

TESIS

**IMPLEMENTASI *PARALLEL ENSEMBLE LEARNING*
NEURAL NETWORK PADA SENTIMENT ANALISIS TEKS
BAHASA INGGRIS**



disusun oleh :

Nama : AGUNG NUGROHO
NIM : 23.55.1440
Konsentrasi : Big Data & Predictive Analytics

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2026**

TESIS

**IMPLEMENTASI PARALLEL ENSEMBLE LEARNING
NEURAL NETWORK PADA SENTIMENT ANALISIS TEKS
BAHASA INGGRIS**

***IMPLEMENTATION OF PARALLEL ENSEMBLE LEARNING
NEURAL NETWORK IN ENGLISH TEXT SENTIMENT
ANALYSIS***

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Pascasarjana
Program Studi S2 PJJ Informatika



Disusun oleh:

Nama : AGUNG NUGROHO
NIM : 23.55.1440
Konsentrasi : Big Data & Predictive Analytics

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

HALAMAN PERSETUJUAN

**IMPLEMENTASI *PARALLEL ENSEMBLE LEARNING NEURAL NETWORK*
PADA SENTIMENT ANALISIS TEKS BAHASA INGGRIS**

**IMPLEMENTATION OF *PARALLEL ENSEMBLE LEARNING NEURAL
NETWORK IN ENGLISH TEXT SENTIMENT ANALYSIS***

yang disusun dan diajukan oleh

Agung Nugroho

23.55.1440

Telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 02 Januari 2026

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

Dosen Pembimbing



Prof. Arief Setyanto, S.Si, M.T., Ph.D.

NIK. 190302036

HALAMAN PENGESAHAN

**IMPLEMENTASI *PARALLEL ENSEMBLE LEARNING NEURAL NETWORK*
PADA SENTIMENT ANALISIS TEKS BAHASA INGGRIS**

***IMPLEMENTATION OF PARALLEL ENSEMBLE LEARNING NEURAL
NETWORK IN ENGLISH TEXT SENTIMENT ANALYSIS***

yang disusun dan diajukan oleh

Agung Nugroho

23.55.1440

Telah dipertahankan di depan Dewan Penguji
pada tanggal 02 Januari 2026

Susunan Dewan Penguji

Nama Penguji

Prof. Arief Setyanto, S.Si, M.T., Ph.D.
NIK. 190302036

Tandatangan



I Made Artha Agastva, S.T., M.Eng., Ph.D.
NIK. 190302352



Dr. Andi Sunyoto, S.Kom., M.Kom.
NIK. 190302052



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer.
Tanggal 02 Januari 2026

DEKAN FAKULTAS ILMU KOMPUTER



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

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Agung Nugroho
NIM : 23.55.1440

Menyatakan bahwa Tesis dengan judul berikut:

Implementasi *Parallel Ensemble Learning Neural Network* Pada Sentiment Analisis Teks Bahasa Inggris

Dosen Pembimbing Utama : Prof. Ariel Setyanto, S.Si, M.T., Ph.D.

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

Yogyakarta, 02 Januari 2026

Yang Menyatakan,



Agung Nugroho

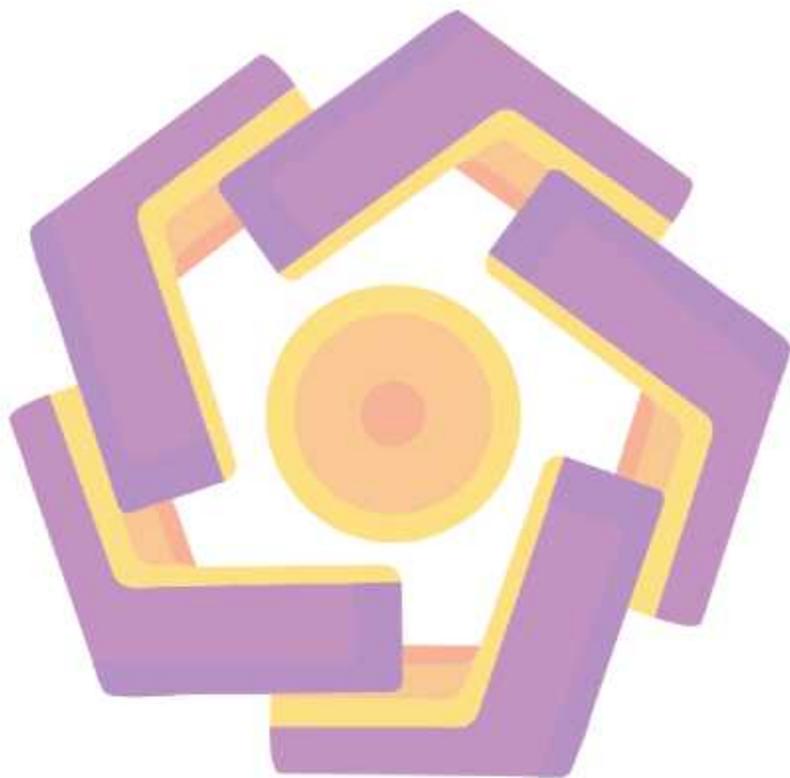
HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur skripsi ini saya persembahkan untuk orang-orang yang sangat berarti di dalam kehidupan saya, teruntuk :

1. Istri dan anak-anak saya yang telah mendukung dan mendoakan segala Upaya dalam menyelesaikan studi Jenjang Magister.
2. Bapak – Ibu yang senantiasa mendukung dengan motivasi dan doanya.
3. Bapak dan Ibu Dosen PJJ S2 Informatika Universitas Amikom Yogyakarta. Terkhusus Bapak Prof. Arief Setyanto S.Si., M.T., Ph.D. atas arahan dan bimbingan Tesis.
4. Ibu Prof. Dr Kusrini, M.Kom dan Bapak I Made Artha Agastya, S.T, M.Eng., P.hD. atas arahan dan bimbingan sehingga penulis mempunyai satu karya tulis Jurnal Ilmiah bereputasi.
5. Rekan-rekan Angkatan 9 PJJ S2 Informatika Amikom atas sharing ilmu, dukungan, dan Kerjasama kelompok dalam menjalani proses perkuliahan.
6. Rekan-rekan kerja di Bagian Perencanaan PLN UP3 Banyuwangi atas saran dan diskusi dalam penulisan Tesis.
7. Seluruh Pihak yang telah mendukung proses belajar jenjang Magister.

HALAMAN MOTTO

“Tetaplah semangat menggapai mimpi sampai kapanpun”



KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, saya dapat menyelesaikan penelitian ini. Penelitian yang berjudul “*Implementasi Parallel Ensemble Learning Neural Network Pada Sentiment Analisis Teks Bahasa Inggris*” disusun sebagai salah satu syarat untuk memperoleh gelar Magister Komputer pada Program Studi S2 PJJ Informatika Universitas Amikom Yogyakarta.

Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak akan sulit bagi saya untuk menyelesaikan tugas akhir dari program magister ini. Oleh sebab itu, saya mengucapkan terima kasih kepada pihak yang telah membantu saya di antaranya:

1. Ida, Azam, dan Mia yang senantiasa mendukung serta mendoakan kelancaran dan kemudahan.
2. Bapak dan Ibu Dosen PJJ S2 Teknik Informatika Universitas Amikom Yogyakarta. Terkhusus Bapak Prof. Arief Setyanto S.Si., M.T., Ph.D. atas arahan dan bimbingan Tesis.
3. Ibu Prof. Dr Kusriani, M.Kom dan Bapak I Made Artha Agastya, S.T, M.Eng., P.hD. atas arahan dan bimbingan sehingga penulis mempunyai satu karya tulis Jurnal Ilmiah bereputasi.
4. Rekan-rekan Angkatan 9 PJJ S2 Informatika Amikom atas sharing ilmu, dukungan, dan Kerjasama kelompok dalam menjalani proses perkuliahan.

5. Rekan-rekan kerja di Bagian Perencanaan PLN UP3 Banyuwangi atas saran dan diskusi dalam penulisan Tesis.
6. Seluruh Pihak yang telah mendukung proses belajar jenjang Magister.

Hanya dengan Rahmat Allah SWT, dzat Yang Maha pengasih, Maha penyayang, Maha mengetahui dan penguasa ilmu dunia akhirat, yang mampu membalas semua kebaikan mereka.

Yogyakarta, 02 Januari 2026

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiv
INTISARI.....	xvi
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	10
1.3. Batasan Masalah.....	12
1.4. Tujuan Penelitian.....	12
1.5. Manfaat Penelitian.....	13
1.6. Hipotesis.....	14
BAB II TINJAUAN PUSTAKA.....	15
2.1. Tinjauan Pustaka.....	15
2.2. Keaslian Penelitian.....	18

2.3. Landasan Teori.....	22
a. Sentiment Analisis.....	22
b. Klasifikasi.....	23
c. <i>Machine Learning & Deep Learning</i>	23
d. <i>Confusion Matrix</i>	24
e. Layer Convolut	24
f. Akurasi	25
g. LSTM	25
h. BI-LSTM	26
i. Simple RNN	26
j. GRU (Gated Recurrent Unit)	27
k. <i>Word Embedding</i>	27
l. <i>Bidirectional Encoder Representations from Transformers</i>	28
m. <i>Keras Embedding</i>	29
n. <i>Decoding-enhanced BERT with disentangled attention</i>	29
o. <i>Robustly optimized BERT approach</i>	29
p. Word2Vec	30
q. FastText	30
r. GloVe	30
S. <i>Parallel dan Sequential Layer Neural Network</i>	31
T. <i>Ensemble learning</i>	31
U. SMOTE	32
V. K-Fold	32

W. T-Test.....	32
X. One Way ANOVA.....	33
BAB III METODE PENELITIAN.....	34
3.1. Jenis, Sifat, dan Pendekatan Penelitian	34
3.2. Metode Pengumpulan Data	34
3.3. Metode Analisis Data	35
3.4. Alur Penelitian.....	37
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	41
4.1. Pengambilan Dataset	41
4.2. Pre-Processing Text dan Tokenizing.....	41
4.3. Implementasi <i>Word Index</i>	42
4.4. <i>Split Train-Test Data</i>	43
4.5. Implementasi SMOTE.....	43
4.6. <i>Build Model dan inisiasi fungsi Word Embedding</i>	45
4.7. <i>Training</i>	46
4.8. Evaluation.....	47
4.9. Store Result.....	53
4.10. <i>Compare Result</i>	54
4.11. Pembahasan	55
BAB V PENUTUP.....	60
5.1. Kesimpulan.....	60
5.2. Saran	61
DAFTAR PUSTAKA	63

DAFTAR TABEL

TABEL 2. 1 MATRIKS LITERATUR REVIEW DAN POSISI PENELITIAN <i>OPTIMALISASI KONFIGURASI PARALEL NEURAL NETWORK PADA SENTIMENT ANALISIS TEKS BAHASA INGGRIS</i>	18
TABEL 4. 1 T-TEST KONFIGURASI <i>SINGLE LEARNER – PARALLEL LEARNER</i>	48
TABEL 4. 2 T-TEST <i>BiLSTM – GRU vs BiLSTM-GRU-RNN</i>	49
TABEL 4. 3 PERBANDINGAN T-TEST <i>SMOTE vs NON-SMOTE</i>	51
TABEL 4. 4 <i>CLASSIFICATION REPORT</i>	52
TABEL 4. 5 REKAP HASIL PERCOBAAN <i>SINGLE LEARNER NEURAL NETWORK</i>	53
TABEL 4. 6 TABEL REKAP HASIL PERCOBAAN <i>PARALLEL LEARNER</i>	54
TABEL 4. 7 TABEL REKAP PERBANDINGAN DENGAN PENELITIAN TERDAHULU	54

DAFTAR GAMBAR

GAMBAR 1. 1 ILUSTRASI RNN	3
GAMBAR 1. 2 ILUSTRASI LSTM	4
GAMBAR 1. 3 KONFIGURASI <i>PARALLEL LEARNER ENSEMBLE</i>	8
GAMBAR 1. 4 <i>SINGLE LEARNER NEURAL NETWORK</i>	9
GAMBAR 2. 1 ILLUSTRASI CONFUSION MATRIX	24
GAMBAR 2. 2 OPERASI CONVOLUSI	25
GAMBAR 2. 3 GAMBAR RUMUS AKURASI	25
GAMBAR 2. 4 ILUSTRASI BERT	28
GAMBAR 3. 1 KOMPOSISI LABEL DATASET	35
GAMBAR 3. 2 CONTOH KONFIGURASI <i>SINGLE LEARNER NEURAL NETWORK</i>	37
GAMBAR 3. 3 CONTOH KONFIGURASI MODEL <i>PARALLEL LEARNER NEURAL NETWORK</i>	38
GAMBAR 3. 4 DIAGRAM ALIR PENELITIAN	39
GAMBAR 4. 1 WEBSITE SUMBER DATASET	41
GAMBAR 4. 2 GAMBAR TEXT <i>PRE-PROCESSING</i>	42
GAMBAR 4. 3 <i>SPLIT DATA TRAIN-TEST</i>	43
GAMBAR 4. 4 DISTRIBUSI <i>KELAS DATASET SEBELUM APPLYING SMOTE</i>	44
GAMBAR 4. 5 DISTRIBUSI <i>KELAS DATASET SETELAH APPLYING SMOTE</i>	44
GAMBAR 4. 6 <i>CONTOH MODEL SUMMARY</i>	45
GAMBAR 4. 7 PROSES TRAINING DATA	47
GAMBAR 4. 8 PERBANDINGAN T-TEST KONFIGURASI <i>SINGLE LEARNER -PARALEL LEARNER</i>	48

GAMBAR 4. 9 PERBANDINGAN T-TEST BiLSTM-GRU-RNN vs BiLSTM-GRU..	49
GAMBAR 4. 10 PERBANDINGAN T-TEST SMOTE DAN NON-SMOTE.....	50
GAMBAR 4. 11 F-STATISTIC <i>LEARNING RATES COMPARISON</i>	51
GAMBAR 4. 12 <i>CONFUSION MATRIX</i>	52
GAMBAR 4. 13 <i>ROC CURVE VALIDATION</i>	53



INTISARI

Analisis sentimen sangat penting untuk mendapatkan data terkait opini pelanggan terhadap produk dan layanan yang ditawarkan. Dalam proses analisis sentimen, kombinasi dan konfigurasi jaringan syaraf tiruan mempengaruhi keakuratan klasifikasi sentimen. Dalam penelitian terbaru yang dilakukan oleh peneliti, ditemukan bahwa algoritma Bi-LSTM, LSTM, GRU, dan BERT merupakan *State of the Art* dalam proses klasifikasi analisis sentimen. Namun, penulis melihat bahwa penelitian sebelumnya yang dilakukan oleh peneliti lain difokuskan untuk menemukan hasil klasifikasi terbaik dari beberapa algoritma yang digunakan. Peneliti melihat bahwa *Word Embedding* memiliki peran penting dalam keakuratan klasifikasi analisis sentimen, dan konfigurasi jaringan syaraf tiruan *parallel ensemble learning neural network* dapat mempengaruhi keakuratan klasifikasi analisis sentimen. Dengan adanya peluang penelitian ini, peneliti mencoba melakukan percobaan dalam dua tahap, tahap pertama dengan konfigurasi *single learner neural network* dengan berbagai kombinasi algoritma *word embedding* dan klasifikasi. Hasil tahap pertama yaitu algoritma Bi-LSTM dengan embedding Keras dengan konfigurasi *single learner* menghasilkan keakuratan terbaik sebesar 80,20%. Hasil tahap pertama ini menjadi baseline yang akan dijadikan acuan untuk kombinasi pada percobaan tahap kedua. Pada tahap kedua percobaan dilakukan penggabungan algoritma Bi-LSTM dengan algoritma lain secara paralel, seperti GRU, RNN, dan Simple RNN dengan embedding Keras. Ditemukan bahwa penggabungan 3 layer paralel GRU-BiLSTM-RNN dengan Embedding Keras menghasilkan akurasi tertinggi untuk analisis sentimen dari 3 kelas sebesar 88%.

Kata kunci : *Sentiment Analysts, Parallel ensemble learning, Bi-LSTM, GRU, Keras.*

ABSTRACT

Sentiment analysis is critical to obtain data related to customer opinions on the products and services offered. In the process of sentiment analysis, the combination and configuration of neural networks affect the accuracy of sentiment classification. In recent research conducted by researchers, it was found that the Bi-LSTM, LSTM, GRU, and BERT algorithms are the State of the Art in the sentiment analysis classification process. However, the author sees that previous research conducted by other researchers focused on finding the best classification results from several algorithms used. Researchers see that Word Embedding has an important role in the accuracy of sentiment analysis classification, and the configuration of the parallel ensemble learning neural network can affect the accuracy of sentiment analysis classification. With this research opportunity, researchers tried to conduct experiments in two stages. The first stage was a single learner configuration with various combinations of word embedding and classification algorithms. The results of the first stage are the Bi-LSTM algorithm with Keras embedding with a single learner configuration, producing the best accuracy of 80.20%. The results of this first stage are the baseline to be used as a reference for the combination in the second stage of the experiment. In the second stage of the experiment, a combination of the Bi-LSTM algorithm with other algorithms was carried out in parallel, such as GRU, RNN, and Simple RNN with Keras embedding. It was found that the combination of 3 parallel layers of GRU-BiLSTM-RNN with Keras Embedding produced the highest accuracy for sentiment analysis of 3 classes of 88%.

