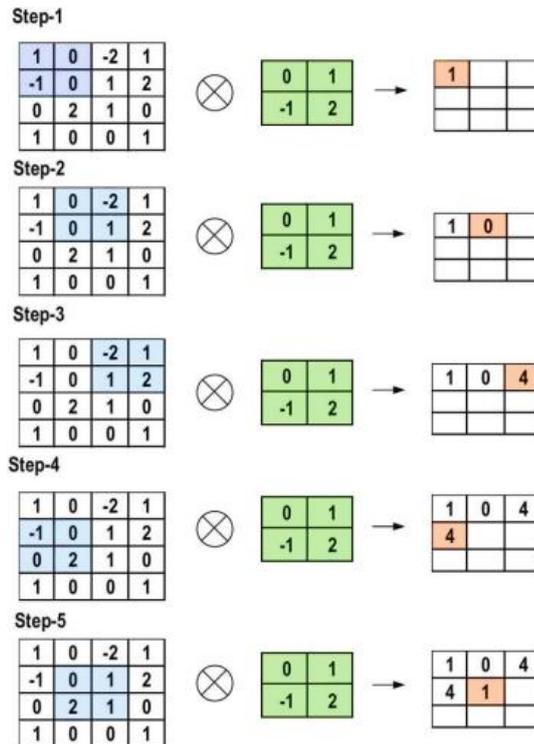
a. Lapisan *input*

Pada lapisan *input*, semua urutan dikonversi ke bentuk bilangan bulat berdasarkan token yang telah diberi indeks unik. Urutan *input* kemudian diberi nilai nol supaya mempunyai panjang yang sama untuk membantu meningkatkan kinerja dengan menjaga informasi tetap terjaga (Kapil dkk., 2020).

b. Lapisan konvolusi

Lapisan konvolusi adalah komponen terpenting dari setiap arsitektur CNN. Lapisan konvolusi terdiri dari kumpulan filter konvolusi yang disebut kernel. Kernel dapat digambarkan sebagai kisi nilai dengan setiap nilai dikenal sebagai bobot kernel. Selama awal proses pelatihan model CNN, semua bobot kernel ditetapkan dengan angka acak. Kemudian, setiap periode pelatihan, bobot diatur dan kernel belajar untuk mengekstraksi fitur yang bermakna (Ghosh dkk., 2019).

Pertama, kernel menggeser *input* secara horizontal dan vertikal. Selain itu, perkalian titik antara *input* dan kernel ditentukan nilainya dengan cara dikalikan dan kemudian dijumlahkan untuk membuat nilai skalar tunggal pada *output feature map* yang dihitung secara bersamaan. Seluruh proses ini kemudian diulang sampai tidak terjadi kemungkinan lain yang muncul (Ghosh dkk., 2019). Perhitungan yang dilakukan pada setiap langkah secara grafis disajikan pada Gambar 9.



Gambar 9. Operasi Konvolusi (Ghosh dkk., 2019).

Setelah melakukan operasi konvolusi lengkap, berikut ini ditampilkan *feature map* hasil akhirnya yang dapat dilihat pada Gambar 10.

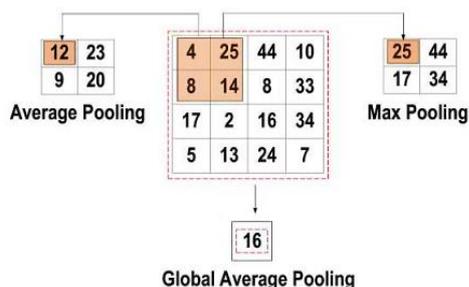
1	0	4
4	1	1
1	1	2

Gambar 10. Setelah Operasi Konvolusi (Ghosh dkk., 2019).

### c. Lapisan *Pooling*

Menurut Alzubaidi dkk., (2021), tugas utama lapisan *pooling* adalah *sub-sampling* dari *feature map*. *Map* ini dihasilkan dengan mengikuti operasi konvolusi. Pendekatan ini memperkecil *feature map* berukuran besar untuk membuat *feature map* yang lebih kecil. Operasi *pooling* dilakukan dengan cara mempertahankan sebagian besar informasi (fitur) dominan di setiap langkah (*stride*). Operasi *pooling* dilakukan setelah menentukan ukuran langkah dan kernel terlebih dahulu. Beberapa

jenis metode *pooling* yang sering digunakan adalah *average pooling* (mengambil nilai rata-rata dari *feature map* tergantung ukuran *pooling*), *max pooling* (mengambil nilai terbesar dari *feature map* tergantung ukuran *pooling*), dan *global average pooling* (mengambil nilai rata-rata keseluruhan dari *feature map*). Ilustrasi dari tiga metode *pooling* dirangkum pada Gambar 11.



Gambar 11. Tiga Metode Lapisan *Pooling* (Alzubaidi dkk., 2021).

#### d. Lapisan *Fully Connected*

Lapisan *fully connected* terletak di ujung setiap arsitektur CNN. Pada lapisan ini, setiap *neuron* terhubung dengan semua *neuron* pada lapisan sebelumnya, yang disebut dengan pendekatan *Fully Connected* (FC) (Alzubaidi dkk., 2021).

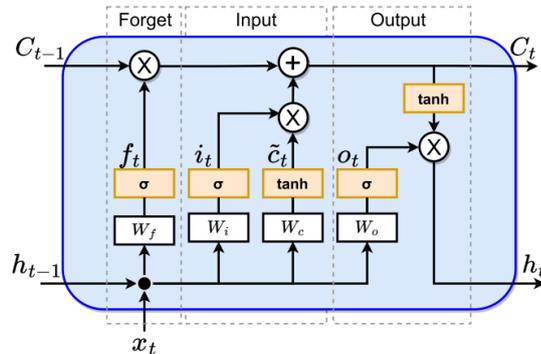
#### e. Lapisan *output*

Lapisan *output* pada CNN menghasilkan label yang menjadi solusi dari permasalahan.

