

**FITUR EKSTRAKSI *NEURAL NET* UNTUK OPTIMALISASI
PERFORMA MLP PADA TUGAS KLASIFIKASI SENTIMEN**

Skripsi

Oleh

**MUHAMMAD FIKRI ALAM
NPM. 2117031005**



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

2025

ABSTRACT

NEURAL NET FEATURE EXTRACTION FOR MLP PERFORMANCE OPTIMIZATION ON SENTIMENT CLASSIFICATION TASK

By

Muhammad Fikri Alam

MLP is a deep learning model that is considered to have many limitations compared to other models. This is due to the MLP's constraints in solving complex tasks and its tendency to overfit, especially when dealing with high-dimensional data. However, MLP has the advantage of being more resource-efficient compared to other models. Although not superior to other models, MLP can still compete and requires less training time. This is the motivation behind this research. The goal of this study is to improve the performance of the MLP model in sentiment classification tasks by adding neural network layers such as CNN, LSTM, and BiLSTM for feature extraction. The evaluation of the proposed model is conducted using accuracy, precision, recall, F1-score, and ROC-AUC metrics. The results indicate that the addition of feature extraction using an artificial neural network model successfully improved the performance of the MLP by 1% in terms of accuracy, precision, recall, and F1-score, and increased the ROC-AUC by 0.06% to 4%.

Keywords: MLP, deep learning, neural network, performance optimization

ABSTRAK

FITUR EKSTRAKSI *NEURAL NET* UNTUK OPTIMALISASI PERFORMA MLP PADA TUGAS KLASIFIKASI SENTIMEN

Oleh

Muhammad Fikri Alam

MLP adalah model pembelajaran mendalam yang dianggap memiliki banyak kelemahan dibandingkan dengan model lainnya. Hal ini didasarkan pada keterbatasan MLP dalam menyelesaikan tugas-tugas kompleks dan seringnya terjadi overfitting terutama ketika dihadapkan dengan data berdimensi tinggi. Namun, MLP memiliki kelebihan dalam hal penggunaan sumber daya yang lebih efisien dibandingkan model lainnya. Meskipun tidak lebih unggul dari model lain, MLP masih dapat bersaing dan membutuhkan waktu pelatihan yang lebih singkat. Alasan inilah yang mendasari penelitian ini. Tujuan dari penelitian ini adalah untuk meningkatkan kinerja model MLP dalam tugas klasifikasi sentimen dengan menambahkan lapisan jaringan saraf seperti CNN, LSTM, dan BiLSTM untuk ekstraksi fitur. Evaluasi model yang diusulkan menggunakan metrik akurasi, presisi, recall, F1-score, dan ROC-AUC. Hasil penelitian menunjukkan bahwa penambahan fitur ekstraksi menggunakan model jaringan saraf tiruan berhasil meningkatkan performa MLP sebesar 1% pada kategori akurasi, presisi, recall, F1-score, dan berhasil meningkatkan ROC-AUC sebesar 0.06% sampai 4%.

Kata-kata kunci: MLP, pembelajaran mendalam, jaringan saraf, optimalisasi performa

**FITUR EKSTRAKSI *NEURAL NET* UNTUK OPTIMALISASI
PERFORMA MLP PADA TUGAS KLASIFIKASI SENTIMEN**

MUHAMMAD FIKRI ALAM

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam



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

2025

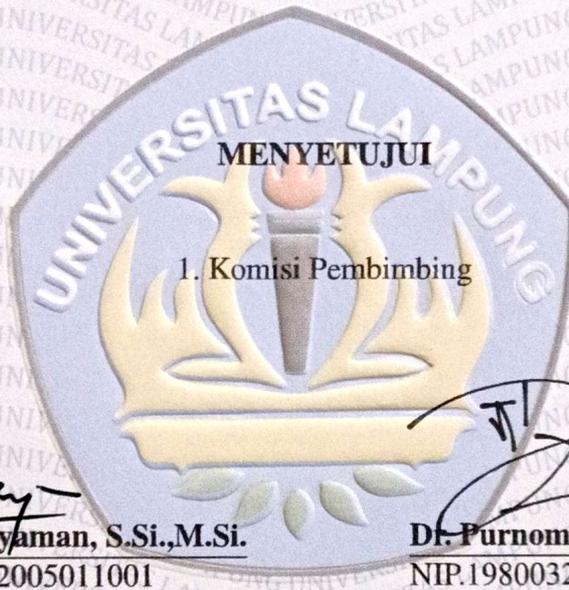
Judul Skripsi : **FITUR EKSTRAKSI NEURAL NET UNTUK OPTIMALISASI PERFORMA MLP PADA TUGAS KLASIFIKASI SENTIMEN**

Nama Mahasiswa : **Muhammad Fikri Alam**

Nomor Pokok Mahasiswa : **2117031005**

Program Studi : **Matematika**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**



1. Komisi Pembimbing

Dr. Aang Nuryaman, S.Si., M.Si.

NIP.197403162005011001

Dr. Purnomo Husnul Khotimah, M.T.

NIP.198003232005022002

2. Ketua Jurusan Matematika

Dr. Aang Nuryaman, S.Si., M.Si.

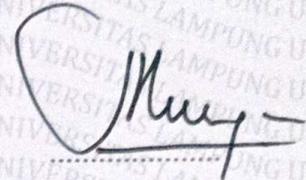
NIP.197403162005011001

MENGESAHKAN

1. tim penguji

Ketua

Dr. Aang Nuryaman, S.Si., M.Si.



Sekretaris

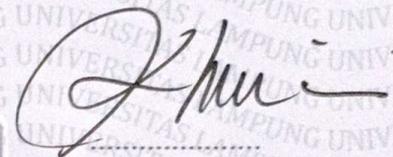
**Dr. Purnomo Husnul Khotimah,
M.T.**



Penguji

Bukan Pembimbing

Dr. Khoirin Nisa, M.Si.

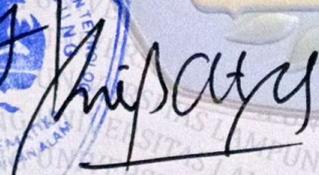


2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 197110012005011002



Tanggal Lulus Ujian Skripsi: **05 Juni 2025**

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama : **Muhammad Fikri Alam**
Nomor Pokok Mahasiswa : **2117031005**
Jurusan : **Matematika**
Judul Skripsi : **Fitur Ekstraksi *Neural Net* Untuk Optimalisasi Performa MLP Pada Tugas Klasifikasi Sentimen**

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri. Apabila kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung,
Penulis,



Muhammad Fikri Alam

RIWAYAT HIDUP

Penulis bernama lengkap Muhammad Fikri Alam, lahir di Bandar Lampung pada tanggal 25 Juni 2002. Penulis merupakan anak ketiga dari tiga bersaudara pasangan Bapak Alamsyah dan Ibu Adiati.

Penulis menempuh pendidikan dasar di SDN 2 Perumnas Way Halim pada tahun 2009-2015. Kemudian penulis melanjutkan jenjang pendidikannya di MTsN 2 Bandar Lampung pada tahun 2015-2018 dan Sekolah Menengah Atas di MAN 1 Bandar Lampung pada tahun 2018-2021. Setelah itu penulis diterima sebagai mahasiswa Program Studi S1 Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN) pada tahun 2021.

Kemudian pada Bulan Desember-Februari 2024 penulis melaksanakan Praktik Kerja Lapangan (PKL) di Badan Riset dan Inovasi Nasional (BRIN) KST Samaun Samadikun yang berlokasi di Bandung. Selanjutnya pada bulan Februari-Juli 2024 penulis melaksanakan magang riset Merdeka Belajar Kampus Merdeka (MBKM) di BRIN KST Samaun Samadikun Bandung.

KATA INSPIRASI

“Thing change, stuff happens, life goes on.”

(Elizabeth Scott)

“Care about what other people think and you will always be their prisoner”

(Lao Tzu)

“As soon as you have made a thought, laugh at it”

(Lao Tzu)

PERSEMBAHAN

Dengan mengucapkan Alhamdulillah dan syukur kepada Allah SWT atas nikmat serta hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya. Dengan rasa syukur dan Bahagia, saya persembahkan rasa terimakasih saya kepada:

Ayah dan Ibuku Tercinta

Terimakasih kepada orang tuaku atas segala pengorbanan, motivasi, doa dan ridho serta dukungannya selama ini. Terimakasih telah memberikan pelajaran berharga kepada anakmu ini tentang makna perjalanan hidup yang sebenarnya sehingga kelak bisa menjadi orang yang bermanfaat bagi banyak orang.