Keyword: *Sentiment Analysis, Parallel ensemble learning, Bi-LSTM, GRU, Keras.*

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

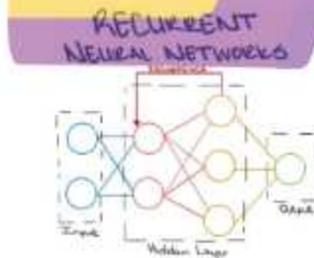
Dalam bisnis kepuasan pelanggan merupakan faktor penting yang harus diperhatikan (Khadka et al., 2017; Lee et al., 2016; Zheng et al., 2017). Untuk toko yang menyediakan berbagai macam produk untuk dijual perlu dievaluasi kualitas product yang ditawarkan apakah sesuai dengan harapan pelanggan. Perkembangan jaringan internet yang pesat telah mendapatkan pengguna aktif sebesar 4.13 milyar pada tahun 2021 yang terus meningkat dibandingkan tahun-tahun sebelumnya (Lukasz Wilk-Jakubowski Politechnika Świętokrzyska & Jacek, n.d., 2021). Pengguna aktif internet mendorong berkembangnya marketplace sebagai platform jual beli online (Humaizi et al., 2020; Ijaz & Rhee, 2018; Singh et al., 2018). Marketplace yang terus berkembang memudahkan baik pelaku usaha maupun konsumen untuk bertransaksi tanpa kendala jarak dan waktu, karena masalah tersebut sudah diberikan solusi berupa jaringan internet dan platform marketplace online. Selain itu marketplace online menyediakan kolom ulasan produk untuk mendapatkan umpan balik produk yang ditawarkan dari konsumen terkait rating produk dan juga komentar atau ulasan atas produk dan layanan yang diberikan oleh *merchant*. Sebagaimana diungkapkan (Zhang et al., 2018) bahwa microblog memberikan informasi terkait opini konsumen terhadap produk dan layanan. Sentiment analysis menjadi hal yang penting untuk dilakukan terhadap opini konsumen atas sebuah produk untuk meningkatkan kualitas pelayanan. Ada

beberapa metode untuk melakukan sentiment analysis terhadap suatu topik permasalahan yaitu *rule based*, *lexical based*, *machine learning* dan *deep learning*. *Machine Learning based* sentiment analysis seperti *Support Vector Machine* (SVM), *Naïve Bayes* (NB), dan *Random Forest*. Sentiment analysis dengan data yang telah diberi label oleh manusia sebagai *ground truth* (Haque et al., 2022).

Pada tahun 2022 (Haque et al., 2022) menyatakan pendekatan *neural network* digunakan untuk mengatasi kelemahan dalam proses sentiment analysis *rule based* dan *machine learning based* yang membutuhkan banyak proses manual dalam membuat *feature*. Inti dari model ini adalah pendekatan *machine learning embedding* model yang memuat *low dimentional continpous feature vector* yang tidak diperlukan lagi diproses secara manual. *Word Embedding* adalah *dense*, *distributed*, *fixed-length word vectors*, yang dibangun berdasarkan statistic kemunculan kata berdasarkan kedekatan distribusi (Almeida & Xexéo, 2019). Model *embedding* pertama kali dikenalkan oleh (Deerwester et al., n.d., 1990) di tahun 1989 adalah *Latent Semantic Analysis* (LSA). Sedangkan (Bengio et al., 2003) mengenalkan *neural language model* berbasis *feed forward neural network* yang ditraining pada 14 juta kata. Hasil dari model ini masih tidak lebih baik daripada metode klasik dengan proses *feature* secara manual sehingga tidak populer digunakan. Pada tahun 2013 (Mikolov et al., 2013) Bersama Google mengembangkan *Word2Vec* model yang ditraining pada 6 milyar kata dan menjadi model yang populer dalam NLP. *GloVe* oleh (Pennington et al., 2014) pada tahun 2014 dikenalkan, *GloVe* menggabungkan *global matrix factorization* dan *local context window*. Metode ini melakukan training pada *non-zero elements* pada *word co-occurrence matrix* dan

tidak melakukan pada seluruh matrix pada *context window* dalam *large corpus*. Pada tahun 2016 (Bojanowski et al., 2016) mengenalkan FastText, pada metode ini diusulkan pendekatan baru berdasarkan *skip gram model*, di mana setiap kata direpresentasikan sebagai atas karakter n-gram. Representasi vektor dikaitkan dengan setiap karakter n-gram, kata-kata direpresentasikan sebagai jumlah atas representasi ini.

Artificial Neural Network (ANN) (Reimers et al., 2020) adalah algoritma yang terinspirasi dari system kerja tubuh manusia berupa jaringan kompleks antar neuron yang membentuk jaringan saraf. Bentuk *neural network* yang sederhana adalah *feed forward neural network* yang terdiri dari *input layer*, *intermediate hidden layer* dan *output layer*. Untuk menyelesaikan permasalahan *time series* maka dikembangkan model *Recurrent Neural Network* (RNN). RNN sesuai dengan penjelasan (Schmidt,n.d., 2019) biasa digunakan untuk menyelesaikan masalah yang tergantung pada urutan data seperti teks, audio, dan waktu. Kelemahan RNN adalah hanya mampu melihat sinyal sebanyak kurang lebih 10 langkah sehingga memiliki masalah mempelajari data yang memiliki ketergantungan jarak jauh karena mengalami *vanishing gradient*.



Gambar 1. 1 Ilustrasi RNN

Untuk mengatasi masalah ini maka dikenalkan metode LSTM (*Long Short-Term Memory*) (Hochreiter & Schmidhuber, 1997) dengan memodifikasi RNN. LSTM memiliki *input gate*, *forget gate*, dan *output gate* sehingga LSTM mampu mempelajari lebih dari 1000 langkah sebelumnya tergantung dari kompleksitas jaringan.



Gambar 1. 2 Ilustrasi LSTM

Pada penelitian yang dilakukan (Alshamsi et al., 2020) dilakukan beberapa percobaan *sentiment analysis* menggunakan berbagai macam algoritma *machine learning* seperti *Decision Tree*, *Naïve Bayes*, *Random Forest*, K-NN, ID3, dan *Random Tree* terhadap dataset dari website Kaggle dengan tema *American airline companies* yang memiliki 14.000 tweet yang telah dilabeli dengan klasifikasi positif, negatif, dan netral. Hasil penelitian ini dengan dataset yang *balance* didapatkan akurasi tertinggi pada algoritma Naïve Bayes dengan rata-rata sebesar 76,10% sedangkan pada dataset *unbalance* didapat akurasi rata-rata tertinggi 58,86%. Namun demikian peneliti melihat masih ada potensi untuk meningkatkan akurasi *sentiment analysis* menggunakan algoritma berbasis RNN seperti LSTM, Bi LSTM, SimpleRNN, *Gated Recurrent Unit* (GRU). Selain itu menurut (Khan et al., 2022) yang melakukan percobaan dengan menambah layer Convulasi sebelum

layer LSTM pada struktur model sentiment analysis dalam tiga kelas pada dataset IMDB Review menghasilkan 84,10%. Penelitian yang dilakukan oleh (Patel et al., 2022) dilakukan klasifikasi dalam tiga kelas, yaitu positif, negatif, dan netral. Dilakukan terhadap dataset *American Airlines Company* dengan menggunakan *Random Forest Classifier* sebagai *baseline* eksperimen berbasis *Machine Learning*. Selanjutnya peneliti menggunakan model berbasis LSTM, Roberta, dan Electra untuk membandingkan akurasi algoritma berbasis *Machine Learning* dengan algoritma *Deep Learning*. Hasil penelitian menunjukkan bahwa BERT menghasilkan akurasi terbaik dengan angka 83%, diikuti oleh model Roberta dengan akurasi 80,8%, kemudian diikuti oleh Model Electra dengan akurasi 79,8%, kemudian model *Random Forest* menghasilkan akurasi 77%, dan terakhir model LSTM menghasilkan akurasi 76%. Peneliti juga menyimpulkan bahwa model berbasis *Deep Learning* mengungguli akurasi algoritma berbasis *machine learning*, kecuali model LSTM yang akurasinya berada di bawah model *Random Forest*. Untuk algoritma berbasis BERT menghasilkan akurasi sebesar 83% dan masih lebih unggul dibandingkan penelitian sebelumnya yang dilakukan (Kumawat et al., 2021) dengan menggunakan RoBERTA yang menghasilkan akurasi sebesar 80,8%.

Berdasarkan referensi penelitian yang telah disebutkan di bagian sebelumnya, ditemukan bahwa dalam proses klasifikasi analisis sentimen, algoritma BERT, BiLSTM, LSTM, dan GRU merupakan algoritma yang merupakan *State Of The Art* dalam proses klasifikasi sentimen.

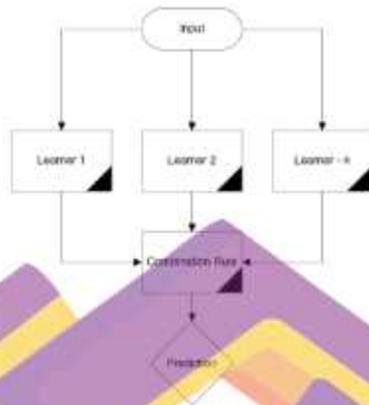
Penelitian yang dilakukan (Wadawadagi & Pagi, 2020) menyatakan bahwa deep learning menjadi salah satu teknik yang paling baik dalam proses klasifikasi

sentiment analysis karena memberikan kemudahan otomatisasi ekstraksi fitur pada teks. Peneliti membandingkan akurasi dari beberapa algoritma kontemporer berbasis *deep learning* dalam proses klasifikasi. Selain itu peneliti juga melakukan tuning *hyperparameter* untuk memberikan gambaran pengaruh pada proses klasifikasi sentiment analysis. Penelitian dilakukan pada dataset IMDB yang terdiri dari dua kelas yaitu positif dan negative. Dataset SEC16 yang terdiri dari tiga kelas yaitu positif, netral, dan negative. Dataset SLS yang terdiri dari dua kelas yaitu positif dan negative. Dan yang terakhir dataset SST yang terdiri dari lima kelas yaitu sangat positif, positif, sangat negative, negative, dan netral. Percobaan pada dataset IMDB algoritma GRU+RAM dengan 32 hidden unit, 64 mini batch, 2 hidden layer, 20 epoch, Adam Optimizer, softmax activation, dan 0,01 learning rate menghasilkan akurasi tertinggi sebesar 0,919. Percobaan pada SEC16 algoritma LSTM+Att dengan 32 hidden unit, 64 minibatch, 1 hidden layer, 10 epoch, AdaGard Optimizer, softmax activation, dan 0,005 learning rate menghasilkan akurasi tertinggi 0,907. Pada dataset SLS algoritma BiGRU+Att dengan 16 hidden unit, 64 minibatch, 2 hidden layer, 20 epoch, AdaDelta Optimizer, Softmax activation, dan 0,005 learning rate menghasilkan akurasi tertinggi sebesar 0,932. Pada dataset SST dilakukan penelitian dalam dua skema yaitu dataset dalam lima kelas (positif, sangat positif, negative, sangat negative, netral) dan regularisasi kelas dalam binary (positif, negative). Hasil klasifikasi lima kelas didapatkan hasil akurasi terbaik 45,78% dengan algoritma RTNN, sedangkan klasifikasi dalam dua kelas didapat akurasi 85,4% dengan algoritma yang sama yaitu RTNN. Peneliti menyimpulkan bahwa LSTM, GRU, CNRR, dan RTNN adalah algoritma yang

powerful dalam proses klasifikasi sentiment analysis. Peneliti juga memberi saran kepada peneliti lain untuk melakukan percobaan menggunakan algoritma *ensemble network* dimana beberapa jenis algoritma dikombinasikan dalam proses klasifikasi sentiment analysis untuk meningkatkan generalisasi dalam proses *feature extractions*.

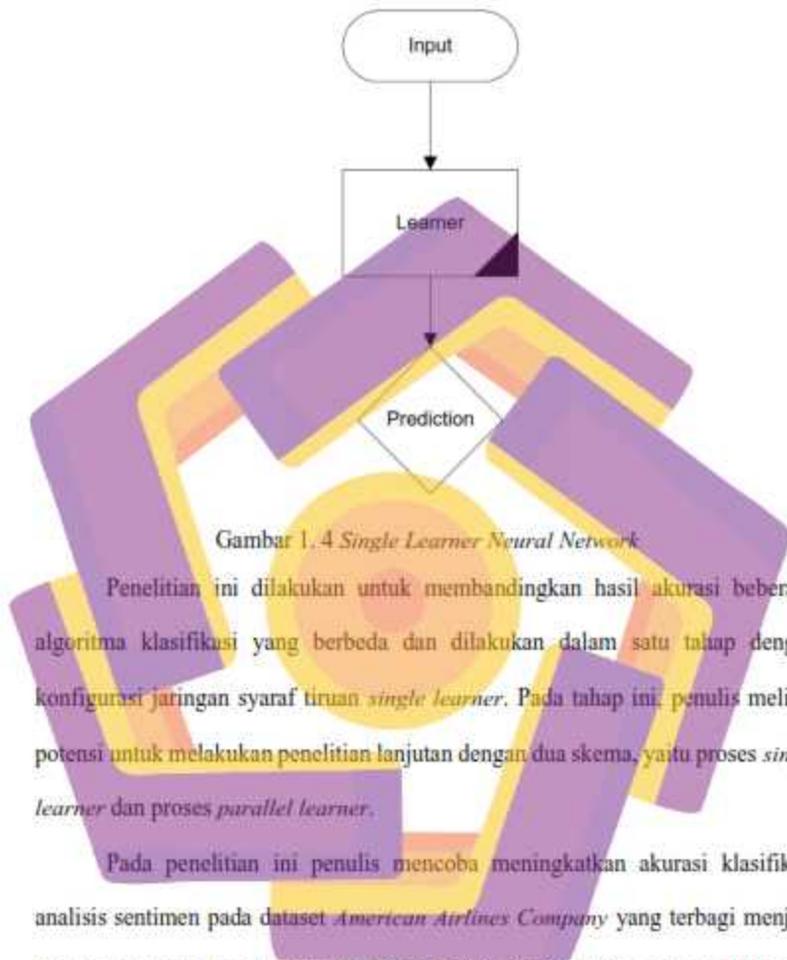
Penelitian (Kazmaier & van Vuuren, 2022) menguatkan saran dari (Wadawadagi & Pagi, 2020) bahwa *ensemble learning* dalam proses klasifikasi sentiment analysis dapat meningkatkan akurasi rata-rata sebesar 5,53%. Menurut peneliti penelitian klasifikasi sentiment analysis menggunakan teknik *ensemble learning* masih terbatas. Ensemble learning memiliki kelebihan yang dapat memberikan kompensasi atas kelemahan algoritma lain sehingga dari beberapa jenis algoritma yang digunakan bersama dapat saling melengkapi.

Penelitian yang dilakukan (Jadama et al., n.d., 2024) juga menguatkan teori bahwa *ensemble learning* meningkatkan *performance machine learning* daripada *single model*. Menurut peneliti teknik *stacking* yang mengkombinasikan algoritma yang berbeda pada proses *ensemble learning* adalah yang paling ideal karena dapat mengakomodir *bias* dan *variance* sekaligus. Menurut peneliti konfigurasi *ensemble learning* dapat disusun secara *parattel learner* dan *sequential learner*. Konfigurasi parallel dapat diilustrasikan pada gambar 1.3.



Gambar 1. 3 Konfigurasi *parallel learner ensemble*

Berdasarkan penelitian (Mohammed & Kora, 2023) saat ini metode *ensemble learning* dan *deep learning* menghasilkan performa model kecerdasan buatan yang baik. Untuk menghasilkan model kecerdasan yang memiliki performa tinggi disebutkan beberapa metode penyusunan baseline model dibuat. Tantangan saat ini menurut disebutkan bahwa pengalaman peneliti untuk mensetting optimal *hyperparameter yang dibutuhkan* dan juga dalam menyusun konfigurasi model. Menurut data yang dikumpulkan oleh peneliti trend penggunaan metode ensemble learning meningkat secara signifikan diantara tahun 2014 sampai dengan tahun 2021. Menurut peneliti, kerangka kerja ensemble learning terdiri dari tiga jenis yaitu konfigurasi baseline model parallel-sequential, metode fusi, dan heterogeneity. Proses training pada umumnya menggunakan satu learner yang pada penelitian ini disebut sebagai *single traditional learner*. Konfigurasi ini digambarkan pada gambar 1.4.



Gambar 1.4 *Single Learner Neural Network*

Penelitian ini dilakukan untuk membandingkan hasil akurasi beberapa algoritma klasifikasi yang berbeda dan dilakukan dalam satu tahap dengan konfigurasi jaringan syaraf tiruan *single learner*. Pada tahap ini penulis melihat potensi untuk melakukan penelitian lanjutan dengan dua skema, yaitu proses *single learner* dan proses *parallel learner*.

