

TESIS

**EVALUASI KINERJA PENGKLASIFIKASIAN PEMBELAJARAN MESIN
MENGUNAKAN METODE EKSTRAKSI FITUR PADA DATA
TWITTER**



Disusun oleh:

Nama : Dedy Suglarto
NIM : 20.55.1383
Konsentrasi : Business Intelligence

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

2023

TESIS

**EVALUASI KINERJA PENGKLASIFIKASIAN PEMBELAJARAN MESIN
MENGUNAKAN METODE EKSTRAKSI FITUR PADA DATA
TWITTER**

**PERFORMANCE EVALUATION OF MACHINE LEARNING
CLASIFIER USING FEATURE EXTRACTION METHODS IN TWITTER
DATA**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Dedy Suglarto
NIM : 20.55.1383
Konsentrasi : Business Intelligence

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

2023

HALAMAN PENGESAHAN

**EVALUASI KINERJA PENGKLASIFIKASIAN PEMBELAJARAN MESIN
MENGUNAKAN METODE EKSTRAKSI FITUR PADA DATA TWITTER**

**PERFORMANCE EVALUATION OF MACHINE LEARNING CLASSIFIER USING
FEATURE EXTRACTION METHODS IN TWITTER DATA**

Dipersiapkan dan Disusun oleh

Dedy Sugiarto

20.55.1383

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 PJJ Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 6 September 2022

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

Yogyakarta, 6 September 2022

Rektor

Prof. Dr. M. Suvanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

EVALUASI KINERJA PENGKLASIFIKASIAN PEMBELAJARAN MESIN MENGUNAKAN METODE EKSTRAKSI FITUR PADA DATA TWITTER

PERFORMANCE EVALUATION OF MACHINE LEARNING CLASSIFIER USING FEATURE EXTRACTION METHODS IN TWITTER DATA

Dipersiapkan dan Disusun oleh

Dedy Suglarto

20.55.1383

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 PJJ Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 6 September 2022

Pembimbing Utama

Anggota Tim Penguji

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

Dr. Andi Sunyoto, M.Kom
NIK. 190302052

Pembimbing Pendamping

Dhani Ariatmanto, M.Kom, Ph.D
NIK. 190302197

Ainul Yaqin, M.Kom
NIK. 190302255

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

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

Yogyakarta, 6 September 2022
Direktur Program Pascasarjana

Dr. Kusriani, M.Kom.
NIK. 190302106

BALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini :

Nama : Dedy Sugiarto
NIM : 20.55.1383
Konsentrasi : Business Intelligence

Menyatakan bahwa tesis dengan judul berikut :
Evaluasi Kinerja Pengklasifikasi Pembelajaran Mesin Menggunakan Metode Ekstraksi
Fitur pada Data Twitter

Dosen Pembimbing Utama : Prof. Dr. Ema Uliani, SSI, M.Kom
Dosen Pembimbing Pendamping : Ainal Yaqin, M.Kom

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 tidak terdapat karya atau peninggalan 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 bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

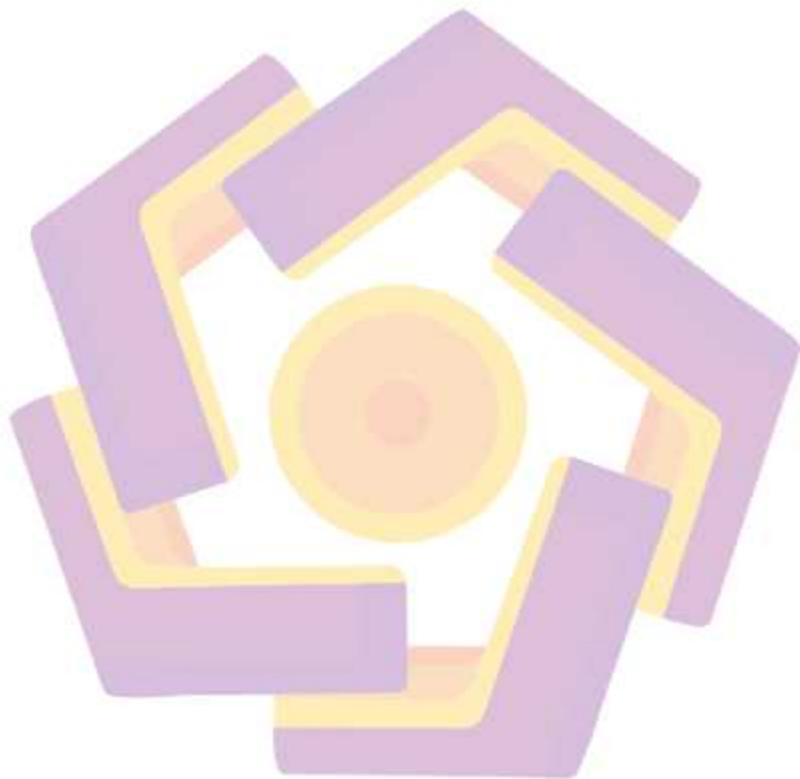
Bekasi, 6 September 2022



Dedy Sugiarto

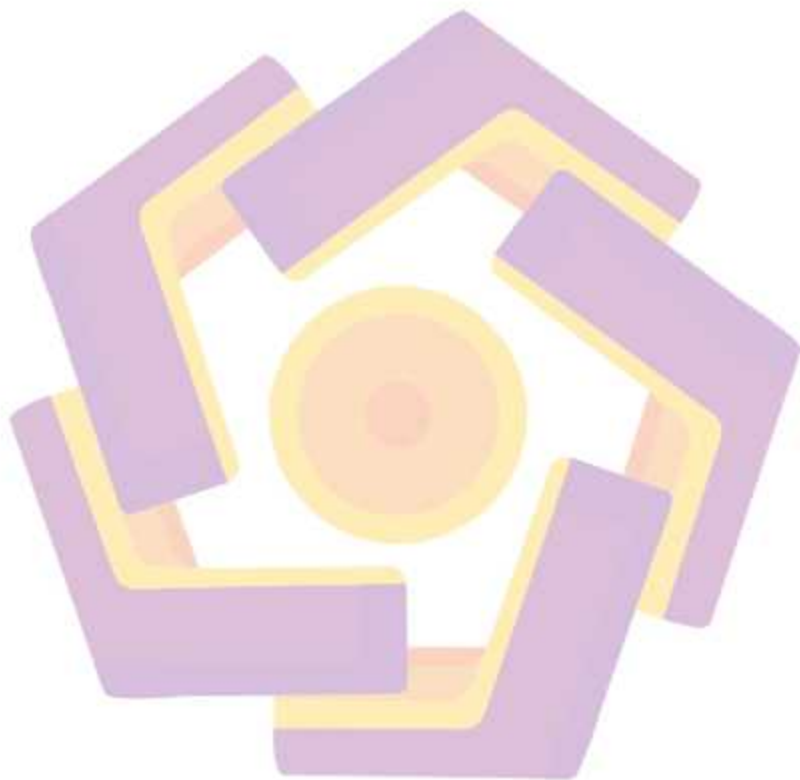
HALAMAN PERSEMBAHAN

Kupersembahkan untuk almarhum bapak, almarhumah ibu, isteri serta ketiga putraku.



HALAMAN MOTTO

Janganlah berputus asa dari rahmat Allah, tetap semangat dan isi sisa usia dengan ibadah yang sungguh-sungguh.



KATA PENGANTAR

Syukur alhamdulillah akhirnya tesis ini dapat diselesaikan. Ucapan terima kasih disampaikan kepada :

1. Komisi pembimbing yaitu ibu Prof. Dr. Ema Utami, SSi, M.Kom dan bapak Ainul Yaqin, M.Kom atas bimbingan, pengertian dan kesabarannya dalam penyelesaian tesis ini
2. Prof. Dr. M. Suyanto, MM sebagai Rektor Universitas AMIKOM Yogyakarta
3. Prof. Dr. Kusriani, M.Kom sebagai Direktur Program Magister Teknik Informatika Program Pascasarjana Universitas AMIKOM Yogyakarta
4. Kolega para pimpinan di Fakultas Teknologi Industri serta dosen, calon dosen dan laboran di Jurusan Teknik Informatika dan Program Studi Magister Teknik Industri FTI Universitas Trisakti Jakarta

Tesis ini masih jauh dari kesempurnaan, oleh karena itu saran dari pembaca akan sangat membantu kelanjutan penelitian ini untuk pengembangan ilmu pengetahuan dan kemaslahatan masyarakat.