### 2.12 *Long Short-Term Memory*

*Long Short-Term Memory* (LSTM) merupakan jenis struktur *Recurrent Neural Networks* (RNN) khusus yang mengatasi kelemahan RNN dalam menyimpan memori jangka panjang (Fan dkk., 2020). Model LSTM dapat mempelajari dependensi jangka panjang menggunakan mekanisme yang disebut *gates*. *Gates* ini

dapat mempelajari informasi apa saja yang penting untuk disimpan atau dibuang (Mousavi dkk., 2021). Arsitektur LSTM terdiri dari beberapa *gates*, yaitu *forget gate* ( $f_t$ ), *input gate* ( $i_t$ ), dan *output gate* ( $o_t$ ) yang dapat dilihat pada Gambar 12.



Gambar 12. Arsitektur Jaringan LSTM (Bohara dkk., 2022).

a. *forget gate* ( $f_t$ )

Langkah pertama dalam membangun jaringan LSTM adalah mengidentifikasi informasi yang tidak diperlukan dan akan dilupakan. Proses identifikasi dan pengecualian data ini ditentukan oleh fungsi *sigmoid* yang mengambil *output* dari unit LSTM terakhir ( $h_{t-1}$ ) pada waktu  $t-1$  dan *input* saat ini ( $x_t$ ) pada waktu  $t$ . *Gates* ini disebut *forget gate* ( $f_t$ ), di mana  $f_t$  adalah vektor dengan nilai berkisar antara 0 hingga 1, sesuai dengan setiap angka dalam keadaan sel  $C_{t-1}$ . Perhitungan pada *forget gate* ( $f_t$ ) dapat dihitung dengan menggunakan Persamaan (2.9).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.9)$$

Keterangan:

- $\sigma$  = Fungsi *sigmoid*
- $W_f$  = Nilai bobot pada *forget gate*
- $b_f$  = Nilai bias pada *forget gate*

b. *input gate* ( $i_t$ )

Langkah selanjutnya adalah menyimpan informasi dari *input* baru ( $X_t$ ) pada *cell state* serta memperbarui *cell state*. Langkah ini terdiri dari dua bagian, lapisan *sigmoid* dan *tanh*. Lapisan *sigmoid* memutuskan apakah informasi baru harus diperbarui atau diabaikan (0 atau 1). *Input gate* ( $i_t$ ) dapat dihitung dengan menggunakan Persamaan (2.10).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2.10)$$

Keterangan:

- $\sigma$  = Fungsi *sigmoid*
- $W_i$  = Nilai bobot pada *input gate*
- $b_i$  = Nilai bias pada *input gate*

Kemudian, fungsi *tanh* memberi bobot pada nilai yang dilewati, menentukan tingkat kepentingannya (-1 hingga 1) yang ditunjukkan pada Persamaan (2.11).

$$N_t = \tanh(W_n[h_{t-1}, x_t] + b_n) \quad (2.11)$$

Keterangan:

- $N_t$  = *Input* baru
- tanh* = Fungsi *tangen hiperbolik*
- $W_n$  = Nilai bobot yang baru
- $b_n$  = Nilai bias yang baru

Kedua nilai tersebut dikalikan untuk memperbarui *cell state*. *Input* baru ini kemudian ditambahkan ke *input* lama ( $C_{t-1}$ ). Sehingga, menghasilkan *cell state* ( $C_t$ ) pada Persamaan (2.12).

$$C_t = C_{t-1}f_t + N_t i_t \quad (2.12)$$

Keterangan:

$C_t$  = *Cell state*

$C_{t-1}$  = *Cell state* sebelum orde ke t

$f_t$  = *Forget gate*

$i_t$  = *Input gate*

c. *output gate* ( $o_t$ )

Pada langkah terakhir, *output values* ( $h_t$ ) didasarkan pada *output cell state* ( $o_t$ ). Lapisan *sigmoid* menentukan bagian mana dari *cell state* yang akan dijadikan *output*. Berikut merupakan *output gate* ( $o_t$ ) yang dapat dilihat pada Persamaan (2.13).

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.13)$$

Keterangan:

$\sigma$  = Fungsi *sigmoid*

$W_o$  = Nilai bobot pada *output gate*

$b_o$  = Nilai bias pada *output gate*

Selanjutnya, *output* dari *sigmoid gates* ( $o_t$ ) dikalikan dengan nilai baru yang dibuat oleh lapisan *tanh* dari *cell state* ( $C_t$ ), dengan nilai berkisar antara  $-1$  dan  $1$  yang ditunjukkan oleh Persamaan (2.14).

$$h_t = O_t \tanh(C_t) \quad (2.14)$$

Keterangan:

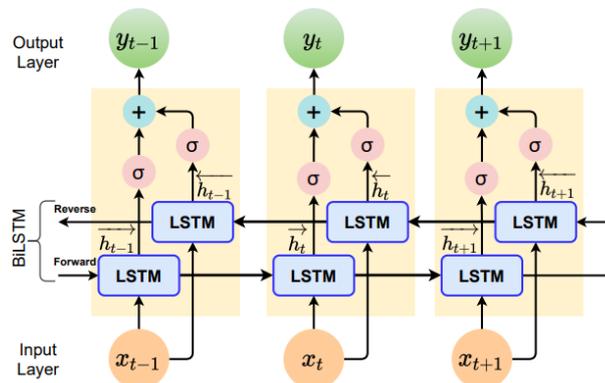
$O_t$  = *Output gate*

*tanh* = Fungsi *tangen hiperbolik*

$C_t$  = *Cell state*

### 2.13 Bidirectional Long Short-Term Memory

*Bidirectional Long Short-Term Memory* (BiLSTM) merupakan penyempurnaan dari jaringan LSTM yang bertujuan untuk meningkatkan kinerja model pada masalah klasifikasi urutan (Hidayatullah dkk., 2020). Berbeda dengan model LSTM tunggal yang hanya dapat menangkap informasi keadaan sebelumnya tetapi tidak dapat menangkap informasi keadaan masa depan. Jaringan BiLSTM melatih urutan *input* menggunakan dua jaringan LSTM. Jaringan BiLSTM menggabungkan antara dua lapisan tersembunyi, yaitu *forward* and *backward*. Jaringan BiLSTM dapat menangkap informasi dari konteks sebelumnya dan masa depan untuk memprediksi urutan langkah selanjutnya. Arsitektur jaringan model BiLSTM dapat dilihat pada Gambar 13.