Pada penelitian ini penulis mencoba meningkatkan akurasi klasifikasi analisis sentimen pada dataset *American Airlines Company* yang terbagi menjadi tiga kelas dengan menerapkan kombinasi RNN, BiLSTM, Keras Embedding dan konfigurasi *parallel ensemble learner* untuk meningkatkan proses ekstraksi fitur yang kemudian digabungkan untuk dilakukan klasifikasi.

Untuk memudahkan pemahaman *term parallel ensemble learner*, pada tahap selanjutnya akan ditulis dengan *term parallel learner* sebagaimana referensi penelitian sebelumnya.

Selanjutnya dari penelitian ini dapat memberikan kontribusi pengetahuan terkait hal sebagai berikut :

- a. Perbandingan akurasi klasifikasi sentiment berdasarkan jenis word embedding yang digunakan.
- b. Perbandingan akurasi jenis algoritma klasifikasi sentimen yang digunakan
- c. Perbandingan konfigurasi *single learner neural network* dan *parallel learner neural network* yang digunakan terhadap akurasi klasifikasi sentimen analisis
- d. Perbandingan nilai akurasi *sentiment analysis* dengan implementasi SMOTE dan Non-SMOTE pada data training.
- e. Perbandingan nilai akurasi *sentiment analysis* dengan implementasi berbagai macam nilai *learning rate*.

1.2. Rumusan Masalah

Berdasarkan penelitian yang dilakukan (Alshamsi et al., 2020) beberapa percobaan sentiment analysis menggunakan berbagai macam algoritma *machine learning* seperti *Decision Tree*, *Naïve Bayes*, *Random Forest*, K-NN, ID3, dan *Random Tree* dengan dataset *American Airlines Compay* dari Kaggle didapatkan akurasi tertinggi pada algoritma Naïve Bayes dengan rata-rata sebesar 76,10% sedangkan pada dataset *unbalance* didapat akurasi rata-rata tertinggi 58,86%. Penelitian yang dilakukan oleh (Patel et al., 2022) dilakukan klasifikasi dalam tiga

kelas, yaitu positif, negatif, dan netral. Dilakukan terhadap dataset *American Airlines Company* dengan menggunakan *Random Forest Classifier* sebagai *baseline* eksperimen berbasis *Machine Learning*. Selanjutnya peneliti menggunakan model berbasis LSTM, Roberta, dan Electra untuk membandingkan akurasi algoritma berbasis *Machine Learning* dengan algoritma *Deep Learning*. Hasil penelitian menunjukkan bahwa BERT menghasilkan akurasi terbaik dengan angka 83%.

Penelitian yang dilakukan (Hidasi et al., 2016) menyimpulkan arsitektur *parallel neural network* dapat meningkatkan kinerja karena efisiensi yang tinggi, proses *parallel network* memungkinkan untuk mendapatkan *feature extraction* yang lebih kaya dari beberapa jaringan *neural network* yang selanjutnya digabung pada *dense layer* sebelum diklasifikasikan sebagai *output*.

Dari penelitian yang dijadikan referensi, algoritma BERT, BiLSTM, LSTM, dan GRU merupakan algoritma yang merupakan *State Of The Art* dalam proses klasifikasi sentimen. Selain itu, berdasarkan penelitian terdahulu dan penelitian sejenis, proses klasifikasi dilakukan dalam satu tahap dengan konfigurasi jaringan syaraf tiruan *single learner*. Pada tahap ini, penulis melihat potensi untuk melakukan penelitian lanjutan dengan dua skema, yaitu proses *single learner* dan proses *parallel learner*.

Selanjutnya dapat dirumuskan beberapa pertanyaan yang perlu dibuktikan lewat penelitian diantaranya :

- a. Apa pengaruh algoritma LSTM, Bi LSTM, dan GRU terhadap akurasi klasifikasi *sentiment analysis*?

- b. Apa pengaruh beberapa *metode word embedding* terhadap akurasi klasifikasi sentimen analisis pada teks Bahasa Inggris?
- c. Konfigurasi *single learner* dan *parallel learner*, apa pengaruhnya terhadap akurasi yang dihasilkan?
- d. Apa pengaruh proses SMOTE pada akurasi klasifikasi *sentiment analysis*?
- e. Apa pengaruh nilai *learning rate* terhadap akurasi klasifikasi *sentiment analysis*?

1.3. Batasan Masalah

Pada penelitian ini Batasan masalah yang dilakukan adalah :

- a. Dataset yang digunakan adalah dataset *American Airlines Compay* dari website www.kaggle.com.
- b. Bahasa dataset yang digunakan dalam Bahasa Inggris
- c. Hasil diukur dengan metrik akurasi dasar yaitu Presisi, Recall, dan F-1 Score dan diuji dengan metode T-Test serta ANOVA.
- d. Platform menggunakan Google Collab Pro dengan Runtime T4-High RAM.

1.4. Tujuan Penelittan

Tujuan penelitian ini adalah mengetahui dan mengembangkan pengetahuan atas hal sebagai berikut :

- a. Mengetahui pengaruh kombinasi implementasi algoritma *neural network* untuk klasifikasi sentimen analisis terhadap dataset yang digunakan.
- b. Mengetahui pengaruh berbagai macam metode *word embedding* dalam akurasi klasifikasi sentimen.

- c. Mengetahui pengaruh konfigurasi *single learner neural network* dan *parallel learner neural network*.
- d. Mengetahui pengaruh implementasi SMOTE dalam proses klasifikasi *sentiment analysis*.
- e. Mengetahui pengaruh nilai *learning rate* pada proses klasifikasi *sentiment analysis*.
- f. Memberikan rekomendasi dan panduan metodologi dalam menerapkan *word embedding* dalam klasifikasi sentimen analisis teks berbahasa Inggris.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

- a. Memberikan kontribusi dalam metodologi klasifikasi sentimen analisis dalam tiga kelas terhadap text berbahasa Inggris.
- b. Memberikan kontribusi pengetahuan pengaruh *Word Embedding* terhadap akurasi klasifikasi dengan berbagai macam teknik *Word Embedding* yang digunakan.
- c. Memberikan kontribusi pengetahuan pengaruh konfigurasi *single learner neural network* dan *parallel learner neural network*.
- d. Memberikan kontribusi pengetahuan terkait pengaruh proses SMOTE pada klasifikasi *sentiment analysis*.
- e. Memberikan kontribusi pengetahuan terkait pengaruh nilai *learning rate* pada klasifikasi *sentiment analysis*.

1.6. Hipotesis

Berdasarkan *literatur review* dan landasan teori maka penggunaan algoritma Bi-LSTM dan kombinasi *word embedding* BERT berpengaruh dalam meningkatkan akurasi klasifikasi sentimen analisis karena merupakan *state of the art* dalam klasifikasi *sentiment analysis*. Konfigurasi *parallel neural network* dapat menghasilkan akurasi yang lebih baik dengan waktu training lebih singkat karena proses ekstraksi *feature* dikerjakan secara *parallel* untuk dilakukan penggabungan pada tahap akhir sebelum masuk ke *dense layer* untuk proses klasifikasi. Sebagaimana rujukan penelitian yang dilakukan oleh (Jadama et al., n.d., 2024) dan penelitian yang dilakukan oleh (Kazmaier & van Vuuren, 2022) menyatakan *ensemble learning* dapat meningkatkan performa *machine learning*. Selanjutnya hipotesis nol (H_0) untuk rumusan penelitian ini dapat ditulis sebagai berikut :

1. Konfigurasi *parallel learner* tidak memberikan peningkatan akurasi klasifikasi *sentiment analysis* yang signifikan dibanding konfigurasi *single learner*.
2. Implementasi SMOTE pada data training tidak memberikan peningkatan akurasi klasifikasi *sentiment analysis* yang signifikan dibanding data training yang tidak dilakukan proses SMOTE.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

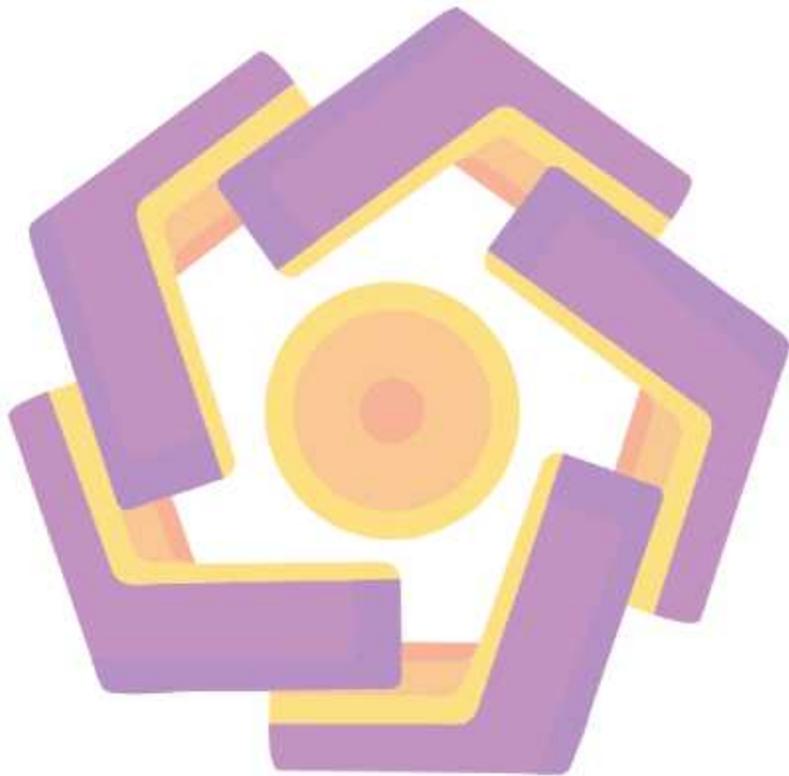
Penelitian yang dilakukan (Singgalen, 2024) melakukan penelitian sentiment analysis untuk layanan *hotel review* dalam tiga kelas menggunakan algoritma LSTM-GRU menghasilkan akurasi 91%. Sedangkan penelitian (Mahadevaswamy & Swathi, 2022) melakukan penelitian sentiment analysis menggunakan algoritma Bi-LSTM untuk dataset *Amazon Product Review* dalam dua kelas menghasilkan akurasi 91,40%. Pada penelitian (Aakash et al., 2024) dilakukan klasifikasi sentiment analysis untuk dataset *amazon product review* menggunakan algoritma GRU menghasilkan akurasi 87%. Pada penelitian (Alshamsi et al., 2020) dilakukan beberapa percobaan sentiment analysis menggunakan berbagai macam algoritma *machine learning* seperti *Decision Tree*, *Naïve Bayes*, *Random Forest*, K-NN, ID3, dan *Random Tree* terhadap dataset dari website Kaggle dengan tema *American airline companies* yang memiliki 14.000 tweet yang telah dilabeli dengan klasifikasi positif, negatif, dan netral. Hasil penelitian ini dengan dataset yang *balance* didapatkan akurasi tertinggi pada algoritma *Naïve Bayes* dengan rata-rata sebesar 76,10% sedangkan pada dataset *unbalance* didapat akurasi rata-rata tertinggi 58,86%. Selanjutnya pada (Khan et al., 2022) dilakukan percobaan dengan menambah layer Convulusi sebelum layer LSTM pada struktur model sentiment serta menggunakan *BERT Embedding* dan *Word2Vec* analysis dalam tiga kelas menghasilkan akurasi 74,8%. Selanjutnya

pada penelitian yang dilakukan oleh (Patel et al., 2022) untuk dataset yang sama yaitu *American airlines company* dilakukan beberapa percobaan algoritma yang berbasis *machine learning* seperti *Logistic Regression*, *KNN*, *Support Vector Machine*, *Decision Tree*, *Random Forest Classifier*, dan *adaboost*. Selain itu pada percobaan yang dilakukan peneliti menggunakan algoritma yang berbasis *Deep Learning* seperti *LSTM* dan *BERT*, dari penelitian tersebut dihasilkan bahwa klasifikasi berbasis *deep learning* menghasilkan akurasi yang lebih baik. Untuk algoritma berbasis *BERT* menghasilkan akurasi 83% dan masih unggul dari penelitian sebelumnya yang dilakukan (Kumawat et al., 2021) menggunakan *RoBERTA* menghasilkan akurasi 80,8%.

Pada penelitian sebelumnya yang menjadi rujukan, penelitian dilakukan percobaan dengan satu algoritma saja dan dengan konfigurasi *single learner neural network*. Selain itu dari penelitian terakhir yang menjadi rujukan, algoritma *LSTM*, *BiLSTM*, *GRU*, dan *BERT* menjadi algoritma populer yang menghasilkan akurasi terbaik atau biasa disebut *State Of The Art* dalam penelitian *sentiment analysis*. Berdasarkan rujukan dan saran dari paper yang menjadi referensi, *ensemble learning* dapat meningkatkan performa algoritma. Berdasarkan hal tersebut maka dalam penelitian ini dilakukan konfigurasi *ensemble learning* sebagai upaya untuk mencoba kombinasi beberapa algoritma *deep learning* dalam satu kali proses klasifikasi. Selain itu dilakukan percobaan dengan konfigurasi paralel untuk mendapatkan akurasi yang lebih baik dan waktu training yang lebih cepat.

Pada penelitian ini penulis mencoba untuk dapat meningkatkan akurasi klasifikasi sentimen analisis atas dataset *American Airlines Company* yang terbagi

dalam tiga kelas dengan menerapkan kombinasi RNN, BiLSTM, Keras Embedding dan konfigurasi paralel layer untuk meningkatkan proses *feature extraction* yang selanjutnya digabung untuk dilakukan klasifikasi, dan didapatkan akurasi terbaik 88%.



2.2. Keaslian Penelitian

Tabel 2. 1 Matriks literatur review dan posisi penelitian
Implementasi Parallel Ensemble Learning Neural Network Pada Sentiment Analisis Teks Bahasa Inggris

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tojuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>Sentiment Analysis of Customer Feedback and Reviews for Airline Services using Language Representation Model (Patel et al. 2022)</i>	Patel et.al, Proecdia Computer Science, 2023	Mendapatkan akurasi terbaik dari beberapa Algoritma	BERT mendapatkan akurasi tertinggi sebesar 83%	Akurasi masih 83%. Penulis memberi saran untuk mencoba varian BERT untuk penelitian selanjutnya	Berbagai kombinasi Word Embedding dan Konfigurasi Parallel Learner Neural Network pada penelitian
2	<i>An Exploration of Airline Sentimental Tweets with</i>	Vadivukarassi et.al, IJREAM, 2018	Mendapatkan akurasi terbaik dari beberapa Algoritma	Random Forest Classifier menghasilkan akurasi	Akurasi masih 81,50%. Potensi penelitian lanjutan	Berbagai kombinasi Word Embedding dan Konfigurasi

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tojuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	<i>Different Classification Model</i> (Vadivukarassi et al., 2018)			81,35% mengungguli algoritma lain yang digunakan	dengan algoritma berbasis deep learning	Parallel Learner Neural Network pada penelitian Menggunakan algoritma Deep Learning.
3	Opinion Mining on US Airline Twitter Data Using Machine Learning Techniques (Saad, 2020)	Saad et.al, ICENCO, 2020	Mendapatkan akurasi terbaik dari beberapa Algoritma	Algoritma SVM menghasilkan akurasi tertinggi sebesar 83,31%	Akurasi masih 83,31%. Potensi penelitian lanjutan dengan algoritma berbasis deep learning.	Menggunakan algoritma Deep Learning, Berbagai kombinasi Word Embedding dan Konfigurasi Parallel Neural Network pada penelitian
4	<i>Tweets Classification on the Base of Sentiments for US</i>	Rustam et.al, ENTROPY- MDPI, 2019	Mendapatkan akurasi terbaik dari beberapa Algoritma	Algoritma Limer Regression + stochastic gradient	Akurasi masih 79,4%. Potensi penelitian lanjutan	Menggunakan algoritma Deep Learning, Berbagai kombinasi Word Embedding dan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tojuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	<i>Airline Companies</i> (Rustam et al., 2019)			descent classifier menghasilkan akurasi 79,4%	dengan algoritma berbasis deep learning	Konfigurasi Parallel Learner Neural Network pada penelitian
5	<i>Sentiment Classification System of Twitter Data for US Airline Service Analysis</i> (Rane & Kumar, 2018)	Rane et.al, COMPSAC, 2018	Mendapatkan akurasi terbaik dari beberapa Algoritma	Algoritma AdaBoost menghasilkan akurasi 84,5%	Akurasi masih 84,5%. Potensi penelitian lanjutan dengan algoritma berbasis deep learning	Menggunakan algoritma Deep Learning, Berbagai kombinasi Word Embedding dan Konfigurasi NN yang dilakukan saat percobaan
6	<i>Sentiment Analysis of US Airline Twitter Data using New Adaboost Approach</i> (Prabhakar et al., 2019)	Prabhakar et.al, IJERT, 2019	Mendapatkan akurasi terbaik dari beberapa Algoritma	Algoritma AdaBoost menghasilkan akurasi 78%	Akurasi masih 78%. Potensi penelitian lanjutan dengan algoritma	Menggunakan algoritma Deep Learning, Berbagai kombinasi Word Embedding dan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tojuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
					berbasis deep learning	Konfigurasi NN yang dilakukan saat percobaan
7	<i>Sentiment Analysis in English Texts</i>	Alshamir et.al, ASTES, 2020	Mendapatkan akurasi terbaik dari beberapa Algoritma	Algoritma Naïve Bayes mendapatkan akurasi rata-rata untuk semua Airlines Company 76,10%	Akurasi masih 76,10% Potensi penelitian lanjutan dengan algoritma berbasis deep learning	Menggunakan algoritma Deep Learning, Berbagai kombinasi Word Embedding dan Konfigurasi NN yang dilakukan saat percobaan

2.3. Landasan Teori

a. Sentiment Analysis

Sentimen Analisis adalah teknik dalam *Natural Language Processing* (NLP) untuk proses identifikasi pola informasi penting dari korpus teks. Proses sentiment analisis yang melakukan klasifikasi dari sebuah opini dalam beberapa target kelas. Dalam melakukan sentiment analisis menurut (Lamba & Madhusudhan, 2022) ada beberapa langkah yang dilakukan, diantaranya adalah sebagai berikut :

1. *Pre-Processing*

Dalam proses klasifikasi sentiment data yang diproses dari opini yang dituangkan dalam microblog adalah text yang tidak terstruktur dan mengandung *slang term* atau *bahasa gaul*, ejaan yang salah atau *wrong spelling*, karakter spesial, tanda baca, dan *stopwords*. *Preprocessing* adalah tahapan yang sangat penting dan menurut (Dharma & Saragih, 2022) dapat meningkatkan akurasi klasifikasi yang signifikan.