Bekasi, 6 September 2022

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR ISTILAH.....	xvi
INTISARI.....	xvii
<i>ABSTRACT</i>	xviii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penelitian.....	7
1.6. Hipotesis.....	8
BAB II TINJAUAN PUSTAKA.....	9

2.1. Tinjauan Pustaka.....	9
2.2. Keaslian Penelitian.....	11
2.3. Landasan Teori.....	14
2.3.1 Analisis Sentimen	14
2.3.2 Text Preprocessing.....	15
2.3.3 Model Bag of Word (BoW)	16
2.3.4 Word Embedding	18
2.3.5 Algoritma Pengklasifikasian	19
2.3.6 Algoritma Pembelajaran Mesin.....	21
2.3.7 Algoritma Pembelajaran Mendalam	23
BAB III METODE PENELITIAN.....	27
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	27
3.2. Metode Pengumpulan Data.....	27
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	32
4.1. Pra-pemrosesan Dataset Penghapusan Harga Eceran Tertinggi (HET)..	32
4.2. Pra-pemrosesan Dataset Pemberian Bantuan Langsung Tunai (BLT) ...	34
4.3. Ekstraksi Fitur Word2vec pada Dataset HET.....	39
Berdasarkan Model CBOW/SG dan Dimensi Vektor	48
BAB V PENUTUP.....	71
5.1. Kesimpulan	71
5.2. Saran	73
DAFTAR PUSTAKA	74
LAMPIRAN.....	80

DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian Evaluasi Kinerja Pengklasifikasian Pembelajaran Mesin Menggunakan Metode Ekstraksi Fitur pada Data Twitter	11
Tabel 4.1. Contoh kamus data pada dataset HET	39
Tabel 4.2. Contoh kamus dan nilai index	40
Tabel 4.3. Contoh vector kata dari kata 'mafia' dan 'kartel'	41
Tabel 4.4. Contoh Kamus Kata pada Dataset BLT	56
Tabel 4.5. Contoh Tweet dan Nilai Indeks pada Dataset BLT	57
Tabel 4.6. Contoh Vector kata dari kata 'presiden' dan 'joko'	58

DAFTAR GAMBAR

Gambar 2.1. Peta Kajian Analisis Sentimen (Lighthart et al., 2021).....	15
Gambar 2.2. Model CBOW dan Skip-Gram dalam Word2vec	19
Gambar 2.3. Posisi bidang keilmuan <i>deep learning</i> (Goodfellow et al., 2012)....	20
Gambar 2.4. Teknik dari SVM untuk Hyperplane emisah yang memiliki margin maksimum.....	22
Gambar 2.5. Skema Model Regresi Logistic (Skansi, 2020).....	23
Gambar 2.6. Skema Model Perceptron	24
Gambar 2.7. Skema Model MLP	24
Gambar 2.8. Model CNN untuk klasifikasi teks (Brownlee, 2017).....	25
Gambar 2.9. Model Bi-LSTM (Jang et al., 2020).....	26
Gambar 3.1. Diagram Alur Penelitian.....	29
Gambar 3.2 Rancangan 1	30
Gambar 3.3 Rancangan 2	31
Gambar 4.1. Contoh tweet hasil crawling dataset HET.....	32
Gambar 4.2. Contoh tweet yang telah melewati tahap pra-proses pada dataset HET	33
Gambar 4.3. Contoh hasil pelabelan sentiment dataset HET.....	33
Gambar 4.4. Diagram lingkaran kategori sentiment pada dataset HET.....	34
Gambar 4.5. Contoh tweet hasil crawling dataset BLT	35
Gambar 4.6. Contoh tweet yang telah melewati tahap pra-proses dataset BLT ...	35
Gambar 4.7. Contoh hasil pelabelan sentiment dataset BLT	36

Gambar 4.8. Word cloud dari sentiment positif.....	36
Gambar 4.9. Word cloud dari sentiment negative.....	37
Gambar 4.10. Diagram lingkaran kategori sentiment pada dataset HET 2.....	38
Gambar 4.11. Contoh nilai ekstraksi fitur BOW.....	38
Gambar 4.12. Contoh nilai ekstraksi fitur TF-IDF	39
Gambar 4.13. Hasil perhitungan cosine similarity kata 'mafia' dan 'kartel'	42
Gambar 4.14. Cosine Similarity berdasarkan arsitektur word2vec dan dimensi dari kata 'Mafia' dan 'kartel'	43
Gambar 4.15. Nilai Cosine Similarity dari beberapa pasangan kata.....	44
Gambar 4.16. Struktur Lapisan BLSTM pada Data HET	45
Gambar 4.17. Kinerja Klasifikasi Dataset HET menggunakan BLSTM dengan Window Size 5.....	46
Gambar 4.18. Perbandingan Rata-rata Kinerja Klasifikasi BLSTM dengan Word2vec CBOW dan SG.....	47
Gambar 4.19. Perbandingan Rata-rata Kinerja Klasifikasi BLSTM dengan Word2vec Berdasarkan Dimensi Vektor	47
Gambar 4.20. Perbedaan Rata-rata Kinerja Klasifikasi BLSTM.....	48
Berdasarkan Model CBOW/SG dan Dimensi Vektor.....	48
Gambar 4.21. Model Akurasi Berdasarkan BLSTM SG 50D	49
Gambar 4.22. Model Loss Berdasarkan BLSTM SG 50D.....	49
Gambar 4.23. Model Akurasi Berdasarkan BLSTM CBOW 200D.....	50
Gambar 4.24. Model Loss Berdasarkan BLSTM CBOW 200D.....	50

Gambar 4.25. Kinerja klasifikasi Dataset HET menggunakan SVM dengan Word2Vec Window Size 5, TF-IDF dan BOW.....	51
Gambar 4.26. Hasil Penyetelan Parameter C dan Kernel pada Klasifikasi SVM-TF-IDF.....	53
Gambar 4.27. Hasil Penyetelan Parameter C dan Kernel pada Nilai Akurasi Klasifikasi SVM-BOW.....	54
Gambar 4.28. Confusion Matrix dari BLSTM dengan model SG 50D	55
Gambar 4.29. Confusion Matrix dari SVM dengan model TF-IDF	55
Gambar 4.30. Confusion Matrix dari SVM dengan model BOW.....	56
Gambar 4.31. Hasil perhitungan cosine similarity kata 'presiden' dan 'joko'	58
Gambar 4.32. Nilai Cosine Similarity dari beberapa pasangan kata pada dataset BLT.....	59
Gambar 4.33. Struktur Lapisan ANN pada Data BLT	60
Gambar 4.34. Model Accuracy dari ANN pada Data BLT.....	61
Gambar 4.35. Model loss dari ANN pada Data BLT.....	61
Gambar 4.36. Struktur Lapisan CNN pada Data BLT	63
Gambar 4.37. Model Accuracy dari CNN pada Data BLT.....	63
Gambar 4.38. Model Loss dari CNN pada Data BLT.....	64
Gambar 4.39. Struktur Lapisan LSTM pada Data BLT	64
Gambar 4.40. Model Accuracy dari LSTM pada Data BLT.....	65
Gambar 4.41. Model loss dari LSTM pada Data BLT.....	65
Gambar 4.42. Struktur Lapisan BLSTM pada Data BLT	66
Gambar 4.43. Model Accuracy dari BLSTM pada Data BLT	66

Gambar 4.44. Model loss dari BSTM pada Data BLT	67
Gambar 4.45. Perbandingan kinerja Klasifikasi Menggunakan Ekstraksi Fitur Word2vec.....	68
Gambar 4.46. Perbandingan Kinerja Klasifikasi Menggunakan Ekstraksi TF-IDF dan BOW	68
Gambar 4.47. Confusion Matrix Klasifikasi LSTM-Word2vec pada Data BLT..	69
Gambar 4.48. Confusion Matrix Klasifikasi LR-BOW pada Data BLT.....	70



DAFTAR ISTILAH

- BLSTM *Bidirectional long short term memory* – salah satu teknik jaringan syaraf tiruan yang termasuk dalam kategori pembelajaran mendalam (*deep learning*) dimana input mengalir di kedua arah
- BLT Bantuan Langsung Tunai – program bantuan pemerintah untuk golongan miskin atau kurang mampu sebagai dampak tingginya harga kebutuhan pokok dan dapat diberikan dalam bentuk uang tunai atau bantuan lainnya
- CBOW *Continuous bag of word* – salah satu arsitektur dari ekstraksi fitur *Word2Vec* yang mencoba untuk memahami konteks kata-kata dan mencoba memprediksi kata berdasarkan tetangganya,
- HET Harga Eceran Tertinggi – merupakan kebijakan harga oleh pemerintah terkait harga jual tertinggi yang dikenakan pada suatu produk
- LR-BOW *Logistic regression – bag of word* – teknik klasifikasi regresi logistik yang digabung dengan penggunaan ekstraksi fitur *bag of word*
- LR-TF-IDF *Logistic regression – term frequency inverse-document frequency* - teknik klasifikasi regresi logistik yang digabung dengan penggunaan ekstraksi fitur *term frequency-inverse document frequency*
- SG *Skip-gram* - mencoba untuk memahami konteks kata-kata serta mencoba memprediksi tetangga dari sebuah kata
- SG 50D *Skip-gram* dengan ukuran dimensi vector kata sebesar 50