Gambar 13. Arsitektur Jaringan BiLSTM (Bohara dkk., 2022).

Menurut Bohara dkk., (2022), jaringan BiLSTM menerima rangkaian *input*  $x = (x_1, x_2, \dots, x_n)$ , dengan  $n$  mewakili panjang rangkaian. Perhitungan menggunakan lapisan tersembunyi *forward* dan *backward* pada jaringan BiLSTM ditunjukkan pada Persamaan (2.15).

$$\begin{aligned} \vec{h}_t &= \sigma(w_{\vec{h}x}x_t + w_{\vec{h}h}h_t + b_{\vec{h}}) \\ \overleftarrow{h}_t &= \sigma(w_{\overleftarrow{h}x}x_t + w_{\overleftarrow{h}h}h_t + b_{\overleftarrow{h}}) \end{aligned} \quad (2.15)$$

Keterangan:

$\vec{h}_t = (\vec{h}_1, \vec{h}_2, \dots, \vec{h}_n)$  lapisan tersembunyi *forward*

$\overleftarrow{h}_t = (\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_n)$  lapisan tersembunyi *backward*

Selanjutnya, kedua lapisan tersembunyi digabungkan untuk menghitung vektor *output* akhir yang ditunjukkan pada Persamaan (2.16).

$$y_t = w_{y\vec{h}} \vec{h}_t + w_{y\overleftarrow{h}} \overleftarrow{h}_t + b_y \quad (2.16)$$

Keterangan:

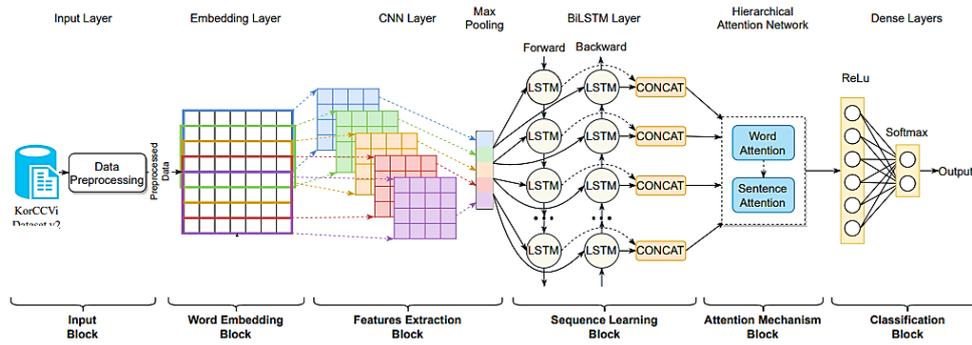
$y_t = f(\vec{h}_t, \overleftarrow{h}_t)$  vektor *output*

## 2.14 Hybrid CNN-BiLSTM

Menurut Bharal dan Krishna (2021), kombinasi model CNN dan Bi-LSTM masing-masing memiliki arsitektur spesifik dan keunggulannya tersendiri.

- Model CNN dikenal dengan ekstraksi fitur semaksimal mungkin dari teks *input* menggunakan lapisan *max pooling*.
- Model Bi-LSTM menyimpan informasi dalam memori dari masa depan dan menggunakan dua lapisan tersembunyi bersama-sama untuk menyimpan informasi dari masa lalu dan masa depan serta memiliki kemampuan untuk mengabaikan informasi tekstual yang tidak perlu menggunakan *forget gate*.

Tujuan dari penggabungan kedua model ini adalah untuk menghasilkan model *hybrid* yang memanfaatkan keunggulan model CNN dan Bi-LSTM. Model *hybrid* ini diharapkan mampu menangkap fitur maksimal yang diekstraksi menggunakan CNN dan menggunakannya sebagai *input* Bi-LSTM (Bharal dan Krishna, 2021). Arsitektur jaringan model *hybrid* CNN-BiLSTM dapat dilihat pada Gambar 14.



Gambar 14. Arsitektur Jaringan CNN-BiLSTM (Boussougou dan Park, 2023).