2. *Feature Extraction*

Proses ini dilakukan sebelum data digunakan sebagai input proses klasifikasi, pada proses ini data text atau bahasa manusia dilakukan konversi ke data numerik maupun vector agar dapat dibaca oleh komputer. Menurut (Ahuja et al., 2019) proses *feature extraction* yang sering digunakan adalah *Bag of Word*, TF-IDF, dan *Word Embedding*.

3. *Performing Sentiment Analysis*

Proses sentiment analisis dilakukan menggunakan algoritma dan tools yang sesuai untuk mengolah data sehingga didapatkan sentiment analisis dari sebuah text maupun artikel.

4. *Visualizing the sentiment*

Tahapan ini adalah proses menyajikan hasil klasifikasi sentiment analisis agar mudah untuk dibaca dan dipahami.

b. Klasifikasi

Klasifikasi opini adalah analisis yang bertujuan untuk menentukan sentimen terhadap aspek tertentu. Klasifikasi opini akan digunakan untuk penentuan untuk melihat pendapat seseorang terhadap suatu kebijakan, ulasan atau produk. Klasifikasi opini dapat dikategorikan menjadi 2 kelas, yaitu positif dan negatif, atau lebih dari 2 kelas sebagaimana disebutkan oleh (F.Y et al., 2017).

c. *Machine Learning & Deep Learning*

Dengan mengamati pola pembelajarannya sendiri dari waktu ke waktu, *machine learning* bertujuan untuk mengajarkan komputer untuk berperilaku seperti perilaku manusia. Saat ini, *machine learning* adalah salah satu cabang AI yang paling populer. Mengembangkan kemampuan mesin untuk mempelajari informasi baru dari data dan memecahkan masalah adalah tugas utama *machine learning*. Ini membutuhkan data pelatihan atau masukan untuk mengajari mesin cara memecahkan masalah, menjawab pertanyaan, dan menarik kesimpulan dari hasil pengolahan data. Algoritma *Machine Learning* terus berkembang dan semakin canggih seiring kemajuan teknologi (Janiesch et al., 2021).

d. Confusion Matrix

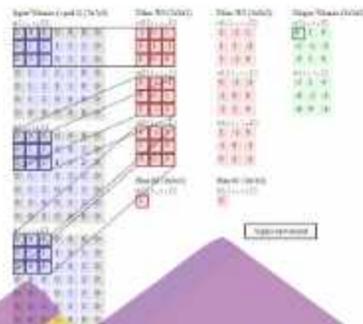
Mencari nilai akurasi, ketepatan, dan recall dari model klasifikasi adalah salah satu cara untuk mengetahui seberapa baik kinerjanya. Untuk mengukur kinerja model klasifikasi, tanda positif dan tanda negatif adalah istilah yang umum digunakan. Tanda positif adalah tanda yang menarik perhatian, sedangkan tanda negatif adalah tanda yang tidak menarik perhatian. Beberapa istilah lain yang digunakan dalam pencarian nilai adalah *accuracy*, *recall*, dan akurasi nilai positif benar (TP), negatif benar (TN), positif salah (FP), dan negatif salah (FN). Gambar 2.1 berikut menunjukkan matriks yang biasa digunakan untuk menggabungkan istilah-istilah tersebut:

		Predicted class		Total
		yes	no	
Actual class	yes	TP	FN	P
	no	FP	TN	N
Total		P ⁺	N ⁺	P + N

Gambar 2.1 Ilustrasi Confusion Matrix

e. Layer Convolusi

Convolutional layer terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixels) dan digunakan untuk ekstraksi fitur. Sebagaimana dijelaskan dalam (Girshick, 2015). Untuk proses operasi konvolusi diilustrasikan dalam Gambar 2.2 berikut.



Gambar 2. 2 Operasi Convulasi

f. Akurasi

Akurasi didefinisikan sebagai persentase dari data uji yang diklasifikasikan ke kelas yang benar. Akurasi dapat dinyatakan dalam persamaan berikut sebagaimana gambar 2.3.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Gambar 2. 3 Gambar Rumus Akurasi

g. LSTM

Artificial Neural Network (ANN) merupakan suatu algoritma yang terinspirasi dari sistem kerja tubuh manusia berupa jaringan neuron yang kompleks yang membentuk suatu *neural network*. Bentuk *neural network* yang paling sederhana adalah *feed-forward neural network* yang terdiri dari *input layer*, *intermediate hidden layer*, dan *output layer*. Untuk menyelesaikan permasalahan *time series*, dikembangkan model *Recurrent Neural Network* (RNN). RNN, (Schmidt, n.d.; Sherstinsky, n.d., 2019) umumnya digunakan untuk menyelesaikan permasalahan yang bergantung pada urutan data seperti teks, audio, dan waktu.

Kelemahan RNN adalah hanya mampu melihat sinyal selama kurang lebih 10 langkah sehingga memiliki kendala dalam mempelajari data yang memiliki dependensi jarak jauh karena mengalami *vanishing gradient*.

Untuk mengatasi permasalahan tersebut maka diperkenalkanlah metode LSTM (*Long Short-Term Memory*) (Hochreiter & Schmidhuber, 1997) dengan melakukan modifikasi pada RNN. LSTM memiliki *input gate*, *forget gate*, dan *output gate* sehingga LSTM dapat mempelajari lebih dari 1000 langkah sebelumnya tergantung dari kompleksitas jaringan.

h. Bi-LSTM

Bidirectional Long Short-Term Memory (BiLSTM) adalah pengembangan dari jaringan *Long Short-Term Memory* (LSTM) konvensional. BiLSTM memungkinkan input mengalir maju dan mundur, memungkinkan untuk dapat menangkap informasi kontekstual yang lebih dalam daripada LSTM tradisional, yang hanya menganalisis urutan dengan satu cara. Karena itu, BiLSTM sangat berguna untuk tugas-tugas yang membutuhkan pemahaman tentang konteks masa lalu dan masa depan sebagaimana dijelaskan dalam (Ghosh et al., 2023).

i. Simple RNN

Simple RNN (*Recurrent Neural Network*) adalah jenis jaringan saraf buatan khusus yang dirancang untuk menangani data deret waktu atau data yang melibatkan urutan. Jaringan saraf *feedforward* biasa hanya berfungsi untuk titik data yang independen satu sama lain, tetapi perlu mengubah jaringan saraf untuk menggabungkan ketergantungan antara titik data dalam urutan. Memiliki konsep "memori", RNN membantu menghasilkan *output* urutan berikutnya dengan

menyimpan status atau informasi input sebelumnya sebagaimana dijelaskan dalam (Sherstinsky, n.d., 2018).

J. GRU (*Gated Recurrent Unit*)

Gated Recurrent Units (GRUs) adalah jenis RNN yang diperkenalkan pada tahun 2014. Menggunakan mekanisme gerbang untuk secara otomatis memperbarui status tersembunyi pada setiap langkah waktu adalah inti dari GRU. Ini memungkinkan untuk dapat mengingat informasi penting dan secara paralel menghilangkan detail yang tidak penting. Dengan berfokus hanya pada dua gerbang utama-gerbang pembaruan dan gerbang *reset*, GRU bertujuan untuk menyederhanakan arsitektur LSTM (Chung et al., 2014).

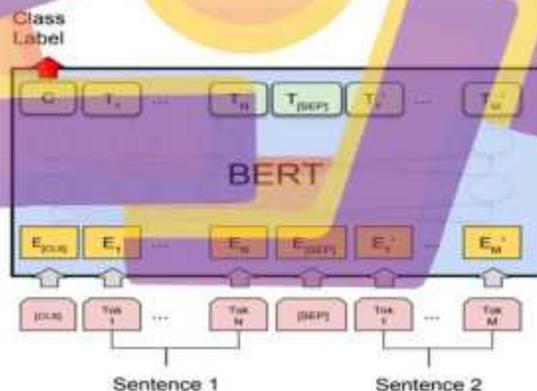
k. *Word Embedding*

Word Embedding (Almeida & Xexéo, 2019) merupakan pendekatan untuk merepresentasikan kata dan dokumen. *Word Embedding* atau *Word Vector* merupakan input vektor numerik yang merepresentasikan kata dalam ruang berdimensi lebih rendah. Pendekatan ini memungkinkan kata-kata dengan makna yang serupa memiliki representasi yang serupa. *Word Embeddings* adalah metode untuk mengekstraksi fitur dari teks sehingga kita dapat memasukkan fitur tersebut ke dalam model *machine learning* untuk bekerja dengan data teks. Proses ini mencoba untuk mempertahankan informasi sintaksis dan semantik. Metode seperti *Bag of Words* (BOW), *Count Vectorizer* dan TFIDF bergantung pada jumlah kata dalam sebuah kalimat tetapi tidak menyimpan informasi sintaksis atau semantik apa pun. Dalam algoritma ini, ukuran vektor adalah jumlah elemen dalam kosakata. Kita bisa mendapatkan matriks renggang jika sebagian besar elemennya adalah nol.

Vektor input yang besar akan berarti sejumlah besar bobot yang akan menghasilkan komputasi tinggi yang diperlukan untuk pelatihan. *Word Embeddings* memberikan solusi untuk masalah ini.

1. *Bidirectional Encoder Representations from Transformers*

Dalam penelitian (Devlin et al., n.d.), representasi linguistik yang disebut BERT (*Bidirectional Encoder Representations from Transformers*) diperkenalkan. BERT dirancang untuk melatih representasi dua dimensi yang mendalam dari teks yang tidak berlabel dengan pengkondisian gabungan pada konteks kiri dan kanan di semua lapisan. Model BERT yang telah dilatih sebelumnya dapat disempurnakan dengan satu lapisan keluaran tambahan untuk membuat model canggih untuk berbagai tugas, seperti menjawab pertanyaan dan inferensi linguistik tanpa harus melakukan beberapa modifikasi arsitektur khusus tugas. Ilustrasi cara kerja BERT dapat dilihat pada gambar 2.4 berikut.



Gambar 2. 4 Ilustrasi BERT

m. Keras Embedding

Keras Embedding Layers adalah lapisan untuk mengubah representasi kata menjadi vektor dengan ukuran tetap pada pemrosesan bahasa alami, di mana kata-kata harus digambarkan sebagai vektor untuk menjadi memasukkan ke lapisan *neural network* sebagaimana dijelaskan dalam (Ketkar, 2017).

n. Decoding-enhanced BERT with disentangled attention

Dengan menggunakan dua pendekatan baru, DeBERTa (*Decoding-enhanced BERT with disentangled attention*) meningkatkan model BERT dan RoBERTa. Yang pertama adalah mekanisme *disentangled attention*. Dalam mekanisme ini, setiap kata diwakili dengan dua vektor yang masing-masing mengkodekan konten dan posisinya. Selanjutnya, matriks terurai yang mengacu pada isi dan posisi relatifnya digunakan untuk menghitung bobot perhatian yang ada di antara kata-kata. Selanjutnya, untuk memprediksi *masked token* untuk pra-pelatihan model, *mask dekoder* yang disempurnakan digunakan untuk menggantikan lapisan keluaran sebagaimana dijelaskan dalam penelitian (He et al., n.d., 2023).

o. Robustly optimized BERT approach

RoBERTa (*Robustly optimized BERT approach*) juga dikenal sebagai Pendekatan BERT yang dioptimalkan dengan kuat, dibuat untuk meningkatkan kinerja model bahasa dengan mengubah *hyperparameter* utama dan pelatihan dengan *batch mini* dan tingkat pembelajaran yang lebih tinggi. RoBERTa menggunakan *tokenizer Byte-Pair-Encoding (BPE)* bertingkat *byte* dan berbagai

skema pelatihan, termasuk pengemasan kalimat dan masking dinamis sebagaimana dijelaskan dalam (Liu et al., 2019).

p. Word2Vec

Word2Vec digunakan untuk menangkap hubungan semantik antar kata, para peneliti di Google mengembangkan Word2Vec, yang merupakan metode pemrosesan bahasa alami atau biasa disebut NLP yang populer yang memungkinkan kata-kata diwakili sebagai vektor dalam ruang vektor kontinyu. Prinsip utama Word2Vec adalah bahwa kata-kata dengan arti yang sama harus memiliki representasi vektor yang serupa sebagaimana dijelaskan dalam penelitian (Mikolov et al., 2013).

q. FastText

FAIR, lab AI Research di Facebook, mengembangkan jenis penyematan kata yang disebut FastText. Ini didasarkan pada konsep penyematan subkata, yang berarti bahwa FastText memecah kata-kata menjadi bagian-bagian yang lebih kecil yang disebut karakter *n*-gram. Oleh karena itu, FastText dapat menangkap makna semantik dari kata-kata yang terkait secara morfologis, bahkan untuk kata-kata yang di luar kosakata atau langka. Ini sangat membantu dalam menangani bahasa yang memiliki morfologi yang kaya atau dalam tugas-tugas yang melibatkan kata-kata di luar kosakata sebagaimana dijelaskan dalam (Joulin et al., 2016).

r. GloVe

GloVe, akronim dari *Global Vectors for Word Representation*, membuat penyematan kata dengan menggunakan statistik kemunculan bersama kata. Pada dasarnya, tujuan GloVe adalah untuk mempelajari hubungan antara kata-kata dalam

korpus dengan menganalisis frekuensi kemunculan bersamanya. Konsep utama didasarkan pada pembuatan matriks kemunculan kata-kata, di mana nilai setiap sel menunjukkan seberapa sering dua kata yang berbeda muncul bersama dalam jendela konteks tertentu di seluruh korpus. Tidak seperti metode penyematan lainnya, GloVe mempertimbangkan apakah dua kata terjadi bersamaan dan seberapa mungkin muncul Bersama sebagaimana dijelaskan dalam (Pennington et al., 2014).

S. *Parallel dan Sequential Layer Neural Network*

Sebagaimana dijelaskan dalam penelitian (Guoyin & Hongbao, 1995) *Single Learner Neural Network* adalah konfigurasi layer yang setiap layer menjadi satu sumber saja untuk layer berikutnya pada jaringan *neural network*. *Parallel Learner Neural Network layer* adalah konfigurasi yang memungkinkan satu layer menjadi sumber bagi lebih dari satu layer berikutnya pada jaringan *neural network*. Implementasi pelatihan untuk konfigurasi *parallel* terstruktur modular, implementasi mudah, dan efisiensi tinggi jika dibandingkan dengan konfigurasi *Single Learner Neural Network*. Penelitian (Mohammed & Kora, 2023) menegaskan proses *ensemble learning* dengan konfigurasi *sequential* dan *parallel* bahwa kerangka kerja *ensemble learning* terdiri dari tiga jenis yaitu konfigurasi *baseline model parallel-sequential*, metode fusi, dan *heterogeneity*.

T. *Ensemble learning*

Ensemble learning adalah proses pembuatan model *machine learning* yang mengkombinasikan beberapa jenis algoritma untuk dilakukan dalam satu proses model. Kelebihan *ensemble learning* adalah kemampuan untuk saling melengkapi

dari beberapa algoritma dengan kelebihan masing-masing algoritma sebagaimana disebutkan dalam penelitian (Jadama et al., n.d., 2024) dan penelitian (Kazmaier & van Vuuren, 2022).

U. SMOTE

Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002) adalah teknik pengayaan data yang didesain untuk meminimalisir yang disebabkan oleh komposisi kelas dalam dataset *machine learning* yang tidak seimbang.