INTISARI

Walaupun analisis sentimen atau penambangan opini saat ini merupakan topik yang paling aktif diteliti di bidang pemrosesan bahasa alami, namun belum banyak digunakan di sektor pertanian. Oleh karena itu dilakukan penelitian terkait evaluasi kinerja hasil klasifikasi sentimen menggunakan data twitter terkait bidang pertanian serta melihat kemampuan ekstraksi fitur *Word2Vec* dalam menangkap hubungan semantik dilihat berdasarkan arsitektur model dan dimensi vector yang digunakan. Kasus yang diangkat adalah mengenai gejolak harga minyak goreng yang terjadi di Indonesia pada tahun 2022. Dataset twitter pertama diambil menggunakan batasan waktu 21-25 Maret 2022 dengan kata kunci minyak goreng. Pada periode itu adalah berlangsungnya kebijakan pemerintah menghapus Harga Eceran Tertinggi (HET) dimulai 16 Maret 2022. Dataset kedua terkait kebijakan Bantuan Langsung Tunai (BLT) minyak goreng sebagai dampak kenaikan harga minyak goreng yang diberikan mulai April 2022. Hasil penelitian menunjukkan bahwa rata-rata nilai akurasi dan F1-score pada model *skip-gram* (SG) cenderung lebih tinggi dibandingkan dengan model *continuous bag of word* (CBOW) pada data HET yang memiliki ukuran data 12408. Nilai rata-rata akurasi pada model CBOW sebesar 0,675 dan SG 0,700 sedangkan rata-rata F1-score pada CBOW sebesar 0,6375 dan SG 0,6550. Berdasarkan pemeriksaan model akurasi dan loss dapat diketahui bahwa model *bidirectional long-short term memory* (BLSTM) dengan SG 50D menunjukkan pola data latih dan data tes yang lebih mirip serta nilai akurasi 0,7 dan F1-Score 0,67. Hasil pemeriksaan pada dimensi yang lebih tinggi menunjukkan adanya gejala *overfitting*. Nilai F1-score dan akurasi tertinggi dengan menggunakan data BLT didapatkan pada model *logistic regression – bag of word* (LR-BOW) dengan nilai berturut-turut sebesar 0,68 dan 0,75 disusul oleh *logistic regression – term frequency inverse document frequency* (LR – TF-IDF) dan *support vector machine – bag of word* (SVM-BOW). Model *Word2Vec* telah berhasil menangkap makna kata yang terkandung dalam data HET seperti kata 'mafia' memiliki nilai *cosine similarity* yang cukup tinggi dengan kata 'kartel', 'bongkar', 'tangkap', 'oligarki' dan 'mendag'. Hal ini perlu menjadi perhatian pemerintah untuk menangani harga minyak goreng. Nilai kemiripan dari pasangan kata akan semakin menurun dengan semakin besarnya nilai dimensi vector.

Kata kunci: analisis sentimen, minyak goreng, twitter, ekstraksi fitur, klasifikasi, hubungan semantik

ABSTRACT

While sentiment analysis or opinion mining is currently the most actively researched topic in the field of natural language processing, it has not been widely used in the agricultural sector. Therefore, research was carried out regarding the performance evaluation of sentiment classification results using Twitter datasets related to agriculture and looking at the ability of Word2Vec feature extraction in capturing the closeness of meaning seen based on the model architecture and vector dimensions used. The case raised is regarding the cooking oil price volatility that occurred in Indonesia in 2022. The first Twitter dataset was retrieved using the time limit of 21-25 March 2022 with the keyword cooking oil. In that period, the government's policy to remove the Highest Retail Price (HET) began on March 16, 2022. The second dataset related to the Direct Cash Assistance (BLT) policy for cooking oil as a result of rising cooking oil prices was given starting April 2022. The results showed that on average the accuracy value and F1-score in the skip-gram (SG) model tend to be higher than the continuous bag of word (CBOW) model on HET data which has a data size of 12408. The average value of accuracy in the CBOW model is 0.675 and SG 0.700 while the average F1-score at CBOW is 0.6375 and SG is 0.6550. Based on the examination of the accuracy and loss models, it can be seen that the bidirectional long-short term memory (BLSTM) model with the SG 50D shows a more similar pattern of training data and test data and an accuracy value of 0.7 and an F1-Score of 0.67. Examination results at a higher dimension show symptoms of overfitting. The highest f1-score and accuracy using BLT data were obtained in the logistic regression – bag of word (LR-BOW) model with values of 0.68 and 0.75 respectively, followed by logistic regression – term frequency inverse document frequency (LR – TF-IDF) and support vector machine – bag of word (SVM-BOW). The Word2Vec model has succeeded in capturing the meaning of the words contained in the data. HET, like the word 'mafia', has a fairly high cosine similarity value to the words 'cartel', 'demolish', 'arrest', 'oligarchy' and 'minister of trade'. This needs to be a concern of the government to handle the price of cooking oil. The similarity value of word pairs will decrease with the increasing value of the vector dimension.

Keyword: sentiment analysis, cooking oil, feature extraction, classification, semantic relationship

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Saat ini masyarakat banyak menggunakan media sosial twitter untuk berbagi pendapat atau opini terhadap suatu isu terkait suatu produk, kebijakan pemerintah maupun berbagai program (Andoh et al., 2021). Pendapat masyarakat atau opini publik baik bersifat positif, negatif maupun netral terkait sebuah kebijakan tertentu atau fenomena di masyarakat ini menjadi hal yang berharga untuk dilacak dan dianalisis melalui sebuah metode yang disebut sebagai analisis sentimen. Penentuan polaritas sentimen tersebut adalah salah satu sub tugas dari klasifikasi sentimen (Wankhade et al., 2022) (Ligthart et al., 2021). Hasil analisis dapat menjadi informasi dan pengetahuan bagi para pengambil keputusan terkait baik dari pihak pemerintah, swasta maupun komunitas masyarakat.

Walaupun analisis sentimen atau penambangan opini saat ini merupakan topik yang paling aktif diteliti di bidang pemrosesan bahasa alami, namun tidak banyak digunakan di sektor pertanian (Novák et al., 2021). Beberapa penelitian dengan dataset twitter terkait pertanian dan perdagangan komoditas pertanian antara lain pernah dilakukan oleh (Surjandari et al., 2014) dimana disimpulkan bahwa perubahan harga susu, telur dan bawang merah adalah tiga komoditas yang memiliki kontribusi tertinggi dalam munculnya sentimen negatif tweet. Hasil penelitian juga menunjukkan bahwa klasifikasi dengan *Support Vector Machine* (SVM) memberikan tingkat akurasi yang paling tinggi dibandingkan metode Naïve

Bayes dan Decision Tree yaitu sebesar 75,19% (tanpa stemming) dan 80,35 % (dengan stemming). Namun demikian penelitian tersebut belum membandingkan pengaruh dari perbedaan ekstraksi fitur yang digunakan serta tidak terkait dengan adanya kebijakan tertentu yang dikeluarkan pemerintah.

Penelitian terkait opini publik tentang kebijakan pemerintah antara lain tentang impor beras pernah dilakukan oleh (Nandini et al., 2019) yang menggunakan metode klasifikasi SVM namun hasil akurasi yang didapatkan masih relative rendah yaitu berada pada kisaran di bawah 61%. Analisis sentiment juga pernah dilakukan terkait kebijakan dibukanya keran ekspor benih lobster namun nilai akurasi yang didapatkan masih berada pada angka 72% (Sulastri, 2020). Berdasarkan kondisi ini dinilai perlu dilakukan penelitian yang dapat meningkatkan kinerja klasifikasi dengan kasus opini masyarakat terhadap kebijakan di bidang pertanian yang dikeluarkan pemerintah.

Terdapat beberapa penelitian yang bertujuan peningkatan kinerja klasifikasi antara lain menggunakan ekstraksi fitur word embedding serta klasifikasi melalui *deep learning (Bidirectional Long Short-Term Memory)* untuk sentiment berbahasa Indonesia dengan kasus non pertanian dimana berhasil meningkatkan kinerja model yang sebelumnya menggunakan ekstraksi fitur *term frequency-inverse document frequency (TF-IDF)* dan model SVM dimana *F1-score* meningkat dari 86,58% ke 93,69% (Purwarianti & Crisdayanti, 2019). Penelitian yang dilakukan (Agustiningsih et al., 2022) membandingkan 8 skenario dari metode ekstraksi fitur *fastText* dan *Global Vector for Word Reprsentatioan (GloVe)* yang divariasikan berdasarkan kondisi dengan atau tanpa stemming serta berdasarkan kondisi pre-

trained dan self-trained. Dataset yang digunakan adalah data twitter berbahasa Indonesia terkait vaksin covid-19. Hasilnya menunjukkan bahwa Model yang menggunakan GloVe menghasilkan akurasi yang lebih tinggi pada dataset tanpa stemming, sedangkan model menggunakan kata *fastText* memiliki akurasi yang lebih tinggi ketika menggunakan data yang telah mengalami stemming. Penelitian oleh (Pasaribu et al., 2020) yang menggunakan ekstraksi fitur Word2Vec dan GloVe dengan model *deep learning* CNN dan LSTM berhasil meningkatkan akurasi menjadi 93 %.