Dosen Pembimbing dan Pembahas

Terimakasih kepada dosen pembimbing dan pembahas yang sudah sangat membantu, memberikan motivasi, memberikan arahan serta ilmu yang berharga.

Sahabat-sahabatku

Terimakasih kepada semua orang-orang baik yang telah memberikan pengalaman, semangat, motivasinya, serta doa-doanya dan senantiasa memberikan dukungan dalam hal apapun.

Almamater Tercinta

Universitas Lampung

SANWACANA

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "Fitur Ekstraksi *Neural Net* Untuk Optimalisasi Performa MLP Pada Tugas Klasifikasi Sentimen" dengan baik dan lancar serta tepat pada waktu yang telah ditentukan. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW.

Dalam proses penyusunan skripsi ini, banyak pihak yang telah membantu memberikan bimbingan, dukungan, arahan, motivasi serta saran sehingga skripsi ini dapat terselesaikan. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terimakasih kepada:

1. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Pembimbing 1 sekaligus Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah banyak meluangkan waktunya untuk memberikan arahan, bimbingan, motivasi, saran serta dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Ibu Dr. Purnomo Husnul Khotimah, M.T. selaku Pembimbing II yang telah memberikan arahan, bimbingan dan dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Ibu Dr. Khoirin Nisa, M.Si. selaku Penguji yang telah bersedia memberikan kritik dan saran serta evaluasi kepada penulis sehingga dapat menjadi lebih baik lagi.
4. Bapak Prof. Drs. Mustofa, M.A., Ph.D. selaku dosen pembimbing akademik.
5. Bapak Andri Fachrur Rozie S.Kom., M.Eng. selaku salah satu pembimbing magang MBKM dari Pusat Riset Sains Data dan Informasi.

6. Papa dan Mama yang selalu memberikan motivasi, semangat, kasih sayang, do'a dan tekanan yang tiada henti-hentinya.
7. Mas idir, Mba nisa, Mas hari, Mba say, serta semua keluarga yang telah memberikan dukungan dalam bentuk semangat, do'a dan sindiran kepada penulis.
8. Jabil, Gading, Randia, dan Amri terima kasih untuk semua dukungan, kenangan yang telah diciptakan dan semua candaan yang sangat berbahaya di setiap saat.
9. Ilma, Arvi, Anggun, dan Falen terima kasih atas semua bantuan, semangat, serta kritikan-kritikan tajam setajam pisau dan pedas sepedas cabai untuk kebaikan penulis.
10. kawan-kawan 24/7 yang menjadi salah satu tempat bersandar dengan *mental disordernya* masing-masing.
11. Anastasia dan Yulina, kawan-kawan seperjuangan di BRIN Bandung yang telah memberikan dukungan dan bantuan dalam penyelesaian penelitian dan penulisan skripsi.
12. Seluruh dosen, staff dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
13. Seluruh peneliti, staff dan karyawan BRIN KST Samaun Samadikun Bandung.
14. Semua pihak yang terlibat dalam penyelesaian skripsi yang tidak dapat penulis sebutkan satu per satu.

Semoga skripsi ini dapat bermanfaat bagi kita semua. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun untuk menjadikan skripsi ini lebih baik lagi.

Bandar Lampung, 05 Juni 2025

Muhammad Fikri Alam

DAFTAR ISI

DAFTAR ISI	xiii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Tujuan Penelitian	5
1.3 Manfaat Penelitian	5
II TINJAUAN PUSTAKA	6
2.1 <i>MultiLayer Perceptron</i>	6
2.2 <i>Convolutional Neural Network</i>	8
2.3 <i>Long Short-Term Memory</i>	10
2.4 <i>Bidirectional Long Short-Term Memory</i>	12
2.5 <i>Word Embedding</i>	14
2.6 TF-IDF	15
2.7 Optimalisasi Performa Model	16
2.8 Metrik Evaluasi Performa	18
2.8.1 <i>Accuracy</i>	18
2.8.2 <i>Precision</i>	19
2.8.3 <i>Recall</i>	19
2.8.4 <i>F1-Score</i>	19
2.8.5 ROC-AUC	19
III METODE PENELITIAN	21
3.1 Waktu dan Tempat Penelitian	21
3.2 Perangkat dan Library Penelitian	21
3.3 Metode Penelitian	21

IV HASIL DAN PEMBAHASAN	24
4.1 Konfigurasi Penelitian	24
4.2 <i>MultiLayer Perceptron</i> (MLP)	26
4.3 <i>Neural Net + MultiLayer Perceptron</i> (MLP)	27
4.3.1 WE+ <i>Convolutional Neural Network</i> (CNN) - <i>MultiLayer Perceptron</i> (MLP)	28
4.3.2 TF-IDF+ <i>Convolutional Neural Network</i> (CNN) - <i>MultiLayer Perceptron</i> (MLP)	31
4.3.3 WE+ <i>Long Short-Term Memory</i> (LSTM) - <i>MultiLayer Perceptron</i> (MLP)	34
4.3.4 TF-IDF+ <i>Long Short-Term Memory</i> (LSTM) - <i>MultiLayer Perceptron</i> (MLP)	37
4.3.5 WE+ <i>Bidirectional Long Short-Term Memory</i> (BiLSTM) - <i>MultiLayer Perceptron</i> (MLP)	40
4.3.6 TF-IDF+ <i>Bidirectional Long Short-Term Memory</i> (BiLSTM) - <i>MultiLayer Perceptron</i> (MLP)	43
4.4 Analisa Hasil Klasifikasi CNN-MLP, LSTM-MLP, dan BiLSTM-MLP	46
V KESIMPULAN DAN SARAN	49
5.1 Kesimpulan	49
5.2 Saran	50
DAFTAR PUSTAKA	51

DAFTAR TABEL

4.1	Performa Klasifikasi MLP	27
4.2	Performa Klasifikasi WE+CNN-MLP	28
4.3	Performa Klasifikasi TF-IDF+CNN-MLP	31
4.4	Performa Klasifikasi WE+LSTM-MLP	34
4.5	Performa Klasifikasi TF-IDF+LSTM-MLP	37
4.6	Performa Klasifikasi WE+BiLSTM-MLP	40
4.7	Performa Klasifikasi TF-IDF+BiLSTM-MLP	43
4.8	Performa Klasifikasi WE+CNN-MLP, LSTM-MLP, BiLSTM-MLP	46
4.9	Performa Klasifikasi TF-IDF+CNN-MLP, LSTM-MLP, BiLSTM-MLP	47

DAFTAR GAMBAR

2.1	Arsitektur MLP	6
2.2	Arsitektur CNN	8
2.3	Arsitektur LSTM	11
2.4	Arsitektur BiLSTM	13
2.5	Diagram Model Hibridasi Oleh (Munandar dkk.,2021	17
3.1	<i>flowchart</i> alur proses penelitian	23
4.1	Distribusi Berita Online Berdasarkan Portal Berita	24
4.2	Perubahan Data Kalimat ke Vektor	26
4.3	WE CNN 17	28
4.4	WE CNN 24	29
4.5	WE CNN 50	29
4.6	WE CNN 100	30
4.7	TFIDF CNN 17	31
4.8	TFIDF CNN 24	32
4.9	TFIDF CNN 50	32
4.10	TFIDF CNN 100	33
4.11	WE LSTM 17	35
4.12	WE LSTM 24	35
4.13	WE LSTM 50	36
4.14	WE LSTM 100	36
4.15	TFIDF LSTM 17	38
4.16	TFIDF LSTM 24	38
4.17	TFIDF LSTM 50	39
4.18	TFIDF LSTM 100	39
4.19	WE BiLSTM 17	41
4.20	WE BiLSTM 24	41

4.21 WE BILSTM 50	42
4.22 WE BILSTM 100	42
4.23 TFIDF BILSTM 17	43
4.24 TFIDF BILSTM 24	44
4.25 TFIDF BILSTM 50	44
4.26 TFIDF BILSTM 100	45
4.27 Distribusi Data Berdasarkan Kelas Sentimen	47

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Multi-Layer Perceptron (MLP) adalah salah satu arsitektur jaringan saraf yang banyak digunakan dalam pembelajaran mesin, termasuk dalam tugas klasifikasi sentimen. Sebagai jaringan saraf *feed-forward*, MLP terdiri dari lapisan *input*, *output*, dan satu atau lebih *hidden layer*, yang berfungsi untuk memproses informasi secara terarah. Dalam tugas klasifikasi sentimen, MLP sering menghadapi tantangan karena keterbatasannya dalam menangkap hubungan hierarkis dan ketergantungan jangka panjang dalam data teks. Selain itu, MLP cenderung mengalami *overfitting* pada data berdimensi tinggi, sehingga membutuhkan teknik regularisasi untuk meningkatkan kemampuan generalisasi (Naskath *et al.*, 2023).