V. K-Fold

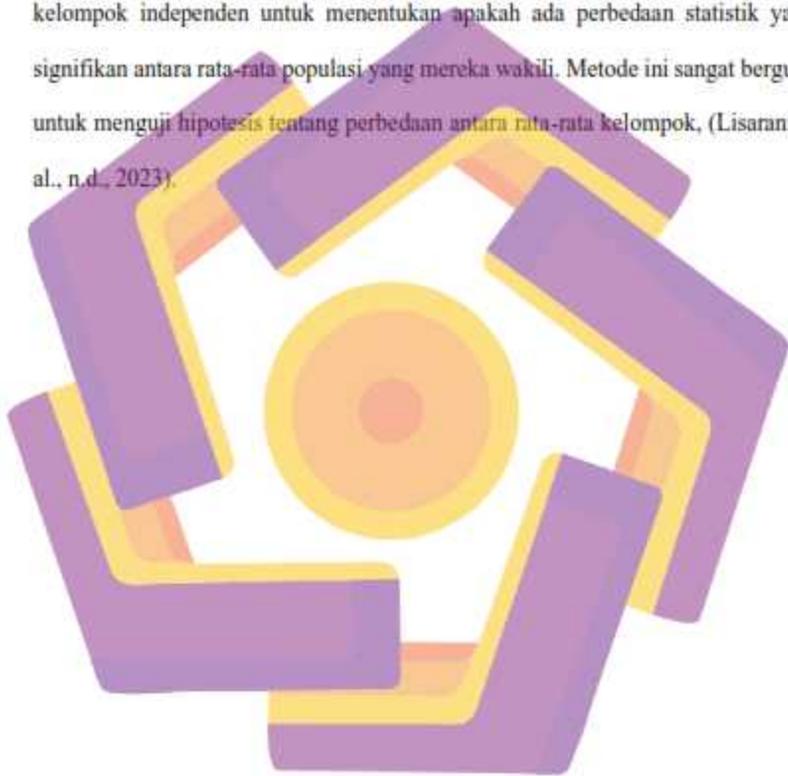
K-Fold Cross Validation (Teodorescu & Obreja Braşoveanu, 2025) adalah teknik statistik yang digunakan untuk mengevaluasi kinerja model *machine learning* dengan membagi dataset kedalam subset K dengan ukuran yang sama. Model dilatih pada K-satu lipatan dan diuji pada lipatan terakhir, proses ini diulang K kali, dengan set pengujian untuk setiap lipatan digunakan hanya sekali. Untuk memberikan perkiraan yang kuat tentang kemampuan generalisasi model, kinerjanya dirata-ratakan selama setiap iterasi K.

W. T-Test

Salah satu metode pengujian hipotesis adalah tes T, yang memerlukan penghitungan deviasi standar dan rata-rata dari sekumpulan data yang diambil dari populasi yang berdistribusi normal ketika simpangan baku populasi tidak diketahui. Tes ini termasuk alat analisis parametrik karena memerlukan penghitungan deviasi standar dan rata-rata populasi, (Lisarani et al., n.d., 2023).

X. One Way ANOVA

ANOVA Satu Arah, juga dikenal sebagai Analisis Varians, adalah teknik statistik yang digunakan untuk membandingkan rata-rata dari tiga atau lebih kelompok independen untuk menentukan apakah ada perbedaan statistik yang signifikan antara rata-rata populasi yang mereka wakili. Metode ini sangat berguna untuk menguji hipotesis tentang perbedaan antara rata-rata kelompok, (Lisarani et al., n.d., 2023).



BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini dilakukan dengan pendekatan kuantitatif – eksperimental. Tujuan dari penelitian ini adalah mendapatkan nilai akurasi klasifikasi terbaik dari proses percobaan konfigurasi neural network pada dataset *American Airlines Company*. Cara dan metode yang akan dijelaskan pada bagian selanjutnya. Hasil dari penelitian juga akan dibandingkan dengan hasil akurasi klasifikasi yang telah dilakukan peneliti sebelumnya.

3.2. Metode Pengumpulan Data

Penelitian diperlukan dataset yang digunakan oleh peneliti sebelumnya agar dapat dibandingkan performa akurasi atas konfigurasi *neural network* yang digunakan. Penelitian ini dilakukan menggunakan dataset yang diambil dari website www.kaggle.com dengan tema *American Airlines Company*. Dataset ini berisi ulasan konsumen maskapai penerbangan di amerika terkait layanan yang diberikan kepada pelanggan. Terdiri dari 14.000 tweet yang telah diberikan label dalam 3 (tiga) kelas yaitu positif, negative, dan netral yang komposisinya terlihat pada gambar 3.1 berikut.



Gambar 3. 1 Komposisi label dataset

Di situs web Kaggle, kumpulan data ini memiliki faktor kegunaan sebesar 8,24 dan memiliki Lisensi CC BY-NC-SA 4.0. Distribusi sentemennya adalah 62,69% negatif, 21,16% netral, dan 16,14% positif. Distribusi ini berdasarkan kecenderungan pelanggan untuk mengungkapkan kekecewaan terhadap layanan perusahaan di media sosial, sedangkan jika layanan telah memenuhi harapan pelanggan, layanan tersebut cenderung tidak terdengar dan dianggap sebagai sesuatu yang biasa terjadi.

3.3. Metode Analisis Data

Berdasarkan referensi penelitian terdahulu dan gap atau potensi penulis dalam melakukan penelitian lanjutan, ditemukan bahwa dalam proses klasifikasi analisis sentimen, algoritma BERT, BiLSTM, LSTM, dan GRU merupakan algoritma yang merupakan *State Of The Art* dalam proses klasifikasi sentimen. Selain itu, berdasarkan penelitian terdahulu dan penelitian sejenis, ditemukan

bahwa penelitian ini dilakukan untuk membandingkan hasil akurasi beberapa algoritma klasifikasi yang berbeda dan dilakukan dalam satu tahap dengan konfigurasi jaringan syaraf tiruan *single learner* pada tahap ini, penulis melihat potensi untuk melakukan penelitian lanjutan dengan dua skema, yaitu proses *single learner* dan proses *parallel learner*.

Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi sentiment analisis dalam tiga kelas untuk dataset *American Airlines Company*. Beberapa penelitian yang dilakukan oleh peneliti pendahulu telah terbit di jurnal yang dijadikan referensi dalam penelitian ini. Untuk menganalisa hasil dari penelitian ini dengan penelitian sebelumnya maka digunakan beberapa hal sebagai berikut :

A. Rumusan akurasi klasifikasi

Untuk dapat dibandingkan hasil penelitian ini dengan penelitian terdahulu diperlukan parameter standar. Dalam hal ini sebagaimana tertuang pada landasan teori, dalam proses klasifikasi sentiment analysis diperlukan besaran angka *accuracy* dari penelitian yang dilakukan.

B. Matriks literatur review dan posisi penelitian

Dalam proses untuk menentukan posisi penelitian yang dilakukan diperlukan sebuah table matrix posisi penelitian untuk mempermudah identifikasi posisi penelitian dan gap terhadap penelitian terdahulu yang telah ditampilkan pada bagian sebelumnya.

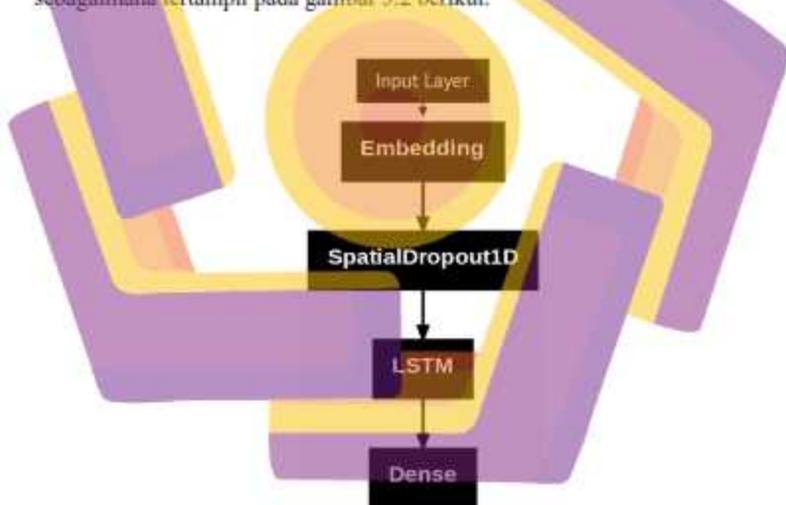
C. Tabel perbandingan akurasi dengan penelitian sebelumnya

Untuk mempermudah Analisa perbandingan hasil penelitian terhadap penelitian yang telah dilakukan sebelumnya diperlukan table

perbandingan akurasi hasil klasifikasi yang tertampil pada bagian hasil dan pembahasan.

3.4. Alur Penelitian

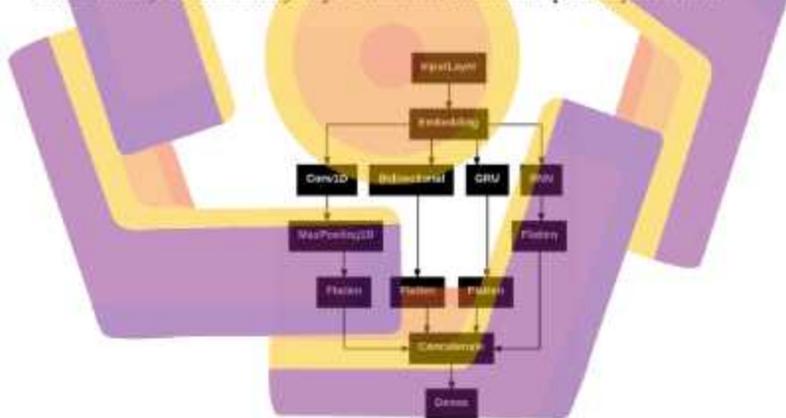
Penelitian dilakukan dalam dua tahap, tahap pertama adalah penelitian dengan skema *Single learner* dengan algoritma yang digunakan adalah LSTM, Bi-LSTM, dan GRU yang dikombinasikan dengan berbagai macam word embedding seperti Word2Vec, GloVe, FastText, BERT, Roberta, DeBERTa, dan Keras. Pada penelitian tahap pertama ini adalah *baseline* dari penelitian tahap 2 dimana hasil akurasi terbaik akan dibandingkan. Contoh konfigurasi model *Single Learner* sebagaimana tertampil pada gambar 3.2 berikut.



Gambar 3. 2 Contoh konfigurasi *Single Learner*

Pada skema penelitian kedua dilakukan kombinasi beberapa algoritma yang disusun secara parallel dan dikombinasikan dengan *word embedding* berdasarkan hasil pada penelitian tahap pertama. Hasil dari penelitian skema tahap kedua ini

akan dibandingkan akurasi yang baik untuk konfigurasi *Single Learner* maupun *Parallel Learner*. Konfigurasi *Parallel Learner* sebagaimana tertampil pada gambar 3.3. Proses penggabungan *concatenate* dilakukan setelah output algoritma yang disusun paralel untuk proses ekstraksi *feature* dilakukan proses *flatten*. Proses *flatten* dilakukan untuk mengubah dimensi output masing-masing proses ekstraksi *feature* yang menggunakan algoritma yang berbeda sehingga output sangat mungkin berbeda dimensi. Output dari proses *flatten* adalah vektor 1 (satu) dimensi yang akhirnya semua output yang telah diflatten memiliki dimensi yang sama. Output data yang telah di-flatten dilakukan *concatenate* sehingga seluruh data digabung menjadi vektor yang berukuran 1 x (total data yang dilakukan *concatenate*). Proses selanjutnya dilakukan klasifikasi pada layer *dense*.

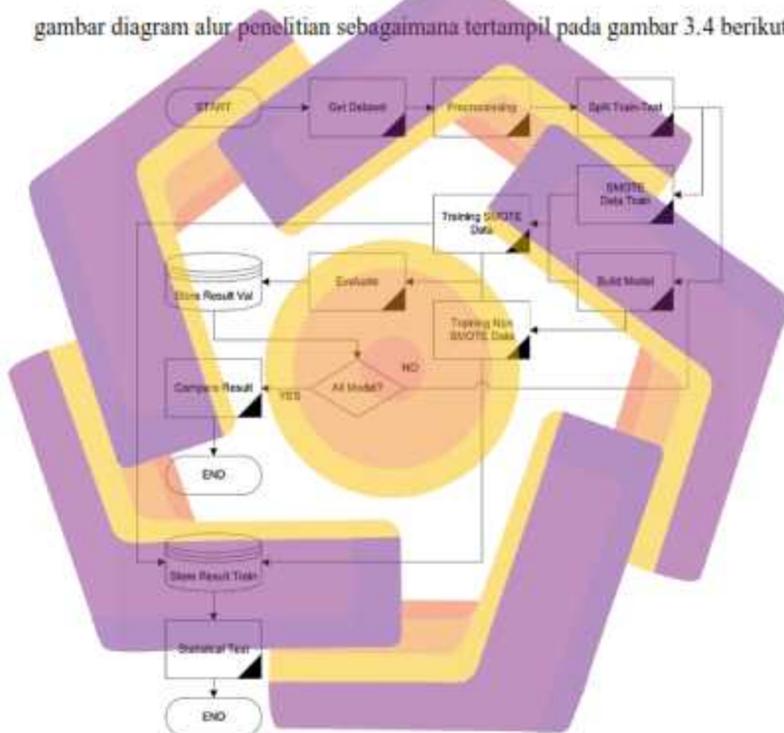


Gambar 3. 3 Contoh konfigurasi model *parallel learner*

Penelitian ini dilakukan dengan menggunakan Google Colab Pro dengan mesin Runtime T4 GPU High RAM dengan media penyimpanan Google Drive. Penulis memutuskan untuk menggunakan platform Google Colab Pro karena pada percobaan awal menggunakan versi gratis dan menemukan beberapa kendala,

antara lain keterbatasan compute unit yang tersedia, keterbatasan storage, keterbatasan memori, dan keterbatasan waktu untuk proses training, sehingga proses training sering terhenti di tengah-tengah proses percobaan. Setelah penulis berlangganan Google Colab Pro, penelitian dapat dilanjutkan dengan lebih lancar.

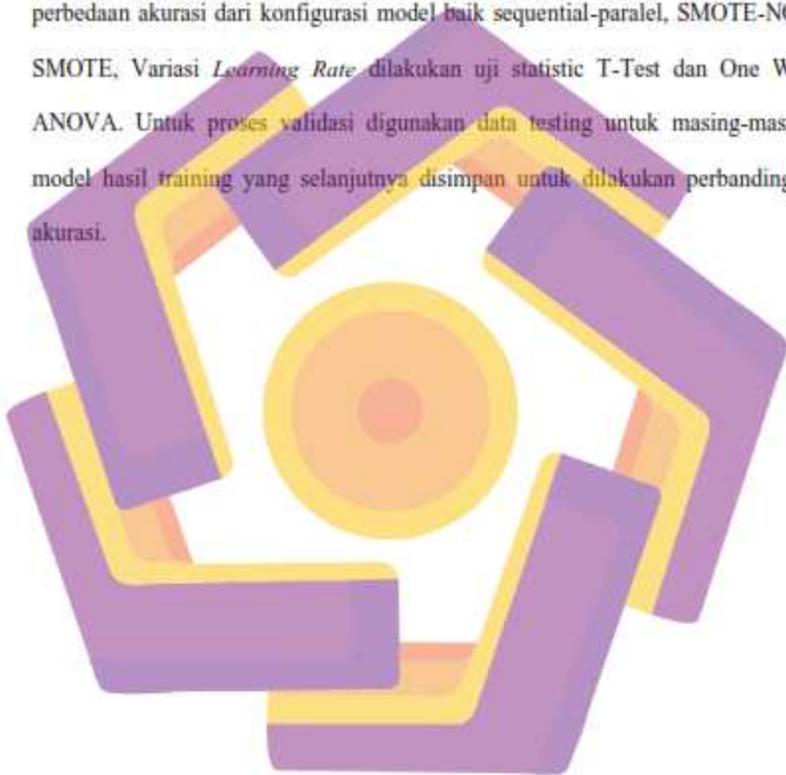
Untuk memperjelas langkah kerja dari penelitian ini dapat dilihat dalam gambar diagram alur penelitian sebagaimana tertampil pada gambar 3.4 berikut.



Gambar 3. 4 Diagram alir penelitian

Dari gambar 3.4 dapat dijelaskan bahwa alur penelitian dimulai dari mengambil dataset lalu dilakukan *preprocessing* agar didapatkan *clean text*. Setelah itu dilanjutkan proses *split data train-test* dengan komposisi 30% data test dan 70% *data train*. *Data train* lalu diproses *tokenize* sehingga didapatkan *data array numpy*

yang dilanjutkan proses SMOTE data *imbalance* sehingga didapatkan *data train* yang *balance*. Paralel dengan proses itu dibuat konfigurasi model yang akan ditraining. Proses training dilakukan dengan data *balance* dan *imbalance* yang hasil data training akan disimpan dan dibandingkan. Untuk menguji signifikansi perbedaan akurasi dari konfigurasi model baik sequential-paralel, SMOTE-NON SMOTE, Variasi *Learning Rate* dilakukan uji statistic T-Test dan One Way ANOVA. Untuk proses validasi digunakan *data testing* untuk masing-masing model hasil training yang selanjutnya disimpan untuk dilakukan perbandingan akurasi.



BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pengambilan Dataset

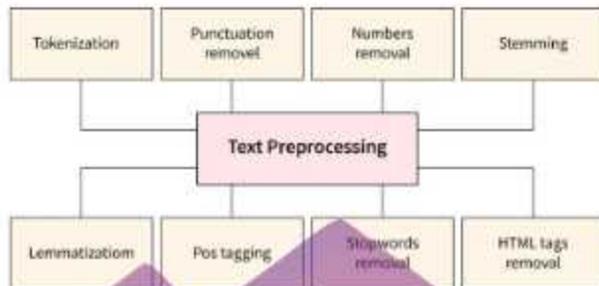
Proses pertama dalam tahapan penelitian adalah pengambilan dataset pada website www.kaggle.com. Untuk mendapatkan dataset yang digunakan dalam penelitian dapat dilakukan dengan membuat akun dahulu dan masuk ke halaman dataset sumber, selanjutnya di-klik tombol download. Dataset yang digunakan memiliki ekstensi .csv yang dapat digunakan untuk proses selanjutnya. Untuk memperjelas hasil dari alur ini sebagaimana tertampil pada gambar 4.1 berikut.



Gambar 4. 1 Website sumber dataset

4.2. Pre-Processing Text dan Tokenizing

Proses selanjutnya adalah *pre-processing text* untuk dataset yang telah didownload. Seluruh proses penelitian dilakukan pada platform Google Colab Pro dengan media penyimpanan Google Drive.



Gambar 4. 2 Gambar text *pre-processing*

Pada proses ini data text sebelum pre-processing masing terdapat karakter khusus seperti “@, *, !, #” sebagaimana gambaran yang tertampil pada gambar 4.2 sehingga perlu dilakukan proses *cleansing* untuk mendapatkan text yang sudah bersih. Proses yang paling penting pada tahap ini adalah tahapan tokenize dimana dataset yang berbentuk kata dipecah menjadi bagian-bagian kata. Hal ini sangat penting untuk proses selanjutnya .

4.3. Implementasi *Word Index*

Pada proses NLP *Word index* Adalah proses yang sangat penting, setiap kata yang telah melalui proses tokenize lalu dilakukan *index word*. Fungsi ini sangat penting karena model *machine learning* tidak bisa langsung memproses teks, melainkan memerlukan representasi numerik. Secara lebih detail proses *word index* adalah mengumpulkan kata unik dalam dataset lalu diberikan nomor urut (*index*) sehingga kedepan dapat diproses dengan mudah dan cepat. Pada proses *word index* ada proses *sequences* yaitu pemberian nomor urut index yang telah dibentuk untuk kata unik pada *word index*. Pada proses ini juga diberikan *pad_sequences* yang

berfungsi untuk menstandarkan panjang kata dari dataset yang akan diproses, jika panjang kata kurang dari standar maka akan diberikan index 0 (*unkwon word*) dan jika kelebihan maka akan dipotong. Dalam penelitian ini maksimal panjang kata diambil dari nilai panjang maksimal yang ada pada dataset sehingga tidak ada kata di dataset yang dipotong, untuk data yang lebih pendek akan diisi dengan index 0 (*unknown word*).

4.4. Split Train-Test Data

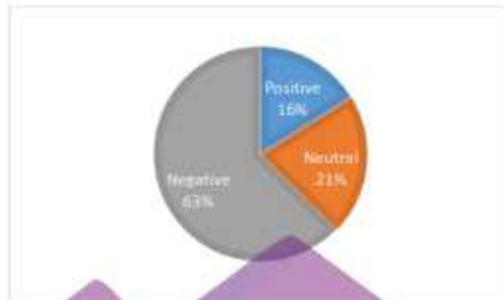
Pada proses ini data yang telah dilakukan *preprocessing* dan *word index* dilakukan *split data* untuk training dan testing. Untuk pembagiannya adalah 70% data training dan 30% data testing sebagaimana tertampil pada gambar 4.3.

```
X_train, X_test, y_train, y_test = train_test_split(model_inputs, labels, train_size=0.7, random_state=22)
X_train.shape
Out: (30000, 47)
```

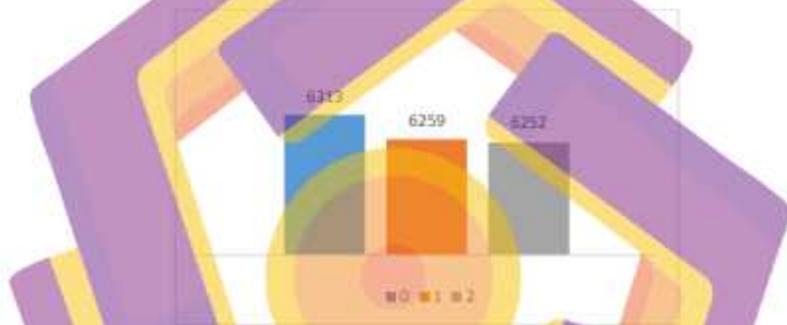
Gambar 4. 3 *Split data train-test*

4.5. Implementasi SMOTE

Proses percobaan menggunakan dataset yang memiliki kelas imbalance yang cukup besar sehingga dalam proses setelah dilakukan *split train – test data* dilakukan sintesis dataset dengan metode SMOTE sehingga didapatkan dataset yang lebih *balance*.



Gambar 4. 4 Distribusi Kelas Dataset sebelum Applying SMOTE



Gambar 4. 5 Distribusi kelas dataset setelah applying SMOTE

Komposisi awal dataset yang digunakan sebagaimana diilustrasikan pada gambar 4.4 terdiri dari 3 (tiga) kelas diantaranya kelas positif 16%, kelas netral 21%, dan kelas negative yang memiliki porsi 63%. Setelah dilakukan implementasi SMOTE maka kelas negative (0) tersusun atas 6313 data, kelas netral (1) tersusun atas 6259 data, dan kelas positif (2) dengan jumlah data 6252. Setelah dilakukan implementasi SMOTE maka komposisi antar kelas sentiment menjadi lebih *balance* sebagaimana diilustrasikan dalam gambar 4.5.

4.6. Build Model dan Instansi fungsi Word Embedding

Proses ini adalah membuat model dengan kombinasi algoritma dan susunan layer yang digunakan untuk proses selanjutnya. Untuk hasil dari *model summary* sebagaimana tertampil pada gambar 4.6

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
text_input [InputLayer]	(None, 34)	0	-
embedding [Embedding]	(None, 34, 32)	102,064	text_input[0][0]
bidirectional [Bidirectional]	(None, 34, 256)	168,864	embedding[0][0]
gru [GRU]	(None, 34, 128)	82,328	embedding[0][0]
numerical_input [InputLayer]	(None, 13)	0	-
flatten_2 [Flatten]	(None, 1088)	0	embedding[0][0]
flatten [Flatten]	(None, 8704)	0	bidirectional[0]-
flatten_1 [Flatten]	(None, 4352)	0	gru[0][0]
concatenate [Concatenate]	(None, 34157)	0	numerical_input_1, flatten_2[0][0], flatten[0][0], flatten_1[0][0]
dense [Dense]	(None, 7)	47,474	concatenate[0][0]

Total params: 629,818 (2,48 MB)
Trainable params: 629,818 (2,48 MB)
Non-trainable params: 0 (0,00 B)
None

Gambar 4. 6 Contoh model summary

Pada tahapan ini fungsi *word embedding* didefinisikan bersama dengan konfigurasi *learner* untuk proses ekstraksi *feature*. *Word embedding* adalah proses yang sangat penting dalam proses NLP dimana proses ini adalah merepresentasikan kata dalam vector numerik sehingga dataset text dapat diproses oleh komputasi komputer. Dataset yang telah diproses *word embedding* selanjutnya diproses pada tahap ekstraksi feature menggunakan algoritma *learner*, pada gambar 4.6 proses

ekstraksi *feature* algoritma Bi-LSTM dan GRU menghasilkan output vector dengan dimensi yang berbeda, Bi-LSTM berdimensi 34 x 256 sedangkan GRU berdimensi 34 x 128. Pada konfigurasi Bi-LSTM dan GRU karena dalam proses NLP adalah proses *sequence to sequence* maka diset *hyperparameter* pada Bi-LSTM dan GRU *return_sequences=True* sehingga input data yang berupa data *sequence* setiap output pada *hidden state* di setiap langkah waktu menjadi input pada langkah berikutnya.

Untuk dapat menggabungkan kedua output ini (*concatenate*) maka perlu dilakukan penyeragaman ukuran vector melalui proses *flatten* sehingga dihasilkan vector 1 (satu) dimensi. Setelah proses *flatten* dilakukan penggabungan atau proses *concatenate* vector 1 dimensi dari masing-masing output algoritma *learner* dimana output hasil *concatenate* adalah vector 1 dimensi dengan panjang sejumlah 3 output vector hasil *flatten*. Pada tahap ini output dari *concatenate* merupakan gabungan dari proses ekstraksi *feature* oleh algoritma *learner* yang sudah digabung dan siap masuk pada tahapan *dense* untuk proses klasifikasi.

4.7. Training

Tahap ini adalah proses training data dengan model yang telah disusun sebelumnya. Contoh Proses training dalam penelitian dapat dilihat dalam gambar 4.7 berikut.

```

model.compile(
    optimizer='adam', # Use 'Adam' optimizer
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])
}
metrics_total = []
# create learning curves for different learning rates
learning_rates = [1E-8, 1E-1, 1E-2, 1E-3, 1E-4, 1E-5, 1E-6, 1E-7]
#learning_rates = [1E-8, 1E-1, 1E-2]
for f in range(len(learning_rates)):
    # determine the plot number
    plot_no = 100 + (f*4)
    pyplot.subplot(plot_no)
    optimizer = tf.keras.optimizers.Adam(learning_rate=learning_rates[f]) # Use the learning rate from the list
    # 5-Fold Cross Validation
    kf = StratifiedKFold(n_splits=5)
    kf.get_n_splits(X_train, y_train, samples)

    metrics = []

    for train_index, test_index in kf.split(X_train, y_train, samples):
        epochs = 100
        batch_size = 32
        history = model.fit(
            X_train[train_index], y_train[train_index],
            X_train[test_index], y_train[test_index],
            validation_split=0.2,
            batch_size=batch_size,
            epochs=epochs,
            callbacks=[tf.keras.callbacks.EarlyStopping,
                       tf.keras.callbacks.ReduceLROnPlateau],
            monitor='val_loss',
            patience=3,
            restore_best_weights=True,
            verbose=1)
        metrics.append(history.history['accuracy'][-1])
    metrics_total.append(history.history['accuracy'][-1])

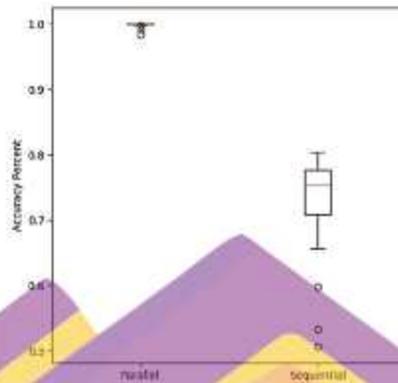
```

Gambar 4. 7 Proses training data

Pada proses training diimplementasikan *K-Fold Cross Validation* dengan nilai `n_splits = 5`, dengan jumlah epoch 100, batch 32. Untuk *Learning Rate* yang digunakan Adalah antara 1 dan 0.0000001.

4.8. Evaluation

Proses ini adalah *evaluasi terhadap model yang dibuat dari proses training*. Sebagaimana tertampil pada gambar 4.8 model dilakukan evaluasi dan validasi dengan data testing. Selanjutnya ditampilkan dengan metode *accuracy table* dan *confusion matrix*. Untuk membuktikan signifikansi akurasi masing-masing konfigurasi dilakukan uji T-Test maupun One Way ANOVA.



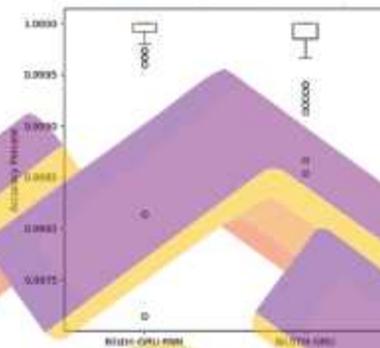
Gambar 4. 8 Perbandingan T-Test Konfigurasi *Single Learner -Parallel Learner*
 Gambar 4.8 mengilustrasikan perbandingan hasil T-Test untuk konfigurasi *Single Learner* dan *parallel Learner*. Hasil akurasi konfigurasi *parallel Learner* memiliki nilai yang lebih tinggi dengan variansi yang lebih rendah.

Tabel 4. 1 T-Test Konfigurasi *Single Learner – Parallel Learner*

No	Model Configuration	Mean Accuracy	STD Accuracy
1	BiLSTM (<i>Single Learner</i>)	0.7328	0.06293
2	BiLSTM-GRU-RNN	0.9998	0.00285
T-Test Value		26.42	
P-Value		0.000000005	

Pada table 4.1 dibandingkan akurasi rata-rata, deviasi standar, nilai T-Test, dan nilai P untuk dua kelompok data, BiLSTM (*single learner*) dan BiLSTM-GRU-RNN. Dalam eksperimen T-Test konfigurasi *Single Learner - Parallel Learner*, akurasi rata-rata dari konfigurasi *single learner* adalah 0.7328, dan deviasi standar untuk akurasi adalah 0.06293, sedangkan untuk konfigurasi *parallel learner* dari algoritma BiLSTM-GRU-RNN, akurasinya adalah 0.9924 dengan deviasi standar 0.00285. Dalam proses perhitungan, nilai T adalah 26.42, dan nilai P adalah 0.000000005. Dari hasil ini, dapat dijelaskan bahwa nilai P lebih kecil dari 0.05,

sehingga kita dapat menolak hipotesis nol, yang menunjukkan perbedaan hasil yang cukup besar antara kedua kelompok data. Nilai P yang lebih besar menunjukkan perbedaan yang lebih besar antara kedua kelompok data.



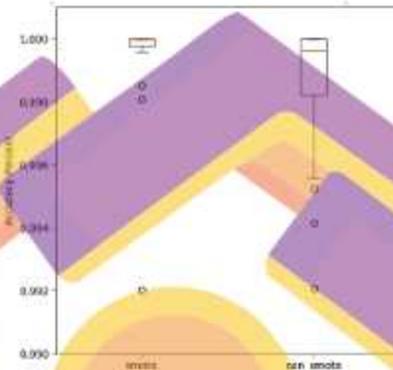
Gambar 4. 9 Perbandingan T-Test BiLSTM-GRU-RNN vs BiLSTM-GRU

Tabel 4. 2 T-Test BiLSTM – GRU vs BiLSTM-GRU-RNN

No	Model Configuration	Mean Accuracy	STD Accuracy
1	BiLSTM-GRU	0.9998	0.00374
2	BiLSTM-GRU-RNN	0.9998	0.00285
T-Test			0.2249
P-Value			0.8225

Tabel 4.2 diperlihatkan Percobaan T-Test dari konfigurasi BiLSTM-GRU-RNN dibandingkan dengan BiLSTM-GRU menurut metode studi *ablation* untuk menentukan pengaruh komponen jaringan saraf terhadap model yang dilatih, diperoleh nilai T yang kecil sebesar 0.2249, menunjukkan bahwa tidak ada perbedaan signifikan dalam hasil akurasi dari model yang dibandingkan. Nilai P juga di atas 0.05, menunjukkan bahwa data akurasi yang dihasilkan tidak berbeda secara signifikan. Gambar 4.9 menunjukkan bahwa konfigurasi BiLSTM-GRU

RNN vs. BiLSTM-GRU tidak memiliki perbedaan signifikan dalam akurasi. Dengan prinsip studi ablation, peningkatan jumlah lapisan tidak secara signifikan mempengaruhi akurasi. Namun, data di atas menunjukkan bahwa peningkatan jumlah lapisan paralel mengurangi varians data akurasi.