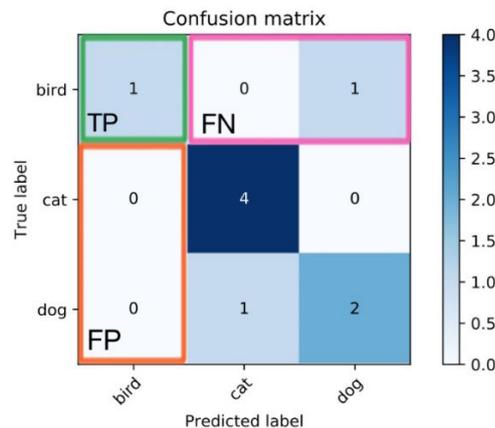
## 2.15 Evaluasi Model

Evaluasi kinerja model dilakukan untuk mengukur seberapa baik kinerja model klasifikasi yang telah dibangun, yaitu TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*). Nilai TP merepresentasikan jumlah kelas positif yang diprediksi dengan benar ke dalam kelas positif, TN merepresentasikan jumlah kelas negatif yang diprediksi dengan benar ke dalam kelas negatif, FP merepresentasikan proporsi kelas negatif yang diprediksi ke dalam kelas positif, dan FN merepresentasikan proporsi kelas positif yang diprediksi ke dalam kelas negatif (Adi dan Wintarti, 2022). Keempat nilai tersebut terdapat pada *confusion matrix*. *Confusion matrix* merupakan matriks yang digunakan untuk menampilkan prediksi klasifikasi dan klasifikasi aktual. *Confusion matrix* dibagi menjadi *confusion matrix* dasar dan *confusion matrix multiclass*. *Confusion matrix* dasar berdimensi 2x2 yang terdiri dari baris dan kolom, dengan baris merupakan kelas prediksi dan kolom merupakan kelas aktualnya. Terdapat dua kelas yang digunakan dalam *confusion matrix* dasar yaitu kelas positif dan negatif (Arifiyanti dkk., 2018). *Confusion matrix* dasar dapat dilihat pada Gambar 15.

		Predicted Values	
		POSITIVE	NEGATIVE
Actual Values	POSITIVE	<b>TP</b> (True positive)	<b>FN</b> (False negative)
	NEGATIVE	<b>FP</b> (False positive)	<b>TN</b> (True negative)

Gambar 15. *Confusion Matrix* Dasar (Habibah dkk., 2023).

Berbeda dengan *confusion matriks* dasar, *confusion matrix multiclass* hanya terdiri dari TP (*True Positive*), FP (*False Positive*), dan FN (*False Negative*) (Habibah dkk., 2023). Nilai TP untuk *confusion matrix multiclass* diperoleh dari diagonal utama matriks, FN pada baris matriks, dan FP pada kolom matriks. *Confusion matrix multiclass* dapat dilihat pada Gambar 16.



Gambar 16. *Confusion Matrix Multiclass* (Habibah dkk., 2023).

Berdasarkan *confusion matrix* tersebut diperoleh nilai *accuracy*, *precision*, *recall* dan *f1-score* pada model klasifikasi sebagai salah satu acuan apakah model yang dibangun memiliki kinerja yang baik.

### 2.15.1 Accuracy

*Accuracy* adalah *metric* yang sering digunakan untuk mengevaluasi kinerja model klasifikasi. Persentase dari keseluruhan data yang diprediksi secara akurat selama proses klasifikasi disebut sebagai *accuracy* (Hossin dan Sulaiman, 2015). *Accuracy* secara matematis dinyatakan dalam Persamaan (2.17).

$$Accuracy = \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N TP_i + FP_i + FN_i} \quad (2.17)$$

### 2.15.2 Precision

*Precision* adalah rasio prediksi positif sejati dibandingkan dengan seluruh hasil yang diprediksi positif (Hossin dan Sulaiman, 2015). *Precision* mengukur sejauh mana model dapat mengidentifikasi kelas positif dengan benar. *Precision* secara matematis dinyatakan dalam Persamaan (2.18).

$$Precision = \frac{TP_i}{TP_i + FP_i} \quad (2.18)$$

### 2.15.3 Recall

*Recall* adalah rasio prediksi positif sejati dibandingkan dengan jumlah data positif yang sebenarnya (Hossin dan Sulaiman, 2015). *Recall* mengukur sejauh mana keberhasilan model dapat menemukan semua kelas positif yang seharusnya ada. *Recall* secara matematis dinyatakan dalam Persamaan (2.19).

$$Recall = \frac{TP_i}{TP_i + FN_i} \quad (2.19)$$

#### 2.15.4 *F1-score*

*F1-score* adalah rata-rata harmonik dari *recall* dan presisi. *F1-score* biasanya digunakan pada data yang memiliki kelas yang tidak seimbang (*imbalance*) (Hossin dan Sulaiman, 2015). *F1-score* secara matematis dinyatakan dalam Persamaan (2.20).

$$F1\text{-score} = \frac{2 \times Recall_i \times Precision_i}{(Recall_i + Precision_i)} \quad (2.20)$$

### III. METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Waktu dan tempat penelitian ini sebagai berikut:

##### a. Waktu Penelitian

Penelitian ini dilakukan dimulai pada Semester Ganjil tahun akademik 2023/2024, tepatnya pada bulan Agustus 2023. Penelitian ini dibagi menjadi tiga tahap, tahap pertama dimulai dari studi literatur pengumpulan jurnal ataupun buku yang berkaitan dengan topik penelitian yang akan digunakan sebagai referensi dalam penyusunan proposal penelitian hingga akhir. Selanjutnya, yaitu pengumpulan data yang akan digunakan sebagai bahan untuk dilakukan penelitian dan penyusunan draf proposal penelitian. Tahap ke-2 merupakan tahap pengerjaan program mulai dari *exploratory data analysis*, *preprocessing data*, *word embedding*, *resampling data*, Pembagian data, membangun model *hybrid CNN-BiLSTM*, dan evaluasi kinerja model. Tahap terakhir pada penelitian ini yaitu penyusunan hasil pengujian dan analisis dalam bentuk draf laporan yang akan disampaikan pada seminar hasil serta sidang komprehensif. Waktu dan kegiatan dalam penelitian ini secara umum dirangkum pada Tabel 2.

##### b. Tempat Penelitian

Penelitian ini dilakukan secara studi pustaka di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Lokasi bertempat di Jalan Prof. Dr. Soemantri Brojonegoro No. 1, Gedong Meneng, Bandar Lampung.