Beberapa peneliti juga telah melakukan penelitian yang bertujuan untuk membandingkan kinerja metode ekstraksi fitur pada beberapa jenis data set serta dengan menggunakan beberapa algoritma klasifikasi. Penelitian yang dihasilkan oleh (Ahuja et al., 2019) menunjukkan bahwa ekstraksi fitur menggunakan TF-IDF memberikan hasil yang lebih baik (3-4%) dibandingkan dengan fitur N-Gram (N=2) yang diterapkan pada 6 algoritma klasifikasi yang berbeda. Dataset yang digunakan adalah SS -TWEET yang merupakan singkatan dari *Sentimen Strength Twitter Dataset*. Dataset ini dianotasi secara manual. Menurut (Sabri et al., 2021) yang membandingkan TF-IDF, word count and word2vec pada dataset dengan tulisan Arab menunjukkan adanya pengaruh signifikan terhadap kinerja model klasifikasi. Pengklasifikasi SVM dan LR memiliki kinerja terbaik diikuti oleh RF, KNN dan DT. Penelitian yang dilakukan (Shahi et al., 2022) yang menggunakan data twitter terkait COVID-19 dalam bahasa Nepal menunjukkan bahwa metode ekstraksi fitur gabungan mengungguli dua metode ekstraksi fitur individu (TF-IDF dan *fastText*) saat menggunakan sembilan algoritma pembelajaran mesin yang

berbeda. Namun demikian beberapa penelitian menunjukkan juga bahwa ekstraksi fitur konvensional masih lebih baik dibandingkan ekstraksi fitur word2vec ((Cahyani & Patasik, 2021)(Oyeon Park et al., 2020).

Selain pengaruh ekstraksi fitur terhadap kinerja klasifikasi, beberapa penelitian juga telah dilakukan untuk menguji pengaruh parameter word2vec seperti arsitektur *Continuous Bag of Word* (CBOW) dan *Skip-gram* (SG), dimensi vector serta ukuran jendela (*window size*) terhadap hubungan semantik dalam berbagai kata bahasa Inggris (Jatnika et al., 2019) dan (Zhang, 2022). Hasil penelitian oleh menunjukkan bahwa semakin tinggi ukuran jendela dan dimensi vektor, semakin banyak konteks kata dan semakin lemah kesamaannya. Namun, kedua penelitian tersebut masih membahas kesamaan dalam konteks bahasa Inggris. Berdasarkan hal ini dipandang perlu untuk melakukan penelitian terkait evaluasi kinerja hasil klasifikasi menggunakan beberapa metode ekstraksi fitur dengan menggunakan dataset twitter terkait bidang pertanian serta kemampuan Word2Vec dalam menangkap kedekatan makna dilihat berdasarkan arsitektur model dan dimensi vector yang digunakan.. Dalam hal ini kasus diangkat adalah mengenai gejolak harga minyak goreng yang terjadi di Indonesia pada tahun 2022.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dibahas sebelumnya maka rumusan masalah dalam penelitian ini sebagai berikut :

Apakah pengaruh model Word2Vec (CBOW dan SG) dan dimensi vector terhadap kinerja klasifikasi menggunakan metode BLSTM dan SVM?

Apakah penggunaan metode ekstraksi fitur yang berbeda (Word2Vec, BOW dan TF-IDF) akan berpengaruh signifikan terhadap akurasi hasil klasifikasi sentiment berdasarkan data twitter terkait opini masyarakat mengenai gejala harga minyak goreng

Apakah penggunaan algoritma klasifikasi pembelajaran mesin yang berbeda baik dengan menggunakan metode non deep learning maupun deep learning akan berpengaruh signifikan terhadap akurasi hasil klasifikasi sentiment berdasarkan data twitter terkait opini masyarakat mengenai gejala harga minyak goreng

Apakah kata yang memiliki kedekatan makna dengan kata yang paling sering dibicarakan dalam kasus gejala harga minyak goreng bila digunakan ekstraksi fitur Word2Vec.

1.3. Batasan Masalah

Beberapa batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut:

- a. Data yang dipakai pada penelitian ini adalah data opini public berbasis twitter yang terbagi menjadi tiga bagian yaitu opini public saat adanya kebijakan Harga Eceran Tertinggi (HET) minyak goreng, kebijakan pencabutan HET dan kebijakan pemberian Bantuan Langsung Tunai (BLT) sebagai dampak kenaikan harga minyak goreng

- b. Metode ekstraksi fitur yang dibandingkan adalah BoW, TF-IDF dan word2vec
- c. Metode pengklasifikasi yang dibandingkan adalah metode *shallow learning* (*naïve bayes, k-nearest neighbor, support vector machine, logistic regression*) serta *deep learning* (*artificial neural network, convolutional neural network, long short term memory, bidirectional long short term memory*).
- d. Penelitian ini hanya berfokus pada teks cuitan berbahasa Indonesia.
- e. Penelitian ini tidak memperhatikan kesalahan ejaan dan sinonim.
- f. Penelitian ini tidak memperhatikan penanganan negasi, sarkasme, hiperbola, dan pendeteksian emosi.
- g. Tahap *stemming* menggunakan algoritma *stemming* Nazief-Adriani.

1.4. Tujuan Penelitian

Bagian ini memuat penjelasan secara spesifik:

Adapun tujuan penulis dalam melakukan penelitian ini adalah sebagai berikut :

1. Mengetahui pengaruh model Word2Vec (CBOW dan SG) dan dimensi vector terhadap kinerja klasifikasi menggunakan metode BLSTM dan SVM
2. Mengetahui kinerja hasil klasifikasi sentiment yang diperlakukan dengan menggunakan metode ekstraksi fitur yang berbeda (Word2Vec, TF-IDF dan BOW) berdasarkan data twitter terkait opini masyarakat mengenai gejolak harga minyak goreng

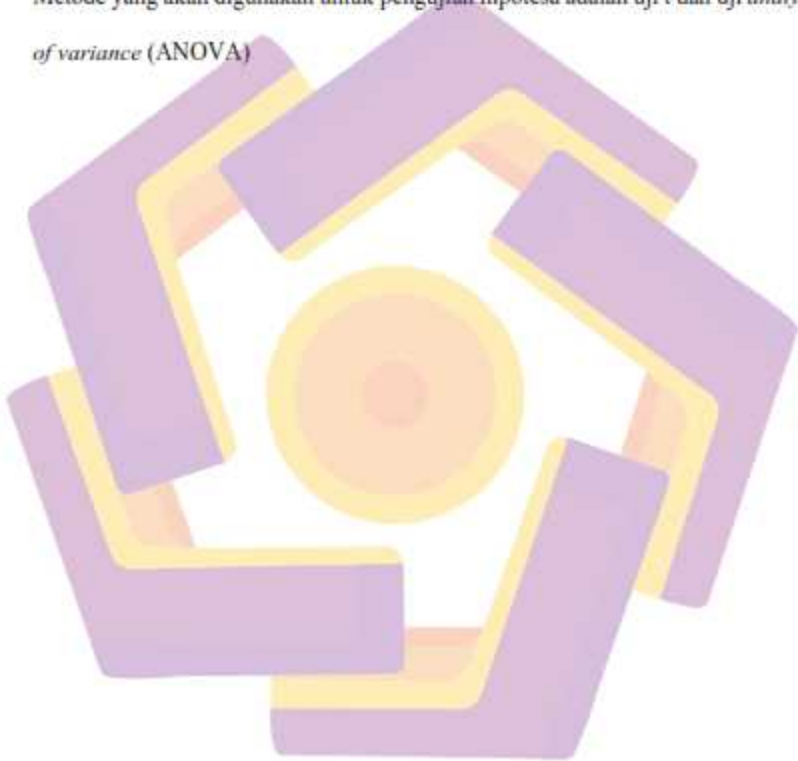
3. Mengetahui hasil kinerja klasifikasi yang diperlakukan dengan menggunakan algoritma klasifikasi pembelajaran mesin yang berbeda baik dengan menggunakan metode non deep learning maupun deep learning berdasarkan data twitter terkait opini masyarakat mengenai gejolak harga minyak goreng
4. Mengidentifikasi kata yang memiliki kedekatan makna dengan kata yang paling sering dibicarakan dalam kasus gejolak harga minyak goreng bila digunakan ekstraksi fitur Word2Vec.

1.5. Manfaat Penelitian

Manfaat dari penelitian diharapkan dapat memberikan informasi mengenai kombinasi metode ekstraksi fitur dan algoritma klasifikasi berbasis pembelajaran mesin yang terbaik untuk kasus opini public terkait gejolak harga komoditas pangan. Penelitian ini juga diharapkan dapat menjadi masukan bagi pemerintah dalam menentukan kebijakan yang sesuai dengan harapan masyarakat.