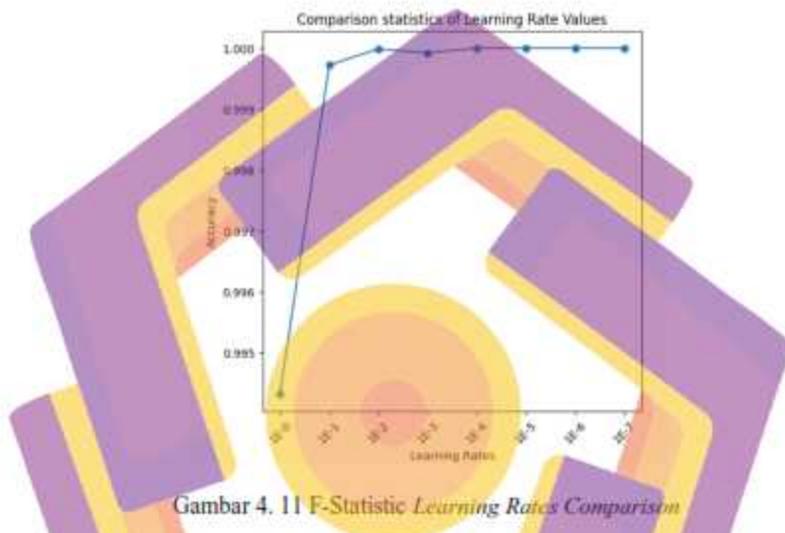


Gambar 4. 10 Perbandingan T-Test SMOTE dan Non-SMOTE

Implementasi SMOTE pada data pelatihan untuk mencapai keseimbangan dalam jumlah kelas data pelatihan tidak menunjukkan perubahan yang signifikan dalam akurasi, tetapi implementasi SMOTE dapat mengurangi varians data akurasi pelatihan, seperti yang ditunjukkan pada Gambar 4.10. Tabel 4.3 merangkum hasil percobaan untuk mengukur pengaruh SMOTE terhadap akurasi klasifikasi. Akurasi klasifikasi dengan implementasi SMOTE sedikit lebih baik daripada akurasi klasifikasi non-SMOTE. Hal ini dibuktikan dengan nilai T-TEST yang kecil yaitu 1.4881 dan nilai P yang lebih besar dari 0.05, yaitu 0.1407. Ini mungkin terjadi karena proses pelatihan telah menerapkan metode *K-Fold Cross Validation* sehingga kelemahan dari dataset yang tidak seimbang dapat diatasi.

Tabel 4. 3 Perbandingan T-Test SMOTE vs NON-SMOTE

No	Model Configuration	Mean Accuracy	STD Accuracy
1	SMOTE	0.99924	0.00285
2	NON-SMOTE	0.99813	0.00369
	T-Test		1.4881
	P-Value		0.1407



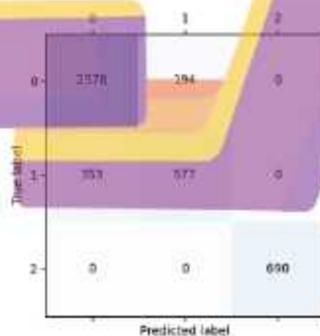
Gambar 4. 11 F-Statistic Learning Rates Comparison

Sebuah eksperimen tambahan dilakukan untuk menentukan pengaruh *learning rate* terhadap akurasi klasifikasi. Algoritma BiLSTM-GRU-RNN digunakan dalam eksperimen ini. ANOVA satu arah digunakan untuk membandingkan statistik akurasi dari setiap laju pembelajaran, yang berkisar dari 1 hingga 0.00000001. Eksperimen ini menemukan bahwa *learning rate* yang lebih tinggi menghasilkan akurasi yang lebih rendah dengan variasi data yang lebih besar. Seiring dengan penurunan nilai *learning rate*, akurasi meningkat dan variasi menurun, seperti yang divisualisasikan dalam Gambar 4.11.

Tabel 4. 4 *Classification Report*

	Precision	Recall	F1-Score	Support
0	0.88	0.93	0.90	2772
1	0.75	0.62	0.68	930
2	1.00	1.00	1.00	690
Accuracy			0.88	4392
Macro AVG	0.88	0.85	0.86	4392
Weighted AVG	0,87	0,88	0,87	4392

Dari laporan klasifikasi yang divisualisasikan dalam Tabel 4.4, validasi data uji memperoleh akurasi sebesar 0,88 dengan nilai F1-score sebesar 0,9 untuk kelas negatif (0), sedangkan untuk kelas netral (1) adalah 0,68, dan untuk kelas positif (2) adalah 1,00. Ini menunjukkan bahwa klasifikasi untuk kelas positif dan negatif telah divalidasi dengan baik, sementara kelas netral memiliki nilai F1-score terkecil.

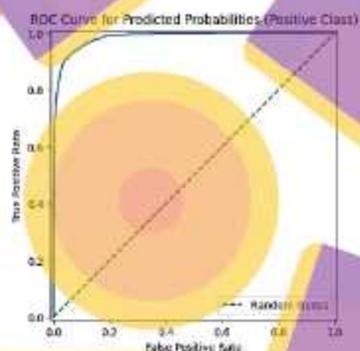


Gambar 4. 12 *Confusion Matrix*

Hasil validasi model juga ditampilkan dalam matriks kebingungan pada Gambar 4.12 untuk menyajikan data. Prediksi negatif dan kelas negatif yang

sebenarnya memiliki nilai terbesar, 2578. Ada 690 prediksi positif dengan nilai aktual positif. Sementara itu, terdapat 577 prediksi netral dengan nilai aktual netral. Kesalahan prediksi terbesar ditemukan dalam 353 prediksi negatif dengan nilai aktual netral.

Proses validasi juga ditunjukkan dalam kurva ROC pada Gambar 4.13 di mana Tingkat Positif Benar menghasilkan kurva di atas batas garis tebakan acak mendekati angka 1 sehingga dapat disimpulkan bahwa akurasi model yang dihasilkan memiliki kinerja yang baik.



Gambar 4.13 ROC Curve Validation

4.9. Store Result

Dari skema pertama *Single Learner Neural Network* direkap dalam sebuah table sebagaimana tertampil pada table 4.5 berikut.

Tabel 4.5 Rekap hasil percobaan *Single Learner Neural Network*

No	Model Configuration	Layar Type	Embedding	Precision	Recall	F1-Score	Accuracy
1	BiLSTM	Single Learner	Keras	0.8	0.8	0.8	80.20%
2	GRU	Single Learner	Keras	0.79	0.79	0.79	79.50%
3	LSTM	Single Learner	Keras	0.79	0.8	0.79	79.80%
4	LSTM	Single Learner	WordVec Google News 200	0.77	0.77	0.77	76.80%
5	LSTM	Single Learner	GloVe Twitter 200	0.76	0.77	0.74	76.70%
6	LSTM	Single Learner	RoBERTa	0.71	0.71	0.69	71.00%
7	LSTM	Single Learner	BERT Large Uncased	0.71	0.71	0.69	71.70%
8	LSTM	Single Learner	FastText Subword 300	0.71	0.72	0.67	71.00%
9	LSTM	Single Learner	DeBERTa V3 Base Pnnl Base	0.69	0.71	0.69	71.00%
10	LSTM	Single Learner	WordVec Stanford Corpus Dataset	0.64	0.66	0.62	68.20%

Sedangkan skema kedua *parallel learner neural network* direkap dalam sebuah table sebagaimana tertampil pada table 4.6 berikut.

Tabel 4. 6 Tabel rekap hasil percobaan *parallel learner*

No	Model Configuration	Layer Type	Embedding	Precision	Recall	F1-Score	Accuracy
1	Paralel GRU-RNN	Paralel Learner	Keras	0,87	0,89	0,87	87,58%
2	Paralel GRU-RNN	Paralel Learner	BERT Large Uncase	0,87	0,87	0,87	87,40%
3	Paralel GRU-BiLSTM-CNN	Paralel Learner	Keras	0,87	0,87	0,87	87,58%
4	Paralel GRU-BiLSTM-RNN	Paralel Learner	Keras	0,89	0,89	0,89	89,09%
5	Paralel GRU-SimpleRNN-CNN	Paralel Learner	BERT Large UnCase	0,86	0,86	0,86	86,20%
6	Paralel GRU-SimpleRNN-RNN	Paralel Learner	BERT Large UnCase	0,87	0,87	0,87	87,58%
7	Paralel GRU-BiLSTM-SimpleRNN-CNN	Paralel Learner	Keras	0,86	0,86	0,87	86,19%
8	Paralel GRU-BiLSTM-SimpleRNN-RNN	Paralel Learner	Keras	0,87	0,89	0,87	87,89%
9	Paralel GRU-LSTM-SimpleRNN-CNN	Paralel Learner	Jakarta large model	0,86	0,87	0,86	87,70%
10	Paralel GRU-LSTM-SimpleRNN-RNN	Paralel Learner	Jakarta large model	0,87	0,87	0,87	87,58%
11	Paralel GRU-LSTM-SimpleRNN-CNN	Paralel Learner	BERT Large UnCase	0,87	0,87	0,87	87,58%
12	Paralel GRU-LSTM-SimpleRNN-RNN	Paralel Learner	DistilBERT-DistilUnCase-L-16	0,79	0,81	0,78	80,00%

4.10. Compare Result

Hasil terbaik dari penelitian ini dibandingkan dengan hasil penelitian sebelumnya yang tertampil pada tabel 4.7 berikut.

Tabel 4. 7 Tabel rekap perbandingan dengan penelitian terdahulu

No	Penelitian	Algoritma	Hasil Akurasi
1	(Patel et al., 2022)	BERT	83%
2	(Vadivukarassi et al., 2018)	Random Forest Classifier	81,35%
3	(Saad, 2020)	Support Vector Machine	83,81%
4	(Rustam et al., 2019)	Linier Regression + SGDC	79,40%
5	(Rane & Kumar, 2018)	Ada Boost	84,50%
6	(Alshamsi et al., 2020)	Naïve Bayes	76,10%
7	Penelitian ini	3 Paralel Layer GRU-BiLSTM-RNN	88%

4.11. Pembahasan

Dari hasil penelitian seperti yang ditunjukkan pada Tabel 4.5, algoritma LSTM dengan kombinasi *Keras Embedding* menghasilkan akurasi tertinggi yaitu 79,90%, disusul Word2Vec Google News 300 dengan akurasi 76,90%, kemudian GloVe Twitter 200 dengan akurasi 76,70%, RoBERTa menghasilkan akurasi 72,90%, disusul BERT Large Uncased dengan akurasi 72,70%, disusul FasText SubWord 300 dengan akurasi 71,60%, disusul DeBERTa V3 Base Fine Tuned dengan akurasi 71,40%. Akurasi terendah adalah embedding Word2Vec Selftrain Corpus Dataset yang menghasilkan akurasi sebesar 68,20%. Hal ini dimungkinkan karena keterbatasan database kata yang digunakan, yaitu dataset itu sendiri, jika dibandingkan dengan *Word Embedding* lain yang telah dilatih pada jumlah kata yang jauh lebih banyak. Dari data ini dapat dilihat bahwa algoritma LSTM dengan kombinasi keras embedding menghasilkan akurasi tertinggi sehingga kombinasi *keras embedding* menjadi *baseline* untuk konfigurasi *single learner* lainnya dengan algoritma yang berbeda.

Pada skema konfigurasi pertama, terlihat bahwa algoritma Bi-LSTM dengan *Keras Embedding* menghasilkan akurasi tertinggi sebesar 80,20%. Hasil ini mengungguli algoritma GRU yang menghasilkan akurasi sebesar 79,30% dan algoritma LSTM yang menghasilkan akurasi sebesar 79,90%.

Algoritma Bi-LSTM dan Keras menghasilkan akurasi tertinggi pada skema pertama ini, selanjutnya algoritma dan word embedding ini akan menjadi acuan atau *baseline* untuk skema kedua, yaitu skema *parallel learner*, dengan menggabungkan algoritma lainnya.

Pada skema penelitian *parallel learner neural network*, hasil terbaik pada konfigurasi *single learner neural network* menjadi tolok ukur untuk melakukan eksperimen dengan menambahkan algoritma lain dan menyusunnya secara paralel. Seperti yang terlihat pada Tabel 4.6, kombinasi algoritma GRU-BiLSTM-RNN dengan *keras embedding* yang disusun dalam 3 (tiga) *parallel learner neural network* menghasilkan akurasi terbaik sebesar 88%. Diikuti oleh konfigurasi 4 *parallel learner* GRU-BiLSTM-SimpleRNN-RNN dengan *keras embedding* yang menghasilkan akurasi sebesar 87,80%. Akurasi terbaik ketiga adalah konfigurasi 3 *parallel learner* GRU-BiLSTM-CNN dengan *Keras Embedding* yang menghasilkan akurasi sebesar 87,50%. Diikuti oleh konfigurasi 3 *parallel learner* GRU-SimpleRNN-CNN dengan BERT *Large Uncased Embedding* dengan akurasi sebesar 86,20%.

Skema 4 lapisan *parallel learner* GRU-BiLSTM-SimpleRNN-CNN dengan *Keras Embedding* menghasilkan akurasi sebesar 86,10%. Diikuti oleh skema 4 *parallel learner*, GRU-LSTM-SimpleRNN-CNN dengan *deberta-large-mnli embedding* menghasilkan akurasi sebesar 85,70%. Akurasi berikutnya diikuti dengan skema 4 *parallel learner* GRU-LSTM-SimpleRNN-CNN dengan RoBERTa Base Open AI menghasilkan akurasi sebesar 85,50%. Diikuti dengan skema 3 *parallel learner* GRU-SimpleRNN-CNN dengan BERT *Large Uncased* menghasilkan akurasi sebesar 85,50%, diikuti dengan 2 *parallel learner* GRU-RNN dengan BERT *Large Uncased Embedding* menghasilkan akurasi sebesar 85,40%. Akurasi selanjutnya dicapai oleh skema GRU-LSTM-SimpleRNN-CNN 4 *parallel learner* dengan BERT *Large Uncased* dengan akurasi sebesar 85,30%. Untuk

akurasi terkecil, konfigurasi GRU-LSTM-SimpleRNN-CNN 4 *parallel-learner* dengan embedding DeepSeek-R1-Distill-Qwen1.5B menghasilkan akurasi sebesar 80,80%.

Dari perbandingan *Word Embedding* yang digunakan, 3 (tiga) akurasi teratas adalah *Keras Embedding*, dan akurasi BERT Embedding terbaik dicapai pada konfigurasi GRU-SimpleRNN-CNN 3 *parallel learner* dengan akurasi sebesar 86,20%.

Jika dibandingkan pada Tabel 4.3, hasil penelitian ini memiliki akurasi sebesar 88%, yang mengungguli hasil penelitian sebelumnya, disusul oleh penelitian (Rane & Kumar, 2018) dengan algoritma AdaBoost, dengan akurasi sebesar 84,5%, kemudian penelitian (Saad, 2020) dengan algoritma *Support Vector Machine* dengan akurasi sebesar 83,81%, kemudian disusul oleh (Patel et al., 2022) dengan algoritma BERT yang menghasilkan akurasi sebesar 83,00%.

Jika dibandingkan antara konfigurasi *single learner neural network* dan *parallel learner neural network*, konfigurasi *parallel learner* mengungguli semua konfigurasi *single learner*. Pada konfigurasi *single learner*, akurasi terbaiknya adalah 80,20%, sedangkan pada skema *parallel learner*, akurasi terendahnya adalah 80,80%, masih lebih unggul dari hasil terbaik pada konfigurasi *single learner*.

Dalam proses penelitian mengenai pengaruh komponen jaringan saraf dengan menggunakan metode studi ablasi, ditemukan bahwa konfigurasi jaringan paralel meningkatkan akurasi klasifikasi. Ketika jumlah konfigurasi jaringan paralel dikurangi dari 3 menjadi 2, tidak menunjukkan efek yang signifikan.

Demikian juga, penerapan SMOTE tidak menghasilkan efek yang signifikan karena proses pelatihan telah menerapkan K-Fold Cross Validation, yang mengatasi kelemahan ketidakseimbangan dataset. Hasil eksperimen dari nilai *learning rate* menunjukkan bahwa semakin kecil nilai laju pembelajaran, semakin baik akurasi dengan nilai varians yang lebih kecil. Proses validasi dari hasil pelatihan dengan laporan klasifikasi, matriks kebingungan, dan ROC Curve memvalidasi akurasi hasil pelatihan dan model memiliki kinerja yang baik.

Proses *parallel ensemble learning* 3 Paralel Layer GRU-BiLSTM-RNN dapat memberikan akurasi yang lebih baik karena dalam proses ekstraksi *feature* algoritma masing-masing memiliki kelebihan dan masing-masing kelebihan itu digabungkan dengan proses *concatenate* sebelum masuk ke *layer dense* sehingga *feature* yang menjadi input *dense* menjadi lebih kaya. Algoritma RNN memiliki kelebihan efisiensi komputasi tinggi untuk teks pendek sehingga RNN dapat menangkap informasi linier sederhana, namun memiliki kelemahan *vanishing gradient* untuk teks panjang. Sehingga peran utama pada konfigurasi model ini RNN digunakan untuk menangani teks yang berukuran pendek. Peran Bi-LSTM adalah untuk menangkap informasi dua arah, mekanisme *gates* (*input*, *forget*, dan *output*) memungkinkan model untuk memilih informasi mana yang dipertahankan dan dibuang dalam jangka panjang. GRU memiliki kelebihan struktur yang lebih sederhana dari Bi-LSTM yang bisa memberikan performa yang hampir serupa dengan Bi-LSTM namun dengan parameter yang lebih sedikit. Karena kesederhanaannya GRU dapat lebih cepat dalam proses pembelajaran atau *training*. Kelebihan lain pada GRU adalah efektif dalam menangkap informasi dependensi

jangka panjang dengan resiko *overfitting* yang lebih rendah. Kombinasi dari ketiga algoritma tersebut digabungkan menghasilkan model yang dapat mengambil informasi teks pendek secara menyeluruh oleh RNN, diperkaya kemampuan untuk mengambil informasi jangka panjang dan jangka pendek secara dua arah oleh Bi-LSTM dan reduksi bias oleh GRU sehingga menghindari *overfitting*.