### 3.2 Data dan Alat Penelitian

Data dan alat yang digunakan pada penelitian ini sebagai berikut:

#### a. Data

Data yang digunakan pada penelitian ini merupakan data ulasan aplikasi Halodoc yang dikumpulkan berdasarkan komentar dari pengalaman pengguna. Data ini diperoleh dari <https://play.google.com/store/apps>. Pengambilan data dari situs *Google Play Store* tersebut dilakukan menggunakan bahasa pemrograman *Python* dengan memanfaatkan *library google-play-scrapper* dengan parameter bahasa dari ulasan aplikasi yaitu bahasa Indonesia (id), asal negara Indonesia (id), ulasan yang diambil berdasarkan urutan relevansi tertinggi, dan jumlah data ulasan yang akan diambil yaitu sebanyak 10000 data. Kemudian data yang digunakan pada penelitian ini di bagi dua, yaitu data pada saat COVID-19 dan setelah COVID-19. Data ini merupakan data teks berupa ulasan pengguna aplikasi kesehatan Halodoc. Ada dua variabel yang diambil, yaitu ulasan dari pengguna aplikasi dan rating yang terdiri dari rating 1 sampai 5. Kemudian, data yang digunakan disimpan dengan format file *csv*. Sampel data ulasan pengguna aplikasi Halodoc pada saat COVID-19 disajikan pada Tabel 2.

Tabel 2. Data Ulasan Pengguna Aplikasi Halodoc Saat COVID-19

Date	Rating	Ulasan
2020-09-12	5	Aplikasinya bagus, dokternya juga ramah pesan saya ...
2020-09-12	5	Terimakasih, aplikasi yg sangat membantu dan ...
2020-09-12	5	Saya sangat puas dengan aplikasi ini.dokternya ...
...	...	...
2021-05-16	3	Sedikit kecewa sih, konsul di batasi 30 mnit, baru di ...
2021-05-16	5	Terimakasih halodoc, berkat adanya aplikasi ini ...
2021-05-16	4	Untuk konsultasi sangat membantu,hanya saja ...

Data pada Tabel 2 merupakan data ulasan pengguna aplikasi Halodoc pada saat COVID-19. Data ini memiliki 5000 ulasan yang terdiri dari 251 ulasan merupakan rating 1, rating 2 yang terdiri dari 89 ulasan, rating 3 yang terdiri dari 184 ulasan, rating 4 yang terdiri dari 567 ulasan, dan rating 5 yang terdiri dari 3909 ulasan. Data ulasan pengguna aplikasi Halodoc setelah COVID-19 disajikan pada Tabel 3.

Tabel 3. Data Ulasan Pengguna Aplikasi Halodoc Setelah COVID-19

Date	Rating	Ulasan
2022-05-16	5	Sangat membantu.. Pendaftaran ke rumah sakit ...
2022-05-16	5	Bener bener ngebantu sih aplikasinya juga lebih ...
2022-05-16	5	aku sagat suka aplikasi ini dokternya ramah lagi ...
...	...	...
2023-10-17	3	Saya puas atas info dari dokter. Terima kasih, ...
2023-10-17	5	Sangat membantu, suka dengan pelayanan dokter ...
2023-10-17	4	Respon yang sangat cepat, ...

Data pada Tabel 3 merupakan data ulasan pengguna aplikasi Halodoc setelah COVID-19. Data ini memiliki 5000 ulasan yang terdiri dari 897 ulasan merupakan rating 1, rating 2 yang terdiri dari 189 ulasan, rating 3 yang terdiri dari 187 ulasan, rating 4 yang terdiri dari 367 ulasan, dan rating 5 yang terdiri dari 3363 ulasan.

#### b. Alat

Adapun alat yang digunakan dalam penelitian ini yaitu sebagai berikut:

##### 1. Perangkat keras (*Hardware*)

Perangkat keras yang digunakan pada penelitian ini yaitu:

- Processor Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz (4 CPUs) ~ 2.49 GHz
- Installed RAM 4,00 GB

## 2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan pada penelitian ini yaitu sebagai berikut:

- Sistem Operasi Windows 10 Pro 64-bit
- *Jupyter Notebook* 6.5.4 (*Python* 3.10)

Adapun *package* yang digunakan dalam penelitian ini yang terdiri dari sebagai berikut:

- *Tensorflow* 2.12.0

*Tensorflow* adalah *package* yang digunakan untuk membangun model *deep learning*.

- *Scikit-learn* 1.2.2

*Scikit-learn* adalah *package* yang digunakan untuk melakukan *preprocessing* data dan evaluasi model.

- *NumPy* 1.23.5

*NumPy* adalah *package* yang digunakan untuk *array* dan matriks multidimensi.

- *Pandas* 1.5.3

*Pandas* adalah *package* yang digunakan untuk membaca dan menganalisis data yang dimiliki.

- *Matplotlib* 3.7.1

*Matplotlib* adalah *package* yang digunakan untuk membuat visualisasi data yang berkualitas tinggi, seperti grafik, bar plot, dan lain-lain.

- *Seaborn* 0.12.2

*Seaborn* adalah *package* yang digunakan untuk membuat visualisasi tingkat tinggi.

- *Imblearn* 0.10.1

*Imblearn* adalah *package* yang digunakan untuk menangani masalah ketidakseimbangan kelas pada data.