1.6. Hipotesis

Beberapa hipotesis ditetapkan dalam penelitian yang terkait dengan pengujian perbedaan rata-rata akurasi kinerja klasifikasi antara model CBOW dan SG serta perbedaan rata-rata akurasi klasifikasi antar ukuran dimensi vector. Metode yang akan digunakan untuk pengujian hipotesa adalah uji t dan uji *analysis of variance* (ANOVA)



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Data twitter yang bersifat teks memerlukan proses konversi ke dalam data vektor yang bersifat numerik dimana beberapa peneliti telah melakukan penelitian tentang bagaimana proses konversi tersebut serta dampaknya terhadap kinerja hasil klasifikasi. Dataset yang digunakan juga bervariasi dari yang menggunakan dataset bersifat publik maupun dataset yang ditarik sendiri terkait kondisi saat ini seperti terjadinya pandemi covid 19. Penelitian yang dihasilkan oleh (Ahuja et al., 2019) membandingkan beberapa metode ekstraksi fitur yaitu TF-IDF dan teknik mengelompokkan kata yaitu N-Gram dimana jumlah kata yang digunakan sebanyak 2 untuk menghindari membengkaknya jumlah kata unik yang digunakan. Hasil penelitiannya menunjukkan bahwa ekstraksi fitur menggunakan TF-IDF memberikan hasil yang lebih baik dibandingkan dengan fitur N-Gram (N=2) yang diterapkan pada 6 algoritma klasifikasi yang berbeda. Terdapat 6 algoritma yang digunakan yaitu regresi logistik, naive bayes, support vector machine, Decision tree, K-nearest neighbour (KNN) serta random forest.

Peneliti lain (Sabri et al., 2021) membandingkan TF-IDF, word count atau dikenal juga dengan istilah BoW serta metode word2vec yang diterapkan pada dataset berbahasa Arab. Terdapat lima pengklasifikasi yang digunakan yaitu SVM, DT, RF, KNN dan LR. Hasilnya menunjukkan adanya pengaruh signifikan terhadap kinerja model klasifikasi. Metode regresi logistic (LR) menunjukkan hasil terbaik

pada ketiga metode ekstraksi yang digunakan. Penelitian lain dilakukan menguji pengaruh ekstraksi fitur dari word2vec dan algoritma SVM, NB, CNN dan LSTM dengan mevariasikan teks dinormalisasi atau tidak. Hasilnya menunjukkan bahwa model CNN analisis sentimen Melayu telah berhasil dibangun dan divalidasi untuk mencapai akurasi hingga 77,59%.

Penelitian lain juga dilakukan untuk membandingkan pengaruh dari TF-IDF, BoW dan penggabungan BoW dan TF-IDF dengan menggunakan beberapa algoritma klasifikasi seperti RF, XGBoost, SVC, ETC dan DT terhadap kinerja klasifikasi seperti yang dilakukan oleh (Rustam et al., 2021). Dari ketiga teknik ekstraksi ciri yang digunakan, TF-IDF, BoW, dan gabungannya, pendekatan gabungan memberikan hasil yang lebih baik. Rekomendasinya disarankan untuk mengarahkan riset ke pendekatan pembelajaran mendalam, khususnya untuk meningkatkan kinerjanya pada kumpulan data kecil.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	The Impact of Features Extraction on the Sentiment Analysis	Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P Procedia Computer Science 152 (2019) 341–348	Mengetahui pengaruh TF-IDF N-Gram (N=2) pada 6 algoritma klasifikasi yang berbeda. Dataset yang digunakan adalah SS -TWEET	TF-IDF memberikan hasil yang lebih baik (3-4%) dibandingkan dengan fitur N-Gram (N=2).	Perlu perbandingan fitur lain seperti fitur skor polaritas kata, penyisipan kata, spesifik twitter Fitur. Nilai kinerja (precision, recall, F-score dan accuracy) masih relative rendah dimana kisaran nilai berada diantara 0.2 sd 0.6	Persamaannya menggunakan ekstraksi TF-IDF juga sebagai salah satu ekstraksi fitur yang dibandingkan. Perbedaannya pada dataset terkait opini public mengenai komoditas pertanian
2	Comparative study of Arabic text classification using feature vectorization methods	Sabri, T., Beggar, O. El. & Kissi, M. (2021). Procedia Computer Science 198 (2022) 269–275	Membandingkan TF-IDF, word count and word2vec pada dataset dengan tulisan Arab	Menunjukkan adanya pengaruh signifikan perbedaan ekstraksi fitur terhadap kinerja model klasifikasi. Nilai F1 dari LR -BOW sebesar 98.91% mengungguli LR TF-IDF 98.79% dan LR Word2vec 98.10%, Pengklasifikasi SVM dan DT	Idenya bisa diperluas untuk menggabungkan metode vektorisasi fitur dan menggunakan set data Arab lainnya untuk menguji kinerja masing-masing klasifikasi	Persamaannya metode ekstraksi fitur yang digunakan. Perbedaannya penelitian yang akan dilakukan akan melibatkan metode klasifikasi deep learning serta menggunakan dataset teks berbahasa Indonesia

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				dan LR memiliki kinerja terbaik diikuti oleh RF, KNN		
3	A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification	Shahi, T. B., Sitaula, C., & Paudel, N. (2022). Computational Intelligence and Neuroscience	Membandingkan pengaruh dari TF-IDF dan fastText terhadap kinerja hasil klasifikasi dari dataset berbahasa Nepal	metode ekstraksi fitur gabungan mengungguli dua metode ekstraksi fitur individu (TF-IDF dan fastText) saat menggunakan Sembilan algoritma pembelajaran mesin yang berbeda	Masih menggunakan ML tradisional metode untuk evaluasi, perlu pengembangan model deep learning end-to-end menggunakan pendekatan seperti itu mungkin berguna di masa depan untuk dipelajari	Kesamaannya menggunakan TF-IDF namun perbedaannya adalah penelitian yang dilakukan akan menggunakan tweet berbahasa Indonesia dan bukan pada kasus Covid-19
4	Sentiment Analysis of COVID-19 Vaccines in Indonesia on Twitter Using Pre-Trained and Self-Training Word Embeddings	Agustiningsih, K. K., Utami, E., Muhammad, O., & Alysabani, A. (2022). <i>Jurnal Ilmu Komputer Dan Informasi</i> , 15(1), 39–46.	membandingkan 8 skenario dari metode ekstraksi fitur fastText dan Global Vector for Word Representation (GloVe) yang divariasikan berdasarkan kondisi dengan atau tanpa stemming serta berdasarkan kondisi	Model yang menggunakan GloVe menghasilkan akurasi yang lebih tinggi pada dataset tanpa stemming, sedangkan model menggunakan kata fastText memiliki akurasi yang lebih tinggi ketika menggunakan data yang telah mengalami stemming.	Perlunya mempelajari dataset bahasa non-formal seperti deteksi emosi, klasifikasi sentimen atau deteksi sarkasme yang menggunakan penyisipan kata untuk menghasilkan vektor penyisipan kata yang dilatih sendiri. Selain itu, penelitian perlu diuji pada	Dataset yang digunakan adalah data twitter berbahasa Indonesia terkait vaksin covid-19. Perbedaannya adalah kata kunci untuk pencarian data teks yang terkait dengan gejala harga pangan serta tidak memasukkan pengaruh dari stemming