Klasifikasi sentimen sendiri merupakan aspek penting dalam *Natural Language Processing* (NLP) dan pembelajaran mesin, yang bertujuan untuk mendeteksi dan mengategorikan emosi atau sikap yang terkandung dalam data teks. Metode klasifikasi ini dapat diterapkan pada berbagai tingkatan, seperti tingkat dokumen, kalimat, atau sub-kalimat, dan memiliki aplikasi yang luas, termasuk analisis sentimen media sosial, ulasan produk, dan survei opini publik (The NYU School of Professional Studies, 2024).

Meskipun MLP sering digunakan untuk tugas-tugas klasifikasi, model ini kurang kompetitif dibandingkan model pembelajaran mendalam yang lebih canggih, seperti *Transformer*, CNN, dan LSTM (Munandar dkk., 2021)(Khotimah *et al.*, 2023). Hal ini menunjukkan bahwa MLP memiliki keterbatasan dalam menangani kompleksitas data tekstual yang semakin meningkat (Dang

et al.,2020)(Tan *et al.*,2022). Akan tetapi dalam beberapa kasus, MLP membutuhkan waktu komputasi lebih cepat tetapi menghasilkan akurasi yang lebih rendah dengan perbedaan 19 detik dibandingkan model lain yang memanfaatkan *word embedding*, seperti yang ditunjukkan oleh (Khotimah *et al.*,2020). Mereka memperlihatkan bahwa MLP mampu menyelesaikan satu iterasi hanya dalam waktu kurang dari 5 detik. Hal ini menunjukkan keunggulan MLP yaitu lebih sedikit memerlukan *resources* komputasi. Pada beberapa kasus ketika *resources* komputasi rendah, maka MLP bisa menjadi salah satu solusi. Oleh karena itu, diperlukan pendekatan yang dapat mengintegrasikan keunggulan MLP dengan kemampuan ekstraksi fitur dari model jaringan saraf canggih untuk meningkatkan performanya.

Dang *et al.*,(2020) melakukan penelitian yang membandingkan kinerja *Deep Neural Networks* (DNN), *Convolutional Neural Network* (CNN), dan *Recurrent Neural Networks* (RNN) pada delapan kumpulan data analisis sentimen dengan ukuran data berkisar dari ribuan hingga jutaan data. Mereka mengevaluasi dan membandingkan akurasi serta waktu pemrosesan setiap model. Eksperimen menunjukkan bahwa CNN memberikan keseimbangan terbaik antara waktu pemrosesan dan akurasi di antara ketiga model pembelajaran mendalam yang digunakan untuk analisis sentimen. Sebaliknya, meskipun RNN mencapai akurasi tertinggi dengan *word embedding*, waktu yang diperlukan sepuluh kali lebih lama dibandingkan CNN. RNN berkinerja buruk dengan TF-IDF, yang menghasilkan hasil yang kurang efektif, sedangkan DNN menunjukkan waktu pemrosesan dan hasil yang rata-rata. Secara umum, *word embedding* mengungguli TF-IDF dalam mengonversi teks menjadi vektor numerik, terutama pada RNN. Selain itu, akurasi klasifikasi sentimen lebih tinggi pada kumpulan data *tweets* dan ulasan film IMDB, sementara kumpulan data bertema spesifik, seperti *Tweets Airline*, menunjukkan kinerja yang lebih baik dibandingkan kumpulan data bertema umum.

Tan *et al.*,(2022) juga mengeksplorasi pembelajaran mendalam untuk analisis sentimen, mengimplementasikan sebelas metode berbeda, termasuk CNN-LSTM, CNN-BiLSTM, dan pendekatan baru mereka, RoBERTa-LSTM. Pada tiga kumpulan data, model yang mereka usulkan menunjukkan kinerja tertinggi, dengan peningkatan yang signifikan dari 79.10% menjadi 89,70% dibandingkan model dasar, RoBERTa. Vektor kata dari RoBERTa secara khusus berkontribusi pada kemampuan LSTM dalam menangkap informasi tem-

poral secara efektif. Dalam studi lain, (Khotimah *et al.*,2020) berusaha mendeteksi kejadian demam berdarah menggunakan berita daring dengan tiga model pembelajaran mendalam. Mereka menemukan bahwa dibandingkan dengan MLP, CNN membutuhkan hampir dua kali waktu komputasi, tetapi mencapai kinerja yang lebih baik dengan peningkatan 3-4%.

Mohbey *et al.*,(2024) melakukan penelitian yang bertujuan menganalisis sentimen publik terhadap wabah *Monkeypox* dengan menggunakan data dari Twitter. Penelitian ini menerapkan arsitektur hibrida CNN-LSTM, di mana CNN berfungsi untuk mengekstraksi fitur lokal, sedangkan LSTM menangani ketergantungan jangka panjang pada teks. Dataset yang digunakan terdiri dari sekitar 555 ribu *tweet* berbahasa Inggris terkait wabah *Monkeypox*. Hasil penelitian menunjukkan bahwa penggunaan model hibrida mampu meningkatkan akurasi sebesar 4-5% dibandingkan dengan model dasar. Selain itu, model hibrida ini juga mengungguli model lainnya dalam metrik seperti F1-score dan AUC yang berhasil mencapai nilai sebesar 91,24% dan 92%.

Penelitian (Kour and Gupta,2022) ini membahas pendekatan pembelajaran mendalam hibrida untuk memprediksi depresi melalui analisis *tweet* di Twitter. Dengan latar belakang depresi sebagai salah satu masalah kesehatan mental terbesar secara global, penelitian ini memanfaatkan data dari media sosial untuk mengidentifikasi pengguna yang menunjukkan tanda-tanda depresi berdasarkan teks dalam *tweet* mereka. Model hibrida yang diusulkan menggabungkan *Convolutional Neural Network* (CNN) dan *Bi-directional Long Short-Term Memory* (BiLSTM). CNN digunakan untuk mengekstraksi fitur spasial dari data teks, sementara BiLSTM menangani konteks temporal dalam data urutan. Dalam eksperimen, model CNN-BiLSTM dibandingkan dengan model *Recurrent Neural Network* (RNN) dan CNN dasar serta model-model dari penelitian terdahulu. CNN-BiLSTM sendiri menunjukkan kinerja yang lebih baik, dengan akurasi mencapai 94,28% dengan adanya peningkatan sebesar 3% pada dataset *benchmark*. Evaluasi dilakukan menggunakan berbagai metrik, termasuk *precision*, *recall*, *F1-score*, dan AUC

Berbagai penelitian terdahulu menunjukkan bahwa penerapan arsitektur *deep learning* seperti CNN, RNN, LSTM, hingga model hibrida berbasis *neural network*, mampu meningkatkan kinerja dalam analisis sentimen dan de-

teks pola teks. Model hibrida berhasil menggabungkan keunggulan masing-masing komponen, dengan peningkatan akurasi hingga 4–10% dibandingkan model dasar pada berbagai kumpulan data, serta menunjukkan nilai evaluasi metrik seperti *F1-score* dan AUC yang tinggi. Pendekatan ini terbukti efektif dalam menangkap pola spasial dan temporal dari data teks, sehingga memberikan potensi besar untuk meningkatkan kinerja pada berbagai aplikasi analisis berbasis teks.