- *Gensim* 4.3.1

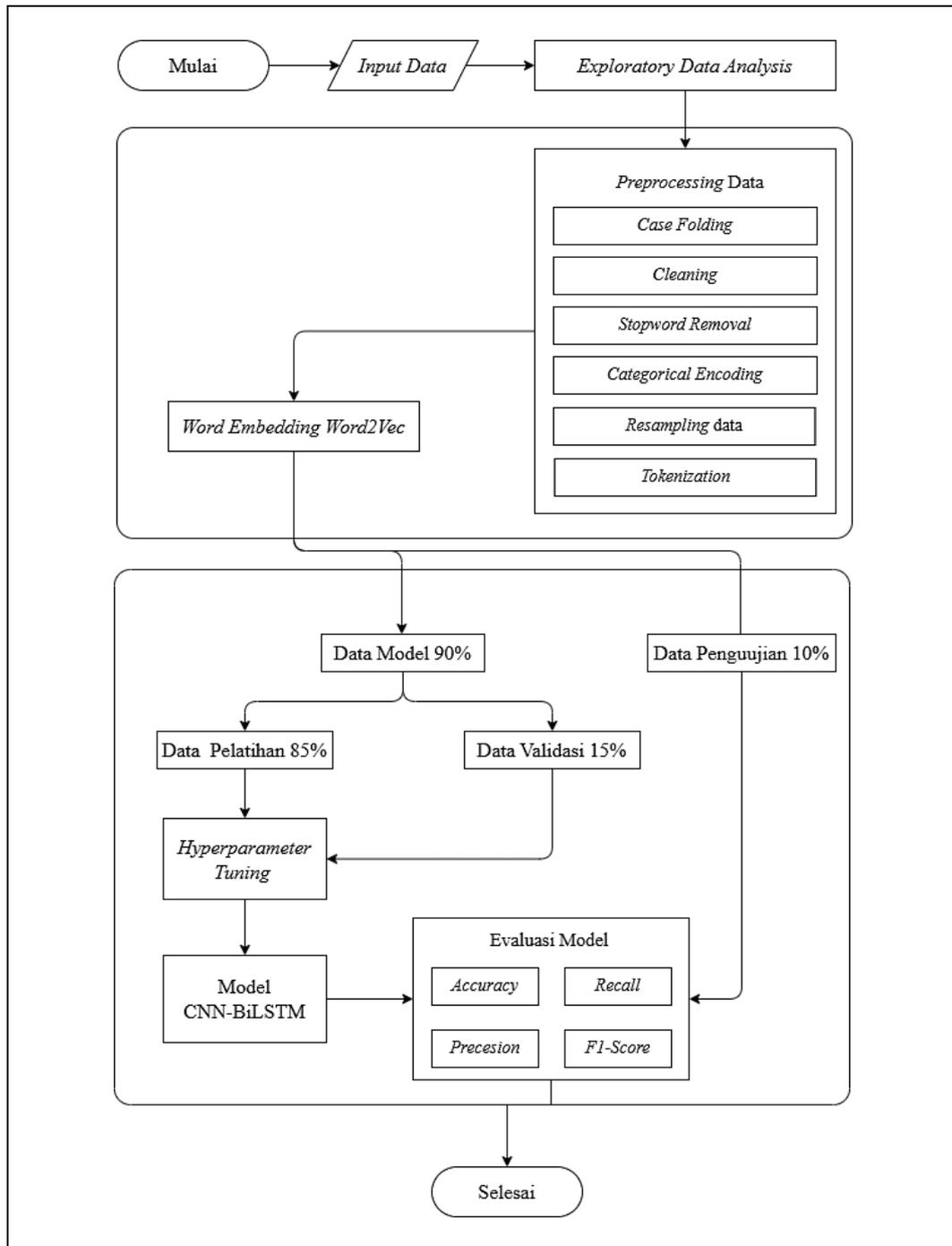
*Gensim* adalah *package* yang digunakan untuk pemrosesan bahasa alami, dan analisis kesamaan semantik.

- *Nltk* 3.8.1

*Nltk* adalah *package* yang juga digunakan untuk pemrosesan bahasa alami.

### 3.3 Alur Kerja Penelitian

Penelitian ini dilakukan berdasarkan alur penelitian pada Gambar 17.



Gambar 17. Flowchart Metodologi Penelitian Model Hybrid CNN-BiLSTM.

Berdasarkan alur penelitian pada Gambar 17, berikut penjelasan setiap tahap yang dilakukan.

1. Data ulasan aplikasi Halodoc

Data yang digunakan pada penelitian ini merupakan data tentang ulasan aplikasi Halodoc. Data ini diperoleh dari <https://play.google.com/store/apps>. Data yang digunakan pada penelitian ini di bagi menjadi dua data, yaitu data pada saat COVID-19 dan setelah COVID-19. Data ini merupakan data teks yang berupa ulasan pengguna aplikasi kesehatan Halodoc. Data saat COVID-19 memiliki 5000 ulasan yang terdiri dari 251 ulasan merupakan rating 1, rating 2 yang terdiri dari 89 ulasan, rating 3 yang terdiri dari 184 ulasan, rating 4 yang terdiri dari 567 ulasan, dan rating 5 yang terdiri dari 3909 ulasan. Sedangkan, data setelah COVID-19 memiliki 5000 ulasan yang terdiri dari 897 ulasan merupakan rating 1, rating 2 yang terdiri dari 189 ulasan, rating 3 yang terdiri dari 187 ulasan, rating 4 yang terdiri dari 367 ulasan, dan rating 5 yang terdiri dari 3363 ulasan.

2. *Exploratory Data Analysis*

*Exploratory Data Analysis* (EDA) dalam penelitian ini bertujuan untuk meningkatkan pemahaman terhadap data sebelum melakukan analisis statistik lebih lanjut ataupun membangun suatu model.

3. *Preprocessing* data

*Preprocessing* data yang dilakukan dalam penelitian ini terdiri beberapa tahapan.

a. *Case folding*

*Case folding* dalam penelitian ini bertujuan untuk mengubah huruf kapital menjadi huruf kecil.

b. *Cleaning* data

*Cleaning* data yang dilakukan meliputi *remove punctuation* (tanda baca), emoji, angka, dan spasi yang berlebih.

c. *Stopword removal*

*Stopword removal* bertujuan untuk menghapus kata-kata yang bersifat umum dan dianggap tidak memiliki makna berarti.

d. *Categorical encoding*

*Categorical encoding* dalam penelitian ini dilakukan agar dapat membantu mesin dalam memahami data.

e. *Resampling data*

Jumlah kelas pada variabel rating dari ulasan aplikasi Halodoc pada saat maupun setelah COVID-19 memiliki perbedaan yang menyebabkan masalah *imbalance data*. Oleh karena itu, *resampling data* dalam penelitian ini dilakukan bertujuan untuk mengatasi masalah *imbalance data*. Metode *resampling* yang digunakan pada penelitian ini adalah *random oversampling*.

f. *Tokenization*

*Tokenization* digunakan untuk memisahkan suatu kalimat menjadi beberapa bagian kata atau yang biasa disebut token.