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

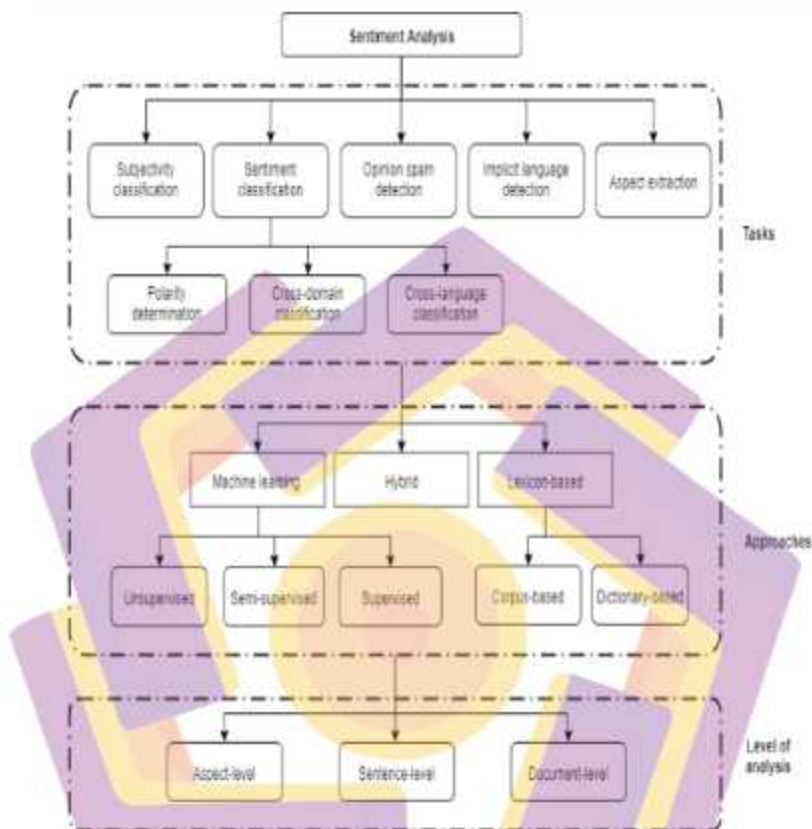
No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			pre-trained dan self-trained.		kumpulan data yang berisi lebih banyak tweet lengkap	
5	A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis	Rustam, F., Khulid, M., Aslam, W., Rupapara, V., Mehmood, A., & Choi, G. S. (2021) <i>PLoS ONE</i>	Membandingkan pengaruh dari TF-IDF, BoW dan penggabungan BoW dan TF-IDF dengan menggunakan beberapa algoritma klasifikasi seperti RF, XGBoost, SVC, ETC dan DT terhadap kinerja klasifikasi	Dari ketiga teknik ekstraksi ciri yang digunakan, TF-IDF, BoW, dan concatenation, pendekatan gabungan memberikan hasil yang lebih baik.	berencana untuk mengarahkan riset ke pendekatan pembelajaran mendalam, khususnya untuk meningkatkan kinerjanya pada kumpulan data kecil.	Perbedaannya dengan penelitian yang akan dilakukan adalah belum menggunakan metode deep learning
6	Improving Bi-LSTM Performance for Indonesian Sentiment Analysis Using Paragraph Vector	Ayu Purwarianti, Ida Ayu Putu Ari Crisdayanti <i>IEEE Xplore</i> : 13 April 2020	Meningkatkan hasil akurasi teks melalui teknik BLSTM serta penggunaan word embedding	Hasil kombinasi BLSTM dan word embedding doc2vec meningkatkan akurasi diatas 90%	Perlu dilakukan beberapa studi kasus tentang data pengujian terkait penggunaan word embedding dan Bi-LSTM	Perbedaan dalam penelitian ini adalah penggunaan rangkaian opini terhadap kebijakan pertanian yang dikeluarkan pemerintah serta melibatkan beberapa metode word embedding dan deep learning

2.3. Landasan Teori

2.3.1 Analisis Sentimen

Metode kecerdasan komputasi memainkan peran sentral dalam analisis sentimen, dan telah terbukti menjadi alat yang ampuh dalam membantu memahami persepsi pelanggan berhubungan dengan produk dan jasa (Farhadloo & Rolland, 2016). Penelitian bidang ini dinilai masih memiliki banyak masalah yang terbuka untuk didalami dan menjadi tantangan penelitian. Meskipun ada banyak kemajuan dari waktu ke waktu namun masih banyak pekerjaan yang harus dilakukan.

Sentimen dalam istilah berarti "apa yang dirasakan seseorang tentang sesuatu", "pengalaman pribadi, perasaan sendiri", "sikap terhadap sesuatu" atau "pendapat". Baik individu maupun organisasi dapat memanfaatkan analisis sentimen dan penambangan opini. Dengan teknik analisis sentimen, kita dapat secara otomatis menganalisis besar jumlah data yang tersedia, dan mengekstrak pendapat yang dapat membantu pelanggan dan organisasi untuk mencapai tujuan mereka. Ini adalah salah satu alasan mengapa analisis sentiment telah menyebar dalam popularitas dari ilmu komputer hingga manajemen dan social ilmu pengetahuan. Bidang kajian analisis sentiment dapat dilihat pada Gambar 2.1.



Gambar 2.1. Peta Kajian Analisis Sentimen (Ligthart et al., 2021)

2.3.2 Text Preprocessing

Langkah pertama dalam klasifikasi sentimen adalah melakukan preprocess teks, proses ini akan membuat data yang tidak terstruktur yang mengandung noise dalam bentuk yang dapat digunakan untuk klasifikasi yaitu dalam bentuk vektor numerik. Pra-pemrosesan melibatkan tugas-tugas seperti tokenisasi, penghapusan

kata yang tidak penting, konversi huruf kecil, stemming, hapus angka, dan sebagainya.

2.3.3 Model Bag of Word (BoW)

Salah satu model untuk mewakili kata melalui penggunaan angka atau vector adalah dengan menggunakan model *Bag of Word* atau dikenal dengan singkatan BoW. Model ini menurut (Brownlee, 2017) merupakan model yang merepresentasikan teks berdasarkan kejadian munculnya berbagai kata dalam sebuah dokumen dan terdiri atas dua hal yaitu kamus yang berisi kata yang telah diketahui serta sebuah ukuran kemunculan kata yang telah diketahui tersebut. Disebut sebagai “bag” karena memang dalam model ini tidak memperhatikan urutan atau struktur dari kata-kata dalam dokumen (Brownlee, 2017). Hal ini ditegaskan bahwa metode representasi berdasarkan model BoW mengabaikan informasi sekuensial dan struktural, yang dapat menjadi penting untuk memahami makna semantic dari kalimat (Liu et al., 2020). Sebelum adanya teknik deep learning (DL) untuk representasi teks kalimat biasanya direpresentasikan sebagai vektor one-hot atau vektor *term frequency – inverse document frequency* (TF-IDF), mengikuti asumsi *bag-of-words* (Liu et al., 2020). (Goldberg, 2017) Prosedur ekstraksi fitur yang sangat umum untuk kalimat dan dokumen adalah pendekatan BoW dimana pendekatan ini dilihat berdasarkan frekuensi setiap kata atau word count sebagai fitur (Goldberg, 2017).

Pendekatan BoW pada awalnya hanya dilakukan berdasarkan representasi vector biner yang hanya berisi angka 0 dan 1 yang disebut dengan one-hot representation/ Formula yang digunakan dapat dilihat pada (1) dan (2).

$$V = \{w_1, w_2, \dots, w_{|V|}\} \dots\dots\dots(1)$$

$$w = [0, \dots, 0, 1, 0, \dots, 0] \dots\dots\dots(2)$$

dimana V adalah kamus (vocabulary) dari kata-kata yang digunakan dan w adalah one-hot representation dimana kode 1 menunjukkan bila dokumen yang dimaksud mengandung kata yang terdapat dalam kamus dan kode 0 bila dokumen tidak mengandung kata yang terdapat dalam kamus.

Pengembangan dari one-hot representation ini adalah dengan cara menghitung frekuensi kemunculan katanya dalam kalimat dan bukan hanya berdasarkan kode 1 dan 0 seperti dapat dilihat pada persamaan (3).

$$s = \sum_{k=1}^l w_k \dots\dots\dots(3)$$

di mana l menunjukkan panjang kalimat s . Representasi kalimat s adalah jumlah representasi *one-hot* dari n kata dalam kalimat, yaitu, setiap elemen di s mewakili *Term Frequency* (TF) dari kata yang sesuai.

Kemudian *Inverse Document Frequency* (IDF) digunakan untuk mengukur pentingnya kata di V pada persamaan (4) :

$$idf_{w_i} = \log \frac{|D|}{df_{w_i}} \dots\dots\dots(4)$$

dimana $|D|$ adalah jumlah semua dokumen dalam korpus D dan idf_{w_i} mewakili

Frekuensi Dokumen dari w_i . Pada akhir nilai tf-idf dihitung dengan rumus pada persamaan 5 (Liu et al., 2020):

$$\hat{s} = s \times idf_{w_i} \dots \dots \dots (5)$$

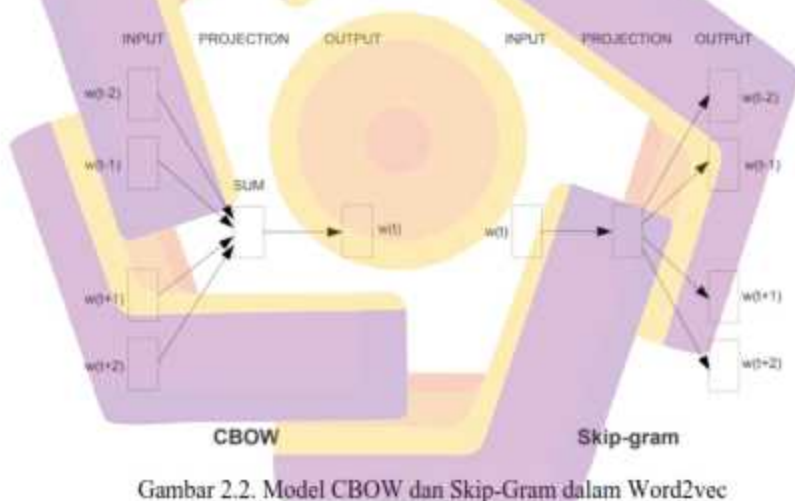
Representasi bag-of-words ini bersifat sederhana namun efisien dan terkadang dapat mencapai kinerja yang sangat baik di banyak aplikasi dunia nyata. Namun kelemahannya adalah representasi bag-of-words masih mengabaikan sepenuhnya informasi urutan kata, yang berarti dokumen yang berbeda dapat memiliki representasi yang sama asalkan sama kata-kata digunakan. Selain itu, representasi bag-of-words memiliki sedikit pengertian tentang semantik kata-kata, lebih formal, jarak antara kata-kata, yang berarti metode ini tidak dapat memanfaatkan informasi kaya yang tersembunyi dalam representasi kata (Liu et al., 2020).