Berdasarkan latar belakang dan penelitian-penelitian terdahulu yang telah dijelaskan, penelitian ini bertujuan untuk mengatasi keterbatasan MLP dalam tugas klasifikasi sentimen dengan mengusulkan pendekatan hibrida yang menggabungkan keunggulan ekstraksi fitur dari jaringan saraf canggih dengan kemampuan generalisasi MLP. Dengan mengintegrasikan fitur yang lebih representatif agar MLP mampu menangkap pola-pola kompleks dalam data teks. Pendekatan ini terinspirasi dari penelitian sebelumnya yang menunjukkan bahwa ekstraksi fitur pada tingkat rendah dapat meningkatkan kinerja model klasifikasi (Socher *et al.*,2013). Kami berhipotesis bahwa dengan mengintegrasikan kekuatan representasi fitur dari jaringan saraf, seperti model BERT yang diusulkan oleh (Devlin *et al.*,2018), dengan kemampuan generalisasi MLP, model hibrida yang dihasilkan akan mampu menangkap informasi yang lebih kaya dari teks dan mencapai akurasi klasifikasi yang lebih tinggi.

1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan metode fitur ekstraksi berbasis *neural network* yang dapat meningkatkan performa *multilayer perceptron* (MLP) dalam tugas klasifikasi sentimen.

1.3 Manfaat Penelitian

Manfaat yang diharapkan dari hasil penelitian ini sebagai berikut:

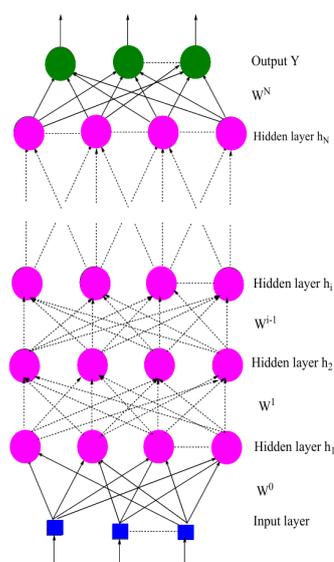
- 1) Menambah pengetahuan dan wawasan mengenai model *multilayer perceptron* (MLP) dalam melakukan tugas klasifikasi sentimen
- 2) Mengetahui perubahan performa yang diberikan model *neural network* kepada *multilayer perceptron* (MLP)
- 3) Mengetahui model *neural network* yang lebih meningkatkan performa *multilayer perceptron* (MLP)

BAB II

TINJAUAN PUSTAKA

2.1 *MultiLayer Perceptron*

MultiLayer Perceptron (MLP) adalah model *deep learning* yang berbasis pada konsep jaringan saraf tiruan. Sesuai dengan namanya, arsitektur model ini terdiri dari beberapa lapisan yang tersusun secara hierarkis dan saling terhubung, yaitu lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*), seperti yang ditunjukkan pada Gambar 2.1. Pada setiap lapisan ini terdapat perseptron yang disebut *neuron*, yang berfungsi untuk memproses data dari *neuron* pada lapisan sebelumnya menggunakan bobot, bias, dan fungsi aktivasi. Jumlah *neuron* pada *input layer* sama dengan jumlah pengukuran untuk masalah pola, sedangkan jumlah *neuron* pada *output layer* sama dengan jumlah kelas (Ramchoun *et al.*,2016).



Gambar 2.1 Arsitektur MLP
Sumber: (Ramchoun *et al.*,2016)

Proses MLP secara berurutan dijelaskan oleh (Ramchoun *et al.*,2016) dengan adanya penambahan bias proses sebagai berikut. Asumsikan data yang digunakan dengan $X = (x_0, x_1, \dots, x_{n_0})$ dengan *input layer* sejumlah n_0 dan aktivasi fungsi sigmoid dengan persamaan.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.1.1)$$

kemudian anggap (h_1, h_2, \dots, h_n) sekumpulan *hidden layer*. Asumsikan n_i adalah jumlah *neuron* di setiap *hidden layer* h_i , sehingga hasil *hidden layer* pertama memiliki persamaan

$$h_i^j = f \left(\sum_{k=1}^{n_{i-1}} W_{k,j}^0 X_k + b \right) \quad j = 1, \dots, n_i \quad (2.1.2)$$

Di mana:

- W^0 adalah bobot data antara *input layer* dan *hidden layer* pertama,
- k adalah *neuron* di *input layer*,
- j adalah *neuron* di *hidden layer* pertama,
- X adalah data yang digunakan,
- b adalah bias.

Karena (2.1.2) adalah rumus untuk mendapatkan hasil *hidden layer* pertama, maka untuk hasil h_i^j yang merupakan *output* di *hidden layer* selanjutnya dapat menggunakan persamaan.

$$h_i^j = f \left(\sum_{k=1}^{n_{i-1}} W_{k,j}^{i-1} h_{i-1}^k + b \right) \quad i = 2, \dots, N \text{ and } j = 1, \dots, n_i \quad (2.1.3)$$

Di mana:

- k adalah *neuron* di *hidden layer* i ,
- j adalah *neuron* di *hidden layer* $i + 1$,
- n_i adalah banyaknya *neuron* di *hidden layer* ke- i .

Sehingga rumus untuk mendapatkan hasil dari *output layer* adalah

$$y_i = f \left(\sum_{k=1}^{n_N} W_{k,j}^N h_N^k + b \right)$$

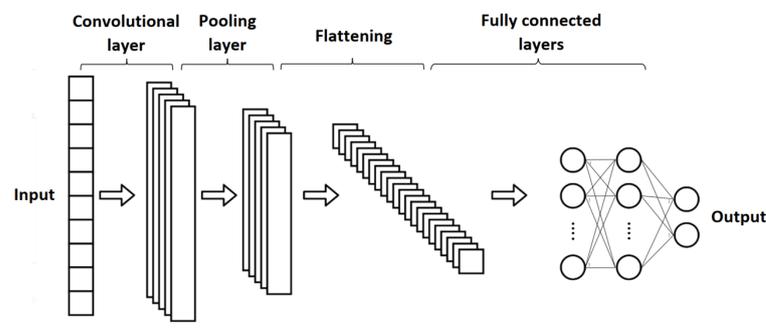
$$Y = (y_1, \dots, y_j, \dots, y_{N+1}) = F(W, X) \quad (2.1.4)$$

Di mana:

- $W_{k,j}^N$ adalah bobot data antara k dan j ,
- k adalah *neuron* di *hidden layer* ke- N ,
- j adalah *neuron* di *output layer*,
- Y adalah vektor dari *output layer*,
- F adalah fungsi yang digunakan,
- W adalah matriks bobot.

2.2 Convolutional Neural Network

Convolutional neural network (CNN) adalah jenis jaringan saraf tiruan yang dirancang untuk menganalisa data yang memiliki struktur *grid*, seperti gambar atau data teks yang direpresentasikan sebagai vektor *embedding*. Berbeda dengan MLP, CNN menggunakan lapisan konvolusi (*convolutional layer*) untuk mengekstraksi fitur utama dari data input melalui operasi lokal menggunakan filter(kernel), sehingga mengurangi jumlah parameter yang diproses. Setelah itu, lapisan pooling (*pooling layer*) digunakan untuk mereduksi dimensi data, sementara lapisan yang terhubung sepenuhnya (*fully connected layer*) menghasilkan prediksi berdasarkan fitur yang telah diekstraksi.



Gambar 2.2 Arsitektur CNN

Sumber: (Serghini *et al.*, 2023)

Karena arsitektur CNN yang ditunjukkan pada Gambar 2.2 , CNN merupakan salah satu metode *deep learning* yang efektif untuk klasifikasi sentimen karena dapat memanfaatkan representasi teks untuk mengekstraksi fitur (Salur and aydin,2020). Dengan kemampuannya dalam mengolah teks secara efisien, cara kerja matematis dari setiap lapisan CNN juga telah dijelaskan oleh (Wu,2017). Pada *convolutional layer*, fitur diekstraksi dari gambar atau teks menggunakan berbagai filter. Proses intermediat dilakukan di antara *convolutional layer* dan *pooling layer* untuk menjadikan fitur bersifat *non-linear*, dengan bantuan fungsi aktivasi *Rectified Linear Unit* (ReLU). Adapun persamaan prosedur *convolutional layer* fungsi aktivasi ReLU adalah

$$y_{i^{l+1},j^{l+1},d} = \sum_{i=0}^H \sum_{j=0}^W \sum_{d^l=0}^{D^l} f_{i,j,d^l,d} \times x_{i^{l+1}+i,j^{l+1}+j,d^l}^l \quad (2.2.5)$$

Di mana:

- H adalah jumlah baris,
- W adalah jumlah kolom,
- D adalah jumlah kedalaman,
- i adalah indeks baris,
- j adalah indeks kolom,
- d adalah indkes kedalaman.