4. *Word embedding*

Setelah dilakukan *Tokenization* pada tahap *preprocessing data*, langkah selanjutnya adalah *word embedding* yang bertujuan untuk merepresentasikan kata menjadi bentuk vektor menggunakan *Word2Vec*.

5. Pembagian data

Pembagian data dalam penelitian ini dilakukan dengan membagi data menjadi data model dan data pengujian. Komposisi yang digunakan pada penelitian ini yaitu data model 90% dan data pengujian 10%. Kemudian, selanjutnya data model dibagi kembali menjadi data pelatihan dan data validasi dengan komposisi 85% untuk data pelatihan dan 15% untuk data validasi.

6. *Hyperparameter tuning*

Melakukan *hyperparameter tuning* dengan menggunakan metode *grid search* untuk menentukan parameter *batch size* dan *learning rate* yang digunakan dalam membangun model *hybrid CNN-BiLSTM*. Penggunaan *grid search* bertujuan agar parameter terbaik yang ditemukan untuk membangun model *hybrid CNN-BiLSTM* dapat lebih akurat.

7. Membangun model *hybrid CNN-BiLSTM*

Setelah didapatkan parameter terbaik hasil dari *hyperparameter tuning*, langkah selanjutnya adalah membangun model *hybrid CNN-BiLSTM*

berdasarkan parameter terbaik yang telah didapatkan. Arsitektur model yang digunakan pada penelitian ini terdiri dari lapisan *input*, lapisan *embedding*, lapisan CNN, lapisan *pooling* menggunakan *max pooling*, lapisan BiLSTM, *flatten*, lapisan *fully connected*, dan lapisan *output*.

#### 8. Evaluasi model

Evaluasi model dalam penelitian ini dilakukan dengan menggunakan *confusion matrix*. Berdasarkan *confusion matrix* tersebut dihitung *metric* evaluasi *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil perhitungan dari *metric-metric* tersebut menunjukkan seberapa baik model yang telah dibangun dalam mengklasifikasikan rating ulasan pengguna aplikasi Halodoc.

## IV. PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan dapat diambil kesimpulan untuk penelitian ini sebagai berikut.

1. Peran dari aplikasi Halodoc pada saat COVID-19 memiliki tingkat kepuasan yang lebih baik dibandingkan setelah COVID-19. Hal ini karena tingkat kebutuhan pengguna yang lebih tinggi saat COVID-19 dibandingkan dengan setelah COVID-19.
2. Penggunaan model *hybrid* yang menggabungkan CNN dan BiLSTM untuk klasifikasi rating ulasan aplikasi Halodoc memiliki kinerja yang baik.
3. Model *hybrid* CNN-BiLSTM dengan *hyperparameter tuning* diperoleh nilai parameter terbaik untuk data saat COVID-19 maupun setelah COVID-19 yaitu *batch size* 64 dan *learning rate*  $1 \times 10^{-3}$  yang menghasilkan nilai *accuracy* 98,71% untuk data saat COVID-19 dan 98,16% untuk data setelah COVID-19. Grafik *loss* menunjukkan nilai *loss* dan validasi *loss* searah, yaitu nilainya semakin menurun pada setiap *epoch*-nya. Hal tersebut menunjukkan bahwa model *hybrid* CNN-BiLSTM memiliki kinerja yang baik dalam melakukan klasifikasi rating ulasan aplikasi Halodoc.

## 5.2 Saran

Berdasarkan hasil dari penelitian ini, adapun saran yang dapat diberikan diantaranya yaitu sebagai berikut.

1. Menggunakan metode *resampling* data seperti SMOTE atau *resampling hybrid* yang menggabungkan *random oversampling* dan *random undersampling* untuk meminimalisir adanya pencilan pada saat mengatasi permasalahan *imbalance* data menggunakan metode *random oversampling*.
2. Penambahan jumlah *hyperparameter* yang disediakan pada saat proses *hyperparameter tuning* untuk meningkatkan keakuratan dalam menentukan parameter yang akan digunakan untuk membangun model.

## DAFTAR PUSTAKA

- Adi, S., dan Wintarti, A. 2022. Komparasi Metode Support Vector Machine (SVM), K-Nearest Neighbors (KNN), dan Random Forest (RF) untuk Prediksi Penyakit Gagal Jantung. *Jurnal Ilmiah Matematika*. **10**(2): 258–268.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Dujaili, A. Al, Duan, Y., Shamma, O. Al, Santamaría, J., Fadhel, M. A., Amidie, M. Al, dan Farhan, L. 2021. Review of deep learning : concepts , CNN architectures , challenges , applications , future directions. In *Journal of Big Data*. **53**(8): 1-74.
- Arifiyanti, A. A., Pradana, R. M., dan Novian, I. F. 2018. Klasifikasi Produk Retur dengan Algoritma Pohon Keputusan C4.5. *Jurnal IPTEK*. **22**(1): 79-86.
- Bellmann, P., Lausser, L., Kestler, H. A., dan Schwenker, F. 2022. A Theoretical Approach to Ordinal Classification: Feature Space-Based Definition and Classifier-Independent Detection of Ordinal Class Structures. *Applied Sciences*. **12**(4): 1–23.
- Bharal, R., dan Krishna, O. V. V. 2021. Social Media Sentiment Analysis Using CNN-BiLSTM. *International Journal of Science and Research (IJSR)*. **10**(9): 656–661.
- Bohara, B., Fernandez, R. I., Gollapudi, V., dan Li, X. 2022. Short-Term Aggregated Residential Load Forecasting using BiLSTM and CNN-BiLSTM. *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies*. **7**(2): 37–43.
- Boussougou, M. K. M., dan Park, D.-J. 2023. Attention-Based 1D CNN-BiLSTM Hybrid Model Enhanced with FastText Word *Embedding* for Korean Voice Phishing Detection †. *Mathematics*. **11**(3217): 1–25.
- Elgeldawi, E., Sayed, A., Galal, A. R., dan Zaki, A. M. 2021. Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. *Informatics*. **8**(4): 1–21.
- Fan, H., Jiang, M., Xu, L., Zhu, H., Cheng, J., dan Jiang, J. 2020. Comparison of long short term memory networks and the hydrological model in runoff simulation. *Water*. **12**(1): 1–15.