2.3.4 Word Embedding

Word Embedding (WE) diartikan sebagai representasi terdistribusi dari kata-kata di mana kata-kata berbeda itu memiliki arti yang sama (berdasarkan penggunaannya) juga memiliki representasi yang serupa (Brownlee, 2017). Konsepnya adalah memiliki arti yang sama maka akan memiliki representasi yang serupa. Terdapat beberapa metode atau algoritma yang termasuk dalam kelompok ini antara lain adalah word2vec dan fasttext. Word2Vec adalah metode statistik untuk mempelajari penyematan kata mandiri secara efisien dari korpus teks. Ini

dikembangkan oleh Tomas Mikolov. Model ini kemudian dikembangkan kembali dengan nama fasttext (Bojanowski et al., 2017). Prinsip utamanya adalah menggunakan struktur internal sebuah kata untuk meningkatkan representasi vektor yang diperoleh dari metode skip-gram.

Word2vec memiliki 2 model yaitu *Continuous Bag-of-Words* atau model CBOW dan model *Continuous Skip-Gram* seperti dapat dilihat pada Gambar 2.2. Model CBOW mempelajari embedding dengan memprediksi kata saat ini berdasarkan konteksnya. Model skip-gram terus menerus belajar dengan memprediksi kata-kata di sekitarnya yang diberikan arus kata.

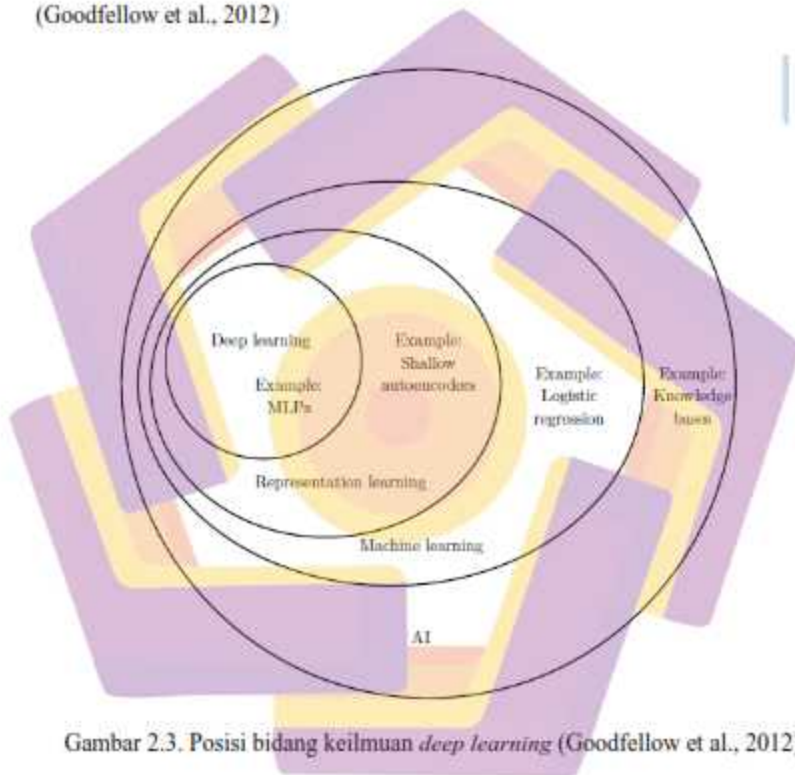


Gambar 2.2. Model CBOW dan Skip-Gram dalam Word2vec

2.3.5 Algoritma Pengklasifikasian

Klasifikasi teks antara lain dalam bidang analisis sentiment antara lain dapat dilakukan melalui berbagai algoritma pembelajaran mesin atau machine learning (ML) seperti terdapat pada Gambar 2.3. ML sendiri seperti dapat dilihat pada

Gambar 2.3 merupakan bagian dari teknik kecerdasan buatan atau artificial intelligence (AI). Sedangkan posisi deep learning (DL) dalam bidang keilmuan AI disebutkan bahwa DL adalah himpunan bagian dari topic Representation learning serta salah satu contoh model dalam DL adalah Multi Layer Perceptron (Goodfellow et al., 2012)



Gambar 2.3. Posisi bidang keilmuan *deep learning* (Goodfellow et al., 2012)

Beberapa algoritma ML yang akan dijelaskan disini adalah naïve bayes (NB), support vector machine (SVM), k-nearest neighbor (k-NN) dan regresi logistic atau logistic regression (LR). Sedangkan algoritma DL yang akan dijelaskan adalah multi layer perceptron (MLP), convolutional neural network (CNN), long short term memory (LSTM) dan bidirectional LSTM (BLSTM).

2.3.6 Algoritma Pembelajaran Mesin

Naïve Bayes merupakan sebuah algoritma pembelajaran sederhana terkait pengklasifikasian yang memanfaatkan *Bayes Rule*. Misalkan terdapat contoh yang akan diklasifikasikan dijelaskan oleh $x = (x_1, \dots, x_n)$ maka prinsip algoritma tersebut adalah sebagai berikut (Kubat, 2017):

- i. Untuk setiap x_i pada setiap kelas c_j hitung probabilitas bersyarat, $P(x_i|c_j)$, sebagai frekuensi relative dari x_i di antara data latih yang termasuk dalam c_j .
- ii. Untuk setiap kelas, c_j , lakukan dua langkah berikut:
 - i) menduga nilai $P(c_j)$ sebagai frekuensi relatif kelas ini dalam set data latih;
 - ii) hitung peluang bersyarat, $p(x_i|c_j)$, dengan menggunakan asumsi "naïve" dari kejadian yang saling bebas dengan rumus:

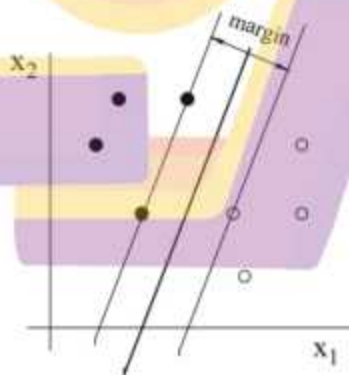
$$P(x|c_j) = \prod_{i=1}^n P(x_i|c_j) \dots\dots\dots(6)$$

- iii. Pilih nilai tertinggi dari $P(c_j) \cdot P(x|c_j)$

Algoritma k-NN secara sederhana dalam (Kubat, 2017). Misalkan kita memiliki mekanisme untuk mengevaluasi kesamaan antara vektor atribut. Obyek x menunjukkan objek yang kelasnya ingin kita tentukan.

1. Di antara contoh pelatihan, identifikasi k tetangga terdekat dari x (contoh paling mirip ke x).
2. Biarkan c_i menjadi kelas yang paling sering ditemukan di antara k tetangga terdekat ini.
3. Beri label x dengan c_i .

Teknik pengklasifikasi berikutnya adalah support vector machine (SVM) dengan ilustrasi seperti dapat dilihat pada Gambar 2.4. Garis tebal adalah pengklasifikasi terbaik. Grafik menunjukkan juga dua garis tipis, sejajar masing-masing pada jarak yang sama. Garis tersebut juga melewati titik contoh yang paling dekat dengan garis pemisah. Titik-titik contoh ini disebut vektor pendukung (support vector). Tugas pembelajaran mesin adalah mengidentifikasi vektor pendukung yang memaksimalkan batas (Kubat, 2017).