(2.2.5) dilakukan berulang kali untuk semua $0 \leq d \leq D = D^{l+1}$, dan untuk semua lokasi spasial (i^{l+1}, j^{l+1}) yang memenuhi $0 \leq i^{l+1} < H^l - H + 1 = H^{l+1}, 0 \leq j^{l+1} < W^l - W + 1 = W^{l+1}$. di persamaan ini, $x_{i^{l+1}+i,j^{l+1}+j,d^l}^l$ merujuk kepada elemen yang diindeks oleh *triplet* $(i^{l+1} + i, j^{l+1} + j, d^l)$.

Setelah proses pada *convolutional layer* selesai, langkah berikutnya adalah pemrosesan menggunakan *pooling layer*. Pada *pooling layer*, dimensi dari *feature map* dikurangi untuk mengurangi beban komputasi pada lapisan berikutnya serta menampilkan fitur pada gambar atau teks secara lebih efektif (Salur and aydin,2020). *Pooling layer* dibagi menjadi dua jenis, yaitu *max pooling layer* dan *average pooling layer*. Pada *max pooling*, operator *pooling*

memetakan suatu *subregion* ke nilai maksimum dalam area tersebut, sedangkan pada *average pooling* operator memetakan *subregion* ke nilai rata-ratanya (Wu,2017). Persamaan untuk *max pooling* dan *average pooling* adalah sebagai berikut

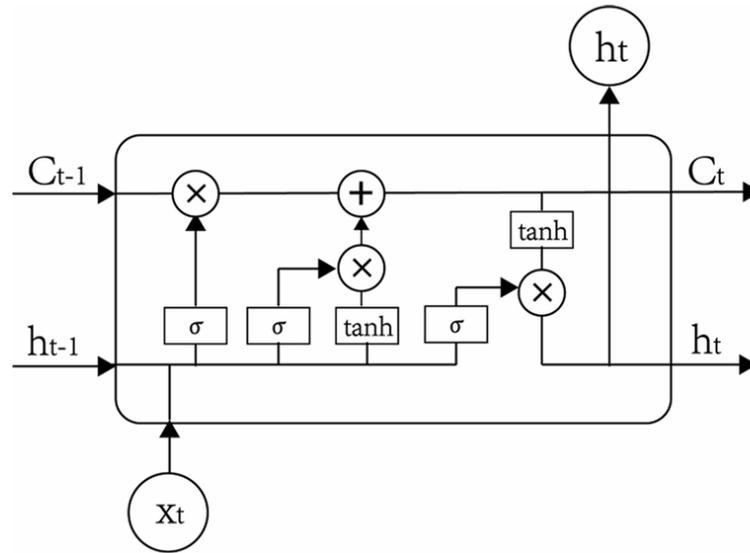
$$\text{max: } y_{i+1,j+1,d}^{l+1} = \max_{0 \leq i < H, 0 \leq j < W} x_{i+1 \times H+i, j+1 \times W+j, d}^l \quad (2.2.6)$$

$$\text{average: } y_{i+1,j+1,d}^{l+1} = \frac{1}{HW} \sum_{0 \leq i < H, 0 \leq j < W} x_{i+1 \times H+i, j+1 \times W+j, d}^l \quad (2.2.7)$$

Lapisan terakhir dari *Convolutional neural network* berbentuk jaringan *neural* buatan *fully connected* klasik. Pada lapisan ini, struktur *fully connected* antara *neuron* buatan merepresentasikan fitur dari gambar atau teks dan label kelas target.

2.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) merupakan model *deep learning* yang menggunakan salah satu bentuk RNN yang paling umum, dirancang untuk mengatasi masalah ketergantungan jangka panjang dan cocok untuk memproses serta memprediksi deret waktu (Qiu *et al.*,2020). Model LSTM terdiri dari serangkaian sel memori yang menggantikan *neuron* pada *hidden layer* dalam RNN. Model ini menyaring informasi melalui struktur gerbang untuk mempertahankan dan memperbarui keadaan sel memori. Struktur gerbang ini mencakup *input gate*, *forget gate*, dan *output gate*. Setiap sel memori memiliki tiga lapisan sigmoid dan satu lapisan tanh, seperti yang ditunjukkan Gambar 2.3



Gambar 2.3 Arsitektur LSTM
Sumber: (Qiu *et al.*,2020)

Lapisan sigmoid dan tanh adalah lapisan yang memproses informasi dengan menggunakan prinsip fungsi aktivasi sigmoid dan fungsi aktivasi tanh. Persamaan untuk fungsi tanh adalah sebagai berikut.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3.8)$$

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} \quad (2.3.9)$$

Penjelasan mengenai proses yang matematis yang dilakukan oleh LSTM telah dijelaskan oleh (Graves *et al.*,2013), dimana diberikan *input* barisan $x = (x_1, \dots, x_T)$, sebuah RNN standar menghitung barisan vektor tersembunyi $h = (h_1, \dots, h_T)$ dan *output* barisan vektor $y = (y_1, \dots, y_T)$ dengan mengiterasi persamaan berikut dari $t = 1$ sampai T

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (2.3.10)$$

$$y_t = W_{hy}h_t + b_y \quad (2.3.11)$$

Di mana:

- W adalah matriks bobot

- \mathcal{H} adalah fungsi *hidden layer*
- b adalah bias

\mathcal{H} diimplementasikan menggunakan persamaan berikut

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2.3.12)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2.3.13)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (2.3.14)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (2.3.15)$$

$$h_t = o_t \tanh(c_t) \quad (2.3.16)$$

Di mana:

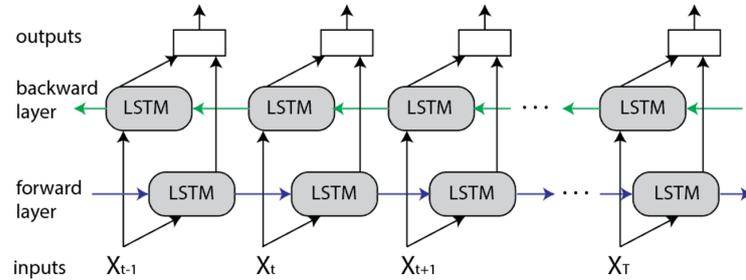
- σ adalah fungsi sigmoid,
- i_t adalah *input gate*,
- f_t adalah *forget gate*,
- o_t adalah *output gate*,
- c_t adalah *cell activation*.

Fungsi utama *forget gate* adalah mencatat seberapa banyak *cell state* C_{t-1} pada waktu sebelumnya yang dipertahankan pada *cell state* C_t pada waktu saat ini. Kemudian *input gate* memiliki dua fungsi, pertama untuk menemukan *cell state* yang harus diperbarui, dan kedua untuk memperbarui informasi yang akan dimasukkan ke dalam keadaan *cell state*. Dan *output gate* menentukan nilai *output* dari unit LSTM.

2.4 Bidirectional Long Short-Term Memory

Bidirectional Long Short-Term Memory (BiLSTM) adalah pengembangan arsitektur jaringan *Long Short-Term Memory* (LSTM) yang dirancang untuk mengatasi keterbatasan LSTM yang hanya memproses data sekuensial dalam satu arah. BiLSTM memanfaatkan dua lapisan LSTM, satu memproses sekuen maju dan satu lagi memproses sekuen mundur. Penggunaan model *bidirectional* ini memungkinkan pemanfaatan konteks masa lalu dan masa

depan, sehingga meningkatkan akurasi prediksi untuk tugas pemodelan sekuens (Schuster and Paliwal,1997)



Gambar 2.4 Arsitektur BiLSTM
Sumber: (Serghini *et al.*,2023)

Pada Gambar 2.4 terlihat arsitektur BiLSTM, karena hal ini lah proses matematis dari BiLSTM sendiri tidak banyak berubah dibandingkan LSTM. (Graves *et al.*,2013) mengatakan *Bidirectional* RNN memproses barisan tersembunyi *forward* \vec{h} , barisan tersembunyi *backward* \overleftarrow{h} dan *output* barisan y dengan memproses *backward layer* dari $t = T$ ke 1, *forward layer* dari $t = 1$ ke T dan kemudian mendapat hasil *output layer*

$$\vec{h}_t = \mathcal{H} \left(W_{x\vec{h}} x_t + W_{\vec{h}\vec{h}} \vec{h}_{t-1} + b_{\vec{h}} \right) \quad (2.4.17)$$

$$\overleftarrow{h}_t = \mathcal{H} \left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}} \right) \quad (2.4.18)$$

$$y_t = W_{\vec{h}y} \vec{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \quad (2.4.19)$$

Di mana:

- \vec{h} adalah proses *hidden layer forward*,
- \overleftarrow{h} adalah proses *hidden layer backward*.