- Gao, D., Liu, X., Zhu, Z., dan Yang, Q. 2022. A hybrid CNN-BiLSTM approach for remaining useful life prediction of EVs lithium-Ion battery. *Measurement and Control*. **10**(5): 1-13.
- Geron, A. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media., Gravenstein Highway North, Sebastopol.
- Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., dan De, D. 2019. Fundamental Concepts of Convolutional Neural Network. In *Intelligent Systems Reference Library*. **172**(1): 519-567.
- Habibah, D., Aini, N., Kurniasari, D., Nuryaman, A., Usman, M. 2023. Implementation of Artificial Neural Network with Backpropagation Algorithm for Rating Classification on Sales of Blackmores in Tokopedia. *Jurnal Teknik Informatika (JUTIF)*. **4**(2): 365–372.
- Han, J., Pei, J., dan Tong, H. 2023. *Data Mining Concepts and Techniques*. 4<sup>th</sup> Edition. Morgan Kaufmann Elsevier, Cambridge.
- Hidayatullah, A. F., Cahyaningtyas, S., dan Pamungkas, R. D. 2020. Attention-based CNN-BiLSTM for Dialect Identification on Javanese Text. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*. **5**(4): 317–324.
- Hossin, M., dan Sulaiman, M. N. 2015. A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining dan Knowledge Management Process*. **5**(2): 01–11.
- Izzati, V., dan Firmanto, Y. 2021. Analisis Kepuasan Pengguna Aplikasi Kesehatan Halodoc Melalui Model End User Computing Satisfaction Selama Masa Pandemi COVID-19. *Jurnal Ilmiah Mahasiswa Fakultas Ekonomi dan Bisnis*. **6**(1): 1–16.
- Jatnika, D., Arif, M., dan Ardiyanti, A. 2019. Word2Vec Model Analysis for Semantic Similarities in English Words. *Procedia Computer Science*. **157**(1): 160–167.
- Kandel, I., dan Castelli, M. 2020. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*. **6**(4): 312–315.
- Kapil, P., Ekbal, A., dan Das, D. 2020. Investigating Deep Learning Approaches for Hate Speech Detection in Social Media. *International Journal of Advanced Computer Science and Applications (IJACSA)*. **7**(1): 1–12.
- Krishna, M. M., Neelima, M., Harshali, M., dan Rao, M. V. G. 2018. Image classification using Deep learning. *International Journal of Engineering and Technology*. **7**(3): 614–617.

- Mohammed, R., Rawashdeh, J., dan Abdullah, M. 2020. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. *2020 11th International Conference on Information and Communication Systems*. **7**(1): 243–248.
- Mousavi, S. M., Ghasemi, M., Manshadi, M. D., dan Mosavi, A. 2021. Deep Learning for Wave Energy Converter Modeling Using Long Short-Term Memory. *Mathematics Article*. **9**(871): 1–16.
- Nwankpa, C., Ijomah, W., Gachagan, A., dan Marshall, S. 2018. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *2nd International Conference on Computational Sciences and Technologies*. **20**(1): 124–133.
- Rhanoui, M., Mikram, M., Yousfi, S., dan Barzali, S. 2019. A CNN-BiLSTM Model for Document-Level Sentiment Analysis. *Machine Learning and Knowledge Extraction*. **9**(1): 41–49.
- Rifai, W., dan Winarko, E. 2019. Modification of Stemming Algorithm Using A Non Deterministic Approach To Indonesian Text. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*. **13**(4): 379–388.
- Salloum, S. A., Al-Emran, M., Monem, A. A., dan Shaalan, K. 2018. Using text mining techniques for extracting information from research articles. *Studies in Computational Intelligence*. **740**(1): 373–397.
- Sarker, I. H. 2021. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*. **2**(3): 1–21.
- Shan, L., Liu, Y., Tang, M., Yang, M., dan Bai, X. 2021. CNN-BiLSTM Hybrid Neural Networks with Attention Mechanism for Well log Prediction. *Journal of Petroleum Science and Engineering*. **205**(12): 1–21.
- Sharma, S., Sharma, S., dan Anidhya, A. 2020. Understanding Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*. **4**(12): 310–316.
- Surjandari, I., Megawati, C., Dhini, A., dan Sanditya Hardaya, I. B. N. 2016. Application of text mining for classification of textual reports: A study of Indonesia's national complaint handling system. *Proceedings of the International Conference on Industrial Engineering and Operations Management*. **8**(3): 1147–1156.
- Wongvorachan, T., He, S., dan Bulut, O. 2023. A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. *Information*. **14**(1): 1-15.

- Xiaoyan, L., Raga, R. C., dan Xuemei, S. 2022. GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews. *Journal of Sensors*. **8**(2): 5073–5081.
- Yu, H., Yang, L. T., Zhang, Q., Armstrong, D., dan Deen, M. J. 2021. Convolutional Neural Networks for Medical Image Analysis: State-of-the-art, comparisons, improvement and perspectives. *Neurocomputing*. **444**(1): 92–110.