Keras embedding menghasilkan akurasi klasifikasi sentiment analysis yang lebih baik karena sifatnya yang *task spesifik*. Word embedding BERT, Word2Vec, GloVe dan yang lainnya ditrain pada korpus umum seperti dataset Wikipedia. Sedangkan Keras ditruining secara spesifik bersamaan dengan proses training model sehingga dapat lebih adaptif terhadap dataset teks yang digunakan.



BAB V

PENUTUP

5.1. Kesimpulan

Pada hasil percobaan dengan konfigurasi *single learner* maupun konfigurasi *parallel learner* dengan kombinasi Keras Embedding menghasilkan akurasi terbaik yaitu 80,80% pada konfigurasi Sequential BiLSTM-Keras. Akurasi 88,00% pada konfigurasi GRU-BiLSTM-RNN dengan Keras Embedding. Sehingga pada penelitian ini pengaruh penggunaan Keras Embedding memberikan akurasi terbaik. Untuk algoritma yang digunakan dalam penelitian, BiLSTM menghasilkan akurasi terbaik pada konfigurasi *single learner* dan pada proses *parallel learner neural network* algoritma GRU dapat memperkaya proses ekstraksi *feature* dari dataset yang digunakan sebelum dilakukan klasifikasi pada lapisan *dense*.

Berdasarkan hasil Analisa dan pembahasan hasil penelitian pada proses konfigurasi *single learner* penggunaan algoritma BiLSTM-Keras Embedding menghasilkan akurasi data validasi terbaik dengan nilai 80,80%. Selanjutnya pada konfigurasi *parallel learner* GRU-BiLSTM-RNN dengan Keras Embedding didapatkan akurasi data validasi 88%. Hasil ini menunjukkan konfigurasi *parallel learner neural network* dapat meningkatkan akurasi klasifikasi yang signifikan. Hal ini diperkuat dengan hasil Uji T-Test data akurasi training dengan nilai P kritis sebesar 0,05. Hasil uji T-Test menghasilkan nilai P : 0,5E9 yang jauh lebih kecil dari nilai P kritis dan T sebesar 26,42 sehingga dapat terkonfirmasi secara uji statistic rata-rata akurasi berbeda secara signifikan.

Pada percobaan juga dilakukan uji hasil akurasi konfigurasi *parallel learner* GRU-BiLSTM-RNN dengan Keras Embedding dengan merubah sumber data training yang dilakukan SMOTE-Non SMOTE. Hasilnya didapatkan akurasi data training yang tidak berbeda signifikan berdasarkan Uji T-Test dengan nilai kritis 0,05. Hasil nilai P Adalah 0,14 yang lebih besar dari 0,05 dan nilai T 1,48 sehingga secara uji statistic terkonfirmasi bahwa akurasi yang dihasilkan tidak berbeda secara signifikan. Proses SMOTE dan Non SMOTE pada peneitian ini tidak berpengaruh signifikan.

Pada percobaan juga dilakukan uji hasil akurasi konfigurasi *parallel learner* GRU-BiLSTM-RNN dengan Keras Embedding juga dilakukan perubahan nilai learning rate antara nilai 1 sampai dengan 0,1E7 dengan hasil pada penelitian ini nilai learning rate yang semakin kecil menghasilkan akurasi yang semakin baik.

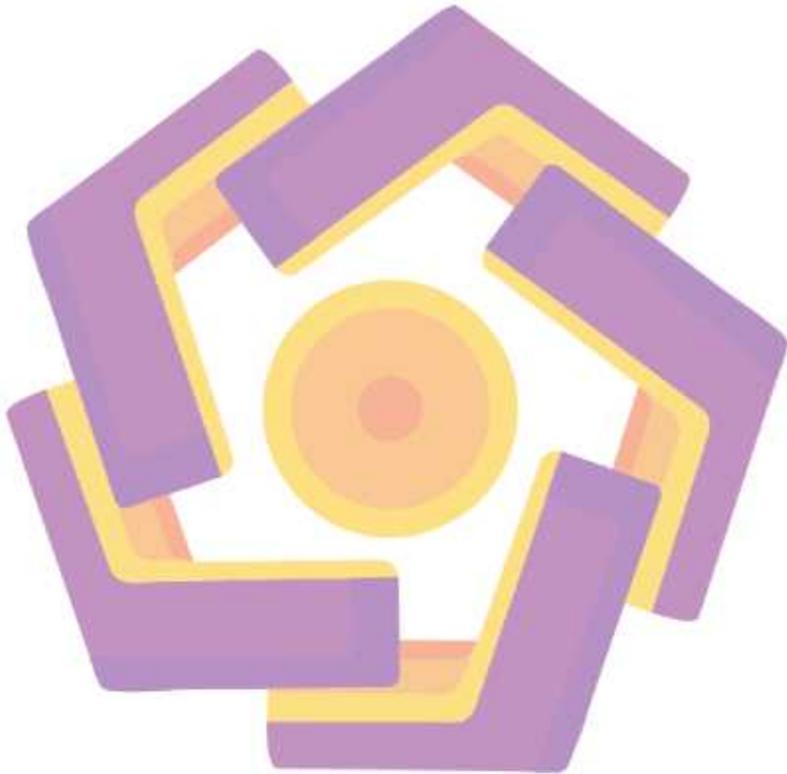
Hasil terbaik penelitian ini juga dibandingkan dengan penelitian sebelumnya, dari 6 (enam) penelitian sebelumnya penelitian ini unggul dalam hal akurasi dengan nilai akurasi data validasi sebesar 88,0% dengan konfigurasi *parallel learner* GRU-BiLSTM-RNN dengan Keras Embedding.

5.2. Saran

Berikut adalah beberapa saran yang dapat dilakukan untuk penelitian selanjutnya :

1. Peneliti selanjutnya dapat mengeksplorasi kombinasi algoritma yang digunakan untuk mendapatkan akurasi yang lebih baik.
2. Peneliti selanjutnya dapat mencoba kombinasi *word embedding* yang digunakan untuk mendapatkan akurasi yang lebih baik.

3. Pada proses penelitian ini memiliki kelemahan beban komputasi yang berat sehingga penelitian selanjutnya dapat mempertimbangkan keefektifan beban komputasi yang dibutuhkan.



DAFTAR PUSTAKA

- Aakash, Gupta, S., & Noliya, A. (2024). URL-Based Sentiment Analysis of Product Reviews Using LSTM and GRU. *Procedia Computer Science*, 235, 1814–1823. <https://doi.org/10.1016/j.procs.2024.04.172>
- Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P. (2019). The impact of features extraction on the sentiment analysis. *Procedia Computer Science*, 152, 341–348. <https://doi.org/10.1016/j.procs.2019.05.008>
- Almeida, F., & Xexéo, G. (2019). *Word Embeddings: A Survey*. <http://arxiv.org/abs/1901.09069>
- Alshamsi, A., Bayari, R., & Salloum, S. (2020). Sentiment analysis in English Texts. *Advances in Science, Technology and Engineering Systems*, 5(6), 1638–1689. <https://doi.org/10.25046/AJ0506200>
- Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3(6), 1137–1155. <https://doi.org/10.1162/153244303322533223>
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). *Enriching Word Vectors with Subword Information*. <http://arxiv.org/abs/1607.04606>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. In *Journal of Artificial Intelligence Research* (Vol. 16).
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. <http://arxiv.org/abs/1412.3555>

- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (n.d.). *Indexing by Latent Semantic Analysis*.
- Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (n.d.). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. <https://github.com/tensorflow/tensor2tensor>
- Dharma, A. S., & Saragih, Y. G. R. (2022). Comparison of Feature Extraction Methods on Sentiment Analysis in Hotel Reviews. *Sinkron*, 7(4), 2349–2354. <https://doi.org/10.33395/sinkron.v7i4.11706>
- F.Y. O., J.E.T. A., O. A., J. O. H., O. O., & J. A. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, 48(3), 128–138. <https://doi.org/10.14445/22312803/IJCTT-V48P126>
- Ghosh, S., Ekbal, A., & Bhattacharyya, P. (2023). Chapter 2 - Natural language processing and sentiment analysis: perspectives from computational intelligence. In D. Das, A. K. Kolya, A. Basu, & S. Sarkar (Eds.), *Computational Intelligence Applications for Text and Sentiment Data Analysis* (pp. 17–47). Academic Press. <https://doi.org/https://doi.org/10.1016/B978-0-32-390535-0.00007-0>
- Girshick, R. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>

- Guoyin, W., & Hongbao, S. (1995). Parallel neural network architectures. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 3, 1234–1239 vol.3. <https://doi.org/10.1109/ICNN.1995.487331>
- Haque, S., Eberhart, Z., Bansal, A., & McMillan, C. (2022). Semantic Similarity Metrics for Evaluating Source Code Summarization. *IEEE International Conference on Program Comprehension*, 2022-March, 36–47. <https://doi.org/10.1145/nnnnnnnn.nnnnnnn>
- He, P., Gao, J., & Chen, W. (n.d.). *DEBERTAV3- IMPROVING DEBERTA USING ELECTRA-STYLE PRE-TRAINING WITH GRADIENT-DISENTANGLED EMBEDDING SHARING*. <https://github.com/microsoft/DeBERTa>.
- Hidasi, B., Quadrana, M., Karatzoglou, A., & Tikk, D. (2016). Parallel recurrent neural network architectures for feature-rich session-based recommendations. *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems*, 241–248. <https://doi.org/10.1145/2959100.2959167>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Humaizi, Asmara, S., Sis, R. L., & Yusuf, M. (2020). The use of online marketplace websites in Indonesia: A study of consumers' motives and gratification. *International Journal of Interactive Mobile Technologies*, 14(7), 133–148. <https://doi.org/10.3991/ijim.v14i07.11385>
- Ijaz, M. F., & Rhee, J. (2018). Constituents and consequences of online-shopping in sustainable e-business: An experimental study of online-shopping malls. *Sustainability (Switzerland)*, 10(10). <https://doi.org/10.3390/su10103756>

- Jadama, A. F., Ca, A. J., & Jobarteh, B. (n.d.). *Ensemble Learning: Methods, Techniques, Application 4 th Modou K Toray*.
<https://doi.org/10.13140/RG.2.2.28017.08802>
- Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3), 685–695. <https://doi.org/10.1007/s12525-021-00475-2>
- Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). *Bag of Tricks for Efficient Text Classification*. <http://arxiv.org/abs/1607.01759>
- Kazmaier, J., & van Vuuren, J. H. (2022). The power of ensemble learning in sentiment analysis. *Expert Systems with Applications*, 187, 115819. <https://doi.org/https://doi.org/10.1016/j.eswa.2021.115819>
- Ketkar, N. (2017). Introduction to Keras. In *Deep Learning with Python: A Hands-on Introduction* (pp. 97–111). Apress. https://doi.org/10.1007/978-1-4842-2766-4_7
- Khadka, K., Maharjan, S., Städtjünster, C. T., & Trivsel, (. (2017). *CUSTOMER SATISFACTION AND CUSTOMER LOYALTY*.
- Khan, L., Amjad, A., Afaq, K. M., & Chang, H. T. (2022). Deep Sentiment Analysis Using CNN-LSTM Architecture of English and Roman Urdu Text Shared in Social Media. *Applied Sciences (Switzerland)*, 12(5). <https://doi.org/10.3390/app12052694>
- Kumawat, S., Yadav, I., Pahal, N., & Goel, D. (2021). Sentiment Analysis Using Language Models: A Study. *2021 11th International Conference on Cloud*

- Computing, Data Science & Engineering (Confluence)*, 984–988.
<https://doi.org/10.1109/Confluence51648.2021.9377043>
- Lamba, M., & Madhusudhan, M. (2022). Text Mining for Information Professionals: An Uncharted Territory. In *Text Mining for Information Professionals: An Uncharted Territory*, Springer International Publishing.
<https://doi.org/10.1007/978-3-030-85085-2>
- Lee, Y. C., Wang, Y. C., Lu, S. C., Hsieh, Y. F., Chien, C. H., Tsai, S. B., & Dong, W. (2016). An empirical research on customer satisfaction study: a consideration of different levels of performance. *SpringerPlus*, 5(1).
<https://doi.org/10.1186/s40064-016-3208-z>
- Lisarani, V., Tinggi, S., Katolik, A., Pontianak, N., Musa, M., Enam, I. E., & Kendari, E. (n.d.). *Rosnani Rosnani Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Profesional Makassar*.
<https://www.researchgate.net/publication/374090875>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. <http://arxiv.org/abs/1907.11692>
- Lukasz Wilk-Jakubowski Politechnika Świętokrzyska, J., & Jacek, W.-J. (n.d.). *ANALYSIS OF BROADBAND INFORMATICS SERVICES PROVIDED VIA THE INTERNET, AND THE NUMBER OF INTERNET USERS ON A GLOBAL SCALE*. <https://www.researchgate.net/publication/342353962>

- Mahadevaswamy, U. B., & Swathi, P. (2022). Sentiment Analysis using Bidirectional LSTM Network. *Procedia Computer Science*, 218, 45–56. <https://doi.org/10.1016/j.procs.2022.12.400>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. <http://arxiv.org/abs/1301.3781>
- Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 35, Issue 2, pp. 757–774). King Saud bin Abdulaziz University. <https://doi.org/10.1016/j.jksuci.2023.01.014>
- Patel, A., Oza, P., & Agrawal, S. (2022). Sentiment Analysis of Customer Feedback and Reviews for Airline Services using Language Representation Model. *Procedia Computer Science*, 218, 2459–2467. <https://doi.org/10.1016/j.procs.2023.01.221>
- Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1532–1543. <https://doi.org/10.3115/v1/d14-1162>
- Prabhakar, E., Santhosh, M., Krishnan, A. H., Kumar, T., & Sudhakar B B Student, R. (2019). Sentiment Analysis of US Airline Twitter Data using New Adaboost Approach. In *Article in International Journal of Engineering Research*. www.ijert.org
- Rane, A., & Kumar, A. (2018). Sentiment Classification System of Twitter Data for US Airline Service Analysis. *Proceedings - International Computer Software*

and Applications Conference, 1, 769–773.
<https://doi.org/10.1109/COMPSAC.2018.00114>

Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., & Gurevych, I. (2020). Classification and clustering of arguments with contextualized word embeddings. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics. Proceedings of the Conference*.
<https://doi.org/10.18653/v1/p19-1054>

Rustam, F., Ashraf, I., Mehmood, A., Ullah, S., & Choi, G. S. (2019). Tweets classification on the base of sentiments for US airline companies. *Entropy*, 21(11). <https://doi.org/10.3390/e21111078>

Saad, A. I. (2020). Opinion Mining on US Airline Twitter Data Using Machine Learning Techniques. *16th International Computer Engineering Conference, ICENCO 2020*, 59–63. <https://doi.org/10.1109/ICENCO49778.2020.9357390>

Schmidt, R. M. (n.d.). *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*.

Sherstinsky, A. (n.d.). *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*.
<https://www.linkedin.com/in/alexsherstinsky>

Sherstinsky, A. (2018). *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*.
<https://doi.org/10.1016/j.physd.2019.132306>

- Singgalen, Y. A. (2024). Sentiment Analysis and Trend Mapping of Hotel Reviews Using LSTM and GRU. *Journal of Information Systems and Informatics*, 6(4), 2814–2836. <https://doi.org/10.51519/journalisi.v6i4.926>
- Singh, S., Zolkepli, I. A., & Kit, C. W. (2018). New wave in mobile commerce adoption via mobile applications in Malaysian market: Investigating the relationship between consumer acceptance, trust, and self efficacy. *International Journal of Interactive Mobile Technologies*, 12(7), 112–128. <https://doi.org/10.3991/ijim.v12i7.8964>
- Teodorescu, V., & Obreja Brasoveanu, L. (2025). Assessing the Validity of k-Fold Cross-Validation for Model Selection: Evidence from Bankruptcy Prediction Using Random Forest and XGBoost. *Computation*, 13(5). <https://doi.org/10.3390/computation13050127>
- Vadivukarassi, M., Puviarasan, N., & Aruna, P. (2018). An Exploration of Airline Sentimental Tweets with Different Classification Model. *International Journal for Research in Engineering Application & Management (IJREAM)*, 04, 2. <https://doi.org/10.18231/2454-9150.2018.0124>
- Wadawadagi, R., & Pagi, V. (2020). Sentiment analysis with deep neural networks: comparative study and performance assessment. *Artificial Intelligence Review*, 53(8), 6155–6195. <https://doi.org/10.1007/s10462-020-09845-2>
- Zhang, S., Wei, Z., Wang, Y., & Liao, T. (2018). Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary. *Future Generation Computer Systems*, 81, 395–403. <https://doi.org/10.1016/j.future.2017.09.048>

Zheng, X., Lee, M., & Cheung, C. M. K. (2017). Examining e-loyalty towards online shopping platforms: The role of coupon proneness and value consciousness. *Internet Research*, 27(3), 709–726. <https://doi.org/10.1108/IntR-01-2016-0002>