Penggabungan *Bidirectional* RNN dengan LSTM inilah yang menciptakan model *Bidirectional LSTM* yang dapat mengakses konteks jangka panjang dalam kedua arah. Karakteristik unik ini memungkinkan jaringan untuk secara efektif menangkap dan memanfaatkan informasi *input* yang lebih luas. Dengan memproses data dalam dua arah, model ini dapat memahami konteks yang lebih baik, terutama pada data sekuensial seperti teks atau waktu (Serghini *et al.*,2023).

2.5 Word Embedding

Word embedding adalah teknik penting dalam pemrosesan bahasa alami yang mengubah kata menjadi representasi numerik dalam bentuk vektor, sehingga memungkinkan model untuk memahami makna dan hubungan antar kata. Salah satu model yang menerapkan konsep ini adalah Word2Vec, yang dikembangkan oleh (Mikolov *et al.*,2013) untuk merepresentasikan kata dalam bentuk vektor di ruang berdimensi tinggi. Model ini dirancang agar kata-kata dengan makna serupa memiliki representasi vektor yang berdekatan. Word2Vec memiliki arsitektur utama, yaitu *Continuous Bag of Words* (CBOW).

Dalam pendekatan CBOW, model berusaha memprediksi kata target berdasarkan kata-kata di sekitarnya dalam suatu jendela konteks. Probabilitas kata target diberikan oleh:

$$P(w_t | w_{t-m}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+m}) = \frac{\exp(v_{w_t} \cdot v_c)}{\sum_{w \in V} \exp(v_w \cdot v_c)} \quad (2.5.20)$$

Di mana:

- v_{w_t} adalah vektor kata target,
- v_s adalah rata-rata vektor kata-kata dalam konteks,
- V adalah ukuran kosakata.

Fungsi loss yang digunakan dalam CBOW adalah:

$$J = - \sum_t \log P(w_t | \text{context}) \quad (2.5.21)$$

Salah satu tantangan utama dalam pelatihan Word2Vec adalah komputasi *softmax* penuh yang mahal secara komputasi ($\mathcal{O}(V)$). Untuk mengatasi hal ini, (Mikolov *et al.*,2013) mengusulkan *Negative Sampling*, yang menggantikan perhitungan *softmax* dengan pendekatan probabilitas biner:

$$J = \log \sigma(v_{w_t} \cdot v_c) + \sum_{i=1}^k \mathbb{E}_{w_{\text{neg}} \sim P(w)} \log \sigma(-v_{w_{\text{neg}}} \cdot v_c) \quad (2.5.22)$$

Di mana:

- σ adalah fungsi sigmoid,
- w_{neg} adalah sampel kata negatif dari $P(w)$,
- $P(w)$ adalah distribusi probabilitas.

Word2vec memberikan keuntungan yang signifikan dalam pemrosesan bahasa alami dengan menghasilkan representasi vektor kata yang mampu menangkap hubungan semantik dan sintaktik antar kata. Model ini memungkinkan pengolahan data teks yang lebih efisien dan akurat, serta meningkatkan kinerja dalam berbagai tugas seperti klasifikasi teks, analisis sentimen, dan penerjemahan mesin. (Mikolov *et al.*, 2013) menunjukkan bahwa word2vec dapat menghasilkan representasi kata berkualitas tinggi dengan biaya komputasi yang rendah.

2.6 TF-IDF

TF-IDF (*Term Frequency - Inverse Document Frequency*) adalah metode numerik yang digunakan dalam pemrosesan teks untuk mengevaluasi seberapa penting suatu kata dalam sebuah dokumen relatif terhadap kumpulan dokumen (*corpus*). TF-IDF banyak digunakan dalam pencarian informasi dan pengklasifikasian teks.

TF mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Persamaannya diberikan oleh:

$$TF(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (2.6.23)$$

Di mana:

- $f_{t,d}$ adalah jumlah kemunculan kata t dalam dokumen d ,

- $\sum_{t' \in d} f_{t',d}$ adalah total jumlah kata dalam dokumen d .

IDF mengukur seberapa jarang suatu kata muncul di seluruh dokumen dalam korpus. Persamaanya adalah:

$$IDF(t, D) = \log \frac{|D|}{|d \in D : t \in d|} \quad (2.6.24)$$

Di mana:

- $|D|$ adalah jumlah total dokumen dalam korpus,
- $|d \in D : t \in d|$ adalah jumlah dokumen yang mengandung kata t .

Nilai TF-IDF dihitung dengan mengalikan nilai TF dan IDF:

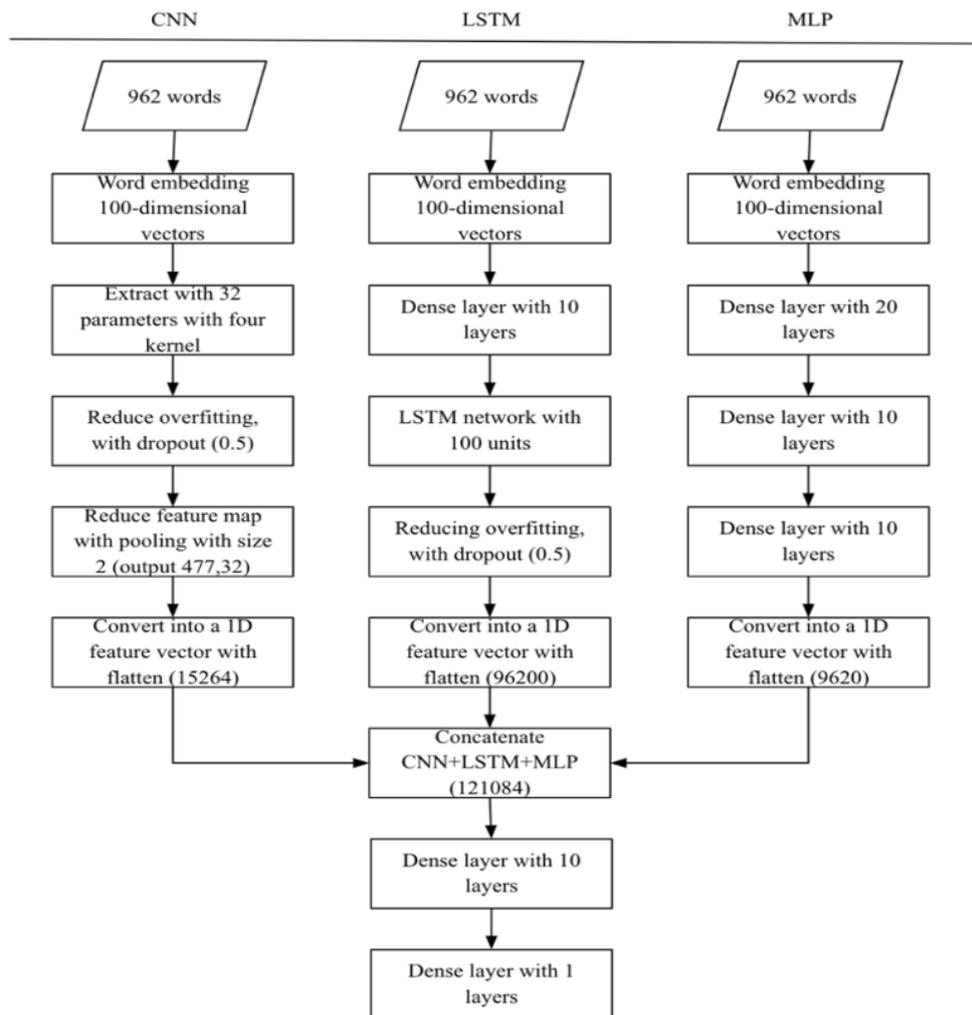
$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (2.6.25)$$

2.7 Optimalisasi Performa Model

Dalam upaya meningkatkan performa MLP, terdapat berbagai metode optimalisasi yang dapat diterapkan. Salah satu metode yang umum digunakan adalah penyesuaian *hyperparameter*, di mana parameter seperti laju pembelajaran, jumlah *neuron*, dan jumlah lapisan disesuaikan untuk menemukan konfigurasi yang paling efektif. Selain itu, teknik regularisasi seperti *Dropout* dan *L2 Regularization* dapat diterapkan untuk mencegah *overfitting*, sehingga model dapat digeneralisasi dengan lebih baik pada data yang tidak terlihat. Penggunaan algoritma optimasi yang lebih canggih, seperti Adam atau RMSprop, juga dapat mempercepat konvergensi dan meningkatkan akurasi model. Metode lain termasuk normalisasi data dan augmentasi data, yang bertujuan untuk meningkatkan kualitas input dan memperluas variasi data pelatihan.