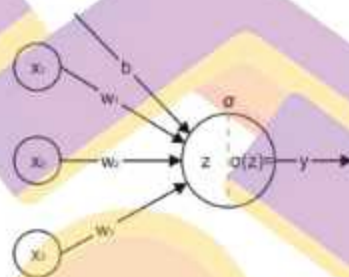


Gambar 2.4. Teknik dari SVM untuk Hyperplane emisah yang memiliki margin maksimum

Model regresi logistic sesungguhnya adalah skema sederhana dari jaringan syaraf tiruan seperti dapat dilihat pada Gambar 2.5. Model tersebut memiliki persamaan seperti dapat dilihat pada persamaan 7 dan 8.

$$z = b + w_1x_1 + w_2x_2 + w_3x_3 \dots\dots\dots(7)$$

$$z = \sigma(z) = \frac{1}{1 + e^{-z}} \dots\dots\dots(8)$$



Gambar 2.5. Skema Model Regresi Logistic (Skansi, 2020)

2.3.7 Algoritma Pembelajaran Mendalam

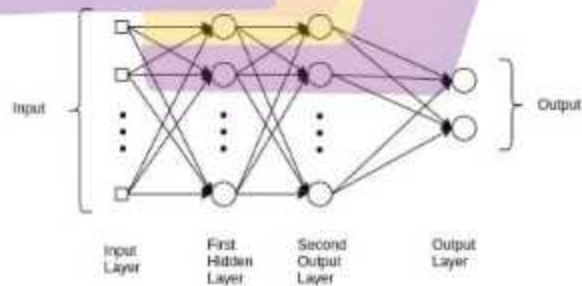
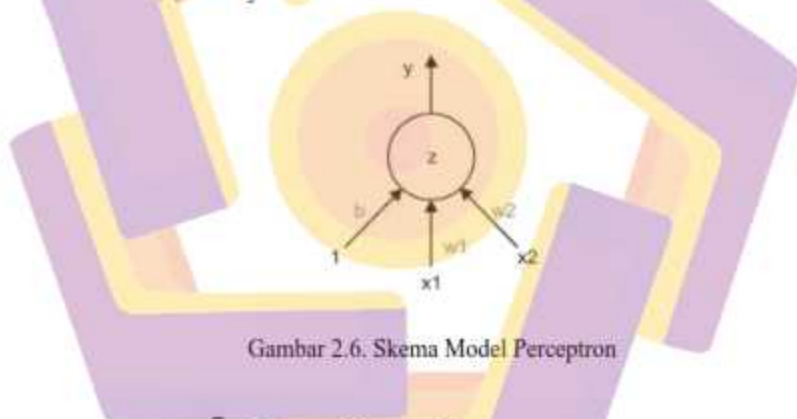
Algoritma pembelajaran mendalam atau deep learning (DL) antara lain adalah Multi layer perceptron (MLP) yang merupakan pengembangan dari model perceptron dengan aturan sebagai berikut (Skansi, 2020) :

1. *Pilih kasus pelatihan.*
2. *Jika keluaran yang diprediksi cocok dengan label keluaran, jangan lakukan apa pun.*
3. *Jika perceptron memprediksi 0 dan seharusnya memprediksi 1, tambahkan vektor input ke vektor bobot*
4. *Jika perceptron memprediksi 1 dan seharusnya memprediksi 0, kurangi vektor input dari vektor bobot*

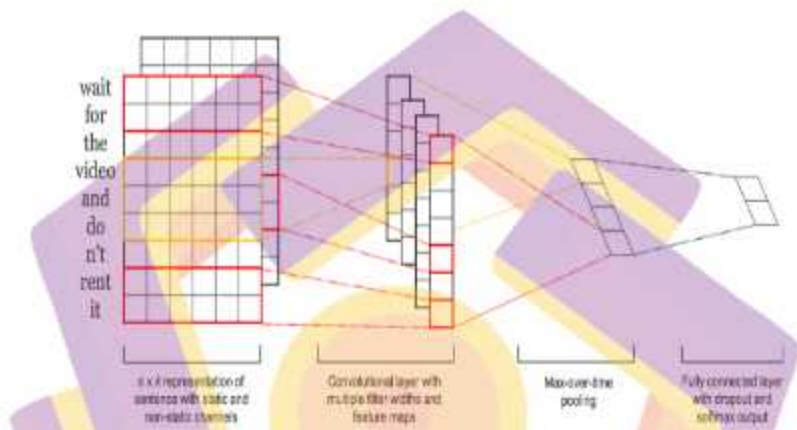
Persamaan matematis yang digunakan dapat dilihat pada 9 dan 10. Persamaan ini digunakan sesuai skema yang dapat dilihat pada Gambar 2.6. Sedangkan pengembangan dari model perceptron menjadi model MLP dengan melibatkan beberapa layar tersembunyi serta menggunakan algoritma back propagation dapat dilihat pada Gambar 2.7.

$$z = b + \sum_i w_i x_i \dots (9)$$

$$y = \begin{cases} 1, & z \geq 0 \\ 0, & \text{lainnya} \end{cases} \dots (10)$$

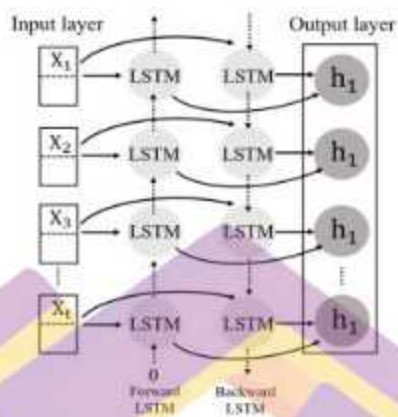


Model Convolutional Neural Network (CNN) merupakan sebuah deep learning lainnya yang dapat digunakan untuk klasifikasi baik teks maupun gambar. Cara kerja model dapat dilihat pada Gambar 2.8.

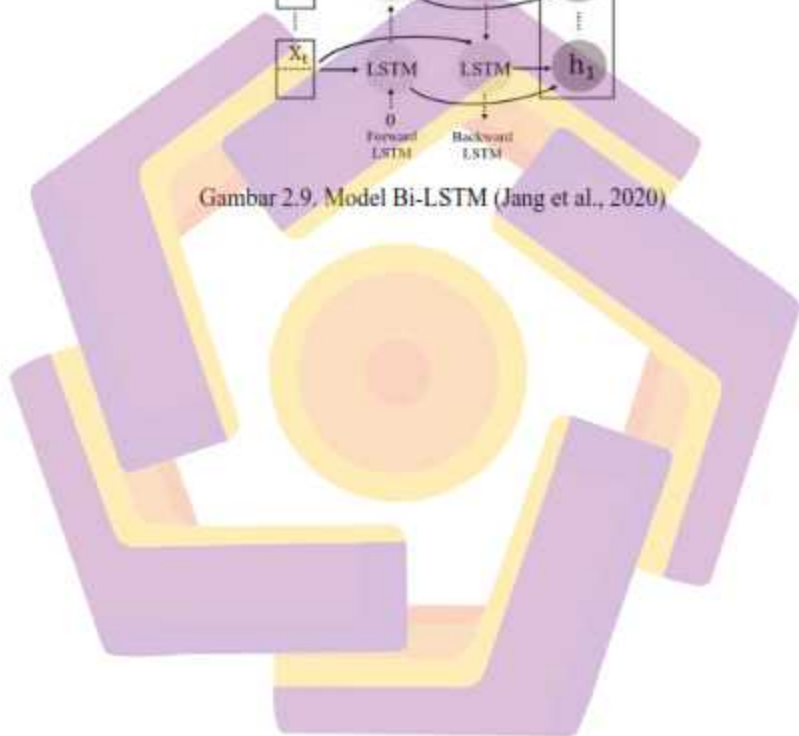


Gambar 2.8. Model CNN untuk klasifikasi teks (Brownlee, 2017)

Model pembelajaran mendalam lainnya yang digunakan dalam penelitian ini adalah model LSTM serta pengembangannya yaitu bidirectional LSTM seperti dapat dilihat pada Gambar 2.9. Model LSTM dapat menangkap ketergantungan jangka panjang antara urutan kata karenanya lebih baik digunakan untuk klasifikasi teks. Sedangkan jaringan Bi-LSTM memiliki dua lapisan paralel yang merambat dalam dua arah dengan maju dan mundur lolos untuk menangkap dependensi dalam dua konteks (Jang et al., 2020).



Gambar 2.9. Model Bi-LSTM (Jang et al., 2020)



BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

3.1.1 Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian kuantitatif yang didasarkan pada dataset yang ditarik dari database twitter. Sifat bersifat kuantitatif dikarenakan proses pengolahan data dilakukan menggunakan model matematis baik ketika ekstraksi fitur maupun proses klasifikasi sentiment.

3.1.2 Pendekatan Penelitian

Pendekatan penelitian yang digunakan adalah pendekatan eksperimen dimana beberapa factor diuji coba pengaruhnya terhadap suatu respon. Faktor dalam hal ini adalah ekstraksi fitur serta algoritma klasifikasi. Respon adalah kinerja hasil klasifikasi.

3.2. Metode Pengumpulan Data

Penelitian ini menggunakan 2 dataset terkait komoditas minyak goreng yang terbagi atas 2 dataset berbasis data twitter. Dataset twitter pertama diambil menggunakan batasan waktu 21-25 Maret 2022 dengan kata kunci minyak goreng. Pada periode itu adalah berlangsungnya kebijakan pemerintah menghapus Harga Eceran Tertinggi (HET) dimulai 16 Maret 2022. Dataset kedua terkait kebijakan Bantuan Langsung Tunai (BLT) minyak goreng sebagai dampak kenaikan harga minyak goreng yang diberikan mulai April 2022.

3.2.1 Metode Analisis Data