Salah satu pendekatan yang efektif untuk meningkatkan kinerja MLP adalah melalui hibridasi model. Hibridasi model melibatkan penggabungan dua atau

lebih algoritma atau arsitektur model untuk memanfaatkan kelebihan masing-masing. Selain itu, hibridasi model juga memungkinkan penggunaan teknik *ensemble*, di mana beberapa model MLP dilatih secara bersamaan dan hasilnya digabungkan untuk menghasilkan prediksi akhir. Pendekatan ini tidak hanya meningkatkan akurasi tetapi juga mengurangi risiko *overfitting* yang sering terjadi pada model tunggal.



Gambar 2.5 Diagram Model Hibridasi Oleh (Munandar dkk.,2021

Sumber: (Munandar dkk.,2021

Teknik *ensemble* pernah diterapkan oleh (Munandar dkk.,2021 untuk meningkatkan performa MLP dalam tugas klasifikasi sentimen. Dalam penelitian tersebut, teknik *ensemble* berhasil meningkatkan akurasi model hing-

ga mendekati 100% dengan mengombinasikan MLP, CNN, dan LSTM. Diagram arsitektur model hibridasi yang digunakan dalam penelitian (Munandar dkk.,2021) dapat dilihat pada Gambar 2.5

Penelitian ini akan menerapkan metode hibridasi model yang berbeda, di mana data akan terlebih dahulu diproses oleh *neural network* sebelum diteruskan ke MLP. Proses ini bertujuan untuk memungkinkan *neural network* melakukan ekstraksi fitur, sehingga dapat mempermudah MLP dalam memproses data secara lebih efisien. Pendekatan ini diharapkan dapat meningkatkan akurasi dan efisiensi model dalam menyelesaikan tugas yang kompleks.

2.8 Metrik Evaluasi Performa

Dalam penelitian ini, kinerja model evaluasi diukur menggunakan beberapa metrik utama, yaitu *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan ROC-AUC. Penjelasan dari masing-masing metrik tersebut adalah sebagai berikut:

2.8.1 Accuracy

Accuracy (akurasi) adalah proporsi prediksi yang benar terhadap keseluruhan data yang diuji. Persamaan akurasi berdasarkan (Powers,2020) adalah:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.8.26)$$

Di mana:

- TP: *True Positive* (prediksi positif yang benar)
- TN: *True Negative* (prediksi negatif yang benar)
- FP: *False Positive* (prediksi positif yang salah)
- FN: *False Negative* (prediksi negatif yang salah)

2.8.2 Precision

Precision (Presisi) adalah proporsi prediksi positif yang benar terhadap semua prediksi positif. Menurut (Powers,2020) persamaannya adalah:

$$Precision = \frac{TP}{TP + FP} \quad (2.8.27)$$

2.8.3 Recall

Recall adalah proporsi kasus positif sebenarnya yang berhasil dideteksi sebagai positif. Berdasarkan (Powers,2020) persamaan *recall* adalah:

$$Recall = \frac{TP}{TP + FN} \quad (2.8.28)$$

2.8.4 F1-Score

F1-score adalah rata-rata harmonis antara *precision* dan *recall*, yang dirancang untuk memberikan evaluasi yang lebih seimbang terhadap performa model, terutama dalam kondisi di mana terdapat ketimpangan antara jumlah kelas positif dan negatif. menurut (Sasaki,2007) persamaannya adalah:

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.8.29)$$

2.8.5 ROC-AUC

Receiver Operating Characteristic (ROC) AUC merupakan metrik evaluasi yang digunakan untuk menilai performa model klasifikasi dengan memvisualisasikan hubungan antara *true positive rate* (TPR) dan *false positive rate*

(FPR). Nilai AUC (*Area Under the Curve*) mencerminkan sejauh mana model mampu membedakan antara kelas positif dan negatif. Menurut (Fawcett, 2006) persamaan untuk TPR dan FPR sendiri adalah:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.8.30)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (2.8.31)$$

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan saat kegiatan magang penelitian MBKM pada semester genap pada tahun pelajaran 2023/2024 di PRSDI BRIN KST Samadikun Bandung dan semester ganjil pada tahun pelajaran 2024/2025 Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung

3.2 Perangkat dan Library Penelitian

Penelitian ini menggunakan perangkat berupa laptop asus TUF *gaming* F15 dengan spesifikasi intel core i5-11400H processor 2.7 GHz, memori 8GB DDR4, *storage* 512GB SSD dan *graphic* Nvidia Geforce RTX3050 4GB GDDR6. Dengan menggunakan *software* Googlecolab dan penggunaan *library* yang digunakan adalah numpy dengan versi 2.0.2, pandas versi 2.2.2, gensim versi 4.3.3, tensorflow versi 2.18.0, nltk versi 3.9.1, scikit-learn 1.6.1, keras tuner versi 1.4.7, matplotlib versi 3.10.0, dan seaborn dengan versi 0.13.2.

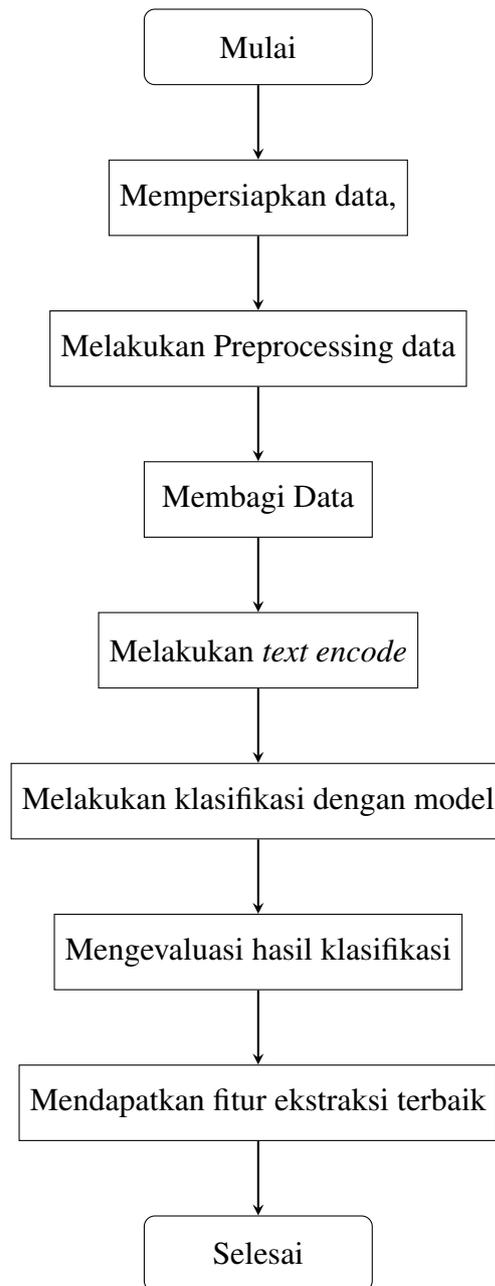
3.3 Metode Penelitian

Langkah-langkah yang dilakukan pada penelitian ini sebagai berikut:

1. Mempersiapkan data yang akan digunakan
2. Melakukan *preprocessing* pada data

3. Membagi data menjadi *training data*, *testing data*, dan *validation data*
4. Melakukan *text encode* dengan menggunakan metode TF-IDF dan *word embedding*
5. Melakukan klasifikasi dengan model MLP dengan tambahan *neural net* sebagai fitur ekstraksi
6. Mengevaluasi hasil klasifikasi model
7. Menganalisa hasil

Berikut ini diberikan *flowchart* untuk memperlihatkan secara jelas tahapan penelitian yang lebih sistematis pada Gambar 3.1 :



Gambar 3.1 *flowchart* alur proses penelitian

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini mengevaluasi kinerja model MLP yang dikombinasikan dengan model CNN, LSTM, dan Bi-LSTM untuk ekstraksi fitur, serta membandingkannya dengan model MLP mandiri. TF-IDF dan WE digunakan untuk mengubah teks menjadi representasi numerik, dan data dibagi menjadi himpunan pelatihan dan pengujian menggunakan metode SKCV. Hasil eksperimen menunjukkan bahwa peningkatan model MLP dengan lapisan ekstraksi fitur berbasis jaringan saraf secara signifikan meningkatkan kinerja, yang diukur menggunakan ROC-AUC.

Di antara konfigurasi yang diuji, CNN muncul sebagai lapisan ekstraksi fitur yang paling unggul untuk MLP. Secara khusus, model WE + CNN-MLP keseluruhan memiliki nilai 89% pada metrik performa *accuracy*, *precision*, *recall*, dan *f1-score* dengan nilai metrik ROC-AUC mencapai 92.89%. Sama seperti model yang menggunakan WE, model TF-IDF + CNN-MLP mendapatkan keseluruhan metrik performa mencapai 89% dan metrik ROC-AUC mencapai 92,49%.

Dengan akurasi sebesar 89% yang diperoleh model, baik saat menggunakan *Word Embedding* maupun TF-IDF, dari total 16,796 data, model berhasil mengklasifikasikan sekitar 14,948 data dengan benar. Sementara itu, nilai ROC-AUC sebesar 92% menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam membedakan berbagai kelas sentimen. Hasil ini menunjukkan bahwa penambahan lapisan fitur ekstraksi berhasil meningkatkan kinerja model berdasarkan evaluasi metrik performa. Selain itu, penggunaan *Word Embedding* sebagai pengkode teks menghasilkan hasil yang relatif lebih baik dibandingkan dengan TF-IDF.

5.2 Saran

Penelitian ini berkontribusi dalam pengembangan model klasifikasi teks menggunakan deep learning, khususnya *Multi-Layer Perceptron* (MLP), untuk mencapai akurasi, efektivitas, dan efisiensi yang lebih tinggi dalam menangani dataset yang besar dan kompleks. Keterbatasan dalam penelitian ini adalah ketergantungan pada satu dataset, sedangkan berbagai jenis data dapat memengaruhi hasil klasifikasi. Penelitian selanjutnya dapat mengembangkan hasil ini dengan mengeksplorasi berbagai model deep learning sebagai arsitektur dasar, meneliti alternatif jaringan saraf untuk ekstraksi fitur, serta menggunakan cakupan dataset yang lebih luas. Selain itu, penelitian selanjutnya juga bisa dilakukan dengan restrukturisasi data, yaitu mengubah atau menyusun ulang format data agar lebih optimal untuk proses pelatihan model.

DAFTAR PUSTAKA

- Carrington, A. M., Manuel, D. G., Fieguth, P. W., Ramsay, T., Osmani, V., Wernly, B., Bennett, C., Hawken, S., Magwood, O., Sheikh, Y., McInnes, M., Holzinger, A. (2023). Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation. *IEEE transactions on pattern analysis and machine intelligence*, 45(1), 329–341. <https://doi.org/10.1109/TPAMI.2022.3145392>
- Dang, N. C., Moreno-García, M. N., De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3), 483.
- Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Heitmann, M., Siebert, C., Hartmann, J., Schamp, C. (2020). More than a Feeling: Benchmarks for Sentiment Analysis Accuracy. *Communication Computational Methods eJournal*.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861-874.
- Graves, A., Jaitly, N., Mohamed, A. R. (2013, December). Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding* (pp. 273-278). IEEE.
- Khotimah, P. H., Rozie, A. F., Nugraheni, E., Arisal, A., Suwarningsih, W., Purwarianti, A. (2020, November). Deep learning for dengue fever event detection using online news. In *2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)* (pp. 261-266). IEEE.
- Khotimah, P. H., Arisal, A., Rozie, A. F., Nugraheni, E., Riswantini, D., Suwarningsih, W., ... Purwarianti, A. (2023). Monitoring Indonesian online news for COVID-19 event detection using deep learning. *International Journal of Electrical Computer Engineering* (2088-8708), 13(1).

- Kour, H., Gupta, M. K. (2022). An hybrid deep learning approach for depression prediction from user tweets using feature-rich CNN and bi-directional LSTM. *Multimedia Tools and Applications*, 81(17), 23649-23685.
- Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mohbey, K. K., Meena, G., Kumar, S., Lokesh, K. (2024). A CNN-LSTM-based hybrid deep learning approach for sentiment analysis on Monkeypox tweets. *New Generation Computing*, 42(1), 89-107.
- Munandar, D., Rozie, A. F., Arisal, A. (2021). A multi domains short message sentiment classification using hybrid neural network architecture. *Bulletin of Electrical Engineering and Informatics*, 10(4), 2181-2191.
- Muschelli J. (2020). ROC and AUC with a Binary Predictor: a Potentially Misleading Metric. *Journal of classification*, 37(3), 696–708. <https://doi.org/10.1007/s00357-019-09345-1>.
- Naskath, J., Sivakamasundari, G., Begum, A. A. S. (2023). A study on different deep learning algorithms used in deep neural nets: MLP SOM and DBN. *Wireless personal communications*, 128(4), 2913-2936.
- Ningsih, F. S. S., Khotimah, P. H., Arisal, A., Rozie, A. F., Munandar, D., Riswantini, D., Nugraheni, E., Suwarningsih, W., and Kurniasari, D. (2022). Synonym-based Text Generation in Restructuring Imbalanced Dataset for Deep Learning Models. *2022 5th International Conference on Networking, Information Systems and Security: Envisage Intelligent Systems in 5g//6G-Based Interconnected Digital Worlds (NISS)*, 1–6. <https://doi.org/10.1109/NISS55057.2022.10085156>
- Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.
- Qiu, J., Wang, B., Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS one*, 15(1), e0227222.
- Ramchoun, H., Ghanou, Y., Ettaouil, M., Janati Idrissi, M. A. (2016). Multi-layer perceptron: Architecture optimization and training.

- Salur, M. U., Aydin, I. (2020). A novel hybrid deep learning model for sentiment classification. *IEEE Access*, 8, 58080-58093.
- Sasaki, Yutaka. (2007). The truth of the F-measure. Teach Tutor Mater.
- Serghini, O., Semlali, H., Maali, A., Ghammaz, A., Serrano, S. (2023). 1-D Convolutional Neural Network-Based Models for Cooperative Spectrum Sensing. *Future Internet*, 16(1), 1
- Schuster, M., Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11), 2673-2681.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. *In Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631-1642).
- Sokolova, M., Japkowicz, N., Szpakowicz, S. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In: Sattar, A., Kang, Bh. (eds) *AI 2006: Advances in Artificial Intelligence*. AI 2006. Lecture Notes in Computer Science(), vol 4304. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11941439_114
- Tan, K. L., Lee, C. P., Anbananthen, K. S. M., Lim, K. M. (2022). RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network. *IEEE Access*, 10, 21517-21525.
- The NYU School of Professional Studies. (2024). *What is Sentiment Classification?*. dari <https://www.aimasterclass.com/glossary/sentiment-classification>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-December(Nips), 5999–6009.
- Wankhade, M., Rao, A. C. S., Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7), 5731-5780.
- Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23), 495.

Zhai, Z., Nguyen, D.Q., Verspoor, K.M. (2018). Comparing CNN and LSTM character-level embeddings in BiLSTM-CRF models for chemical and disease named entity recognition. *ArXiv*, abs/1808.08450.

Zhou, B., Skiena, S.S. (2023). Does it pay to optimize AUC? *ArXiv*, abs/2306.01528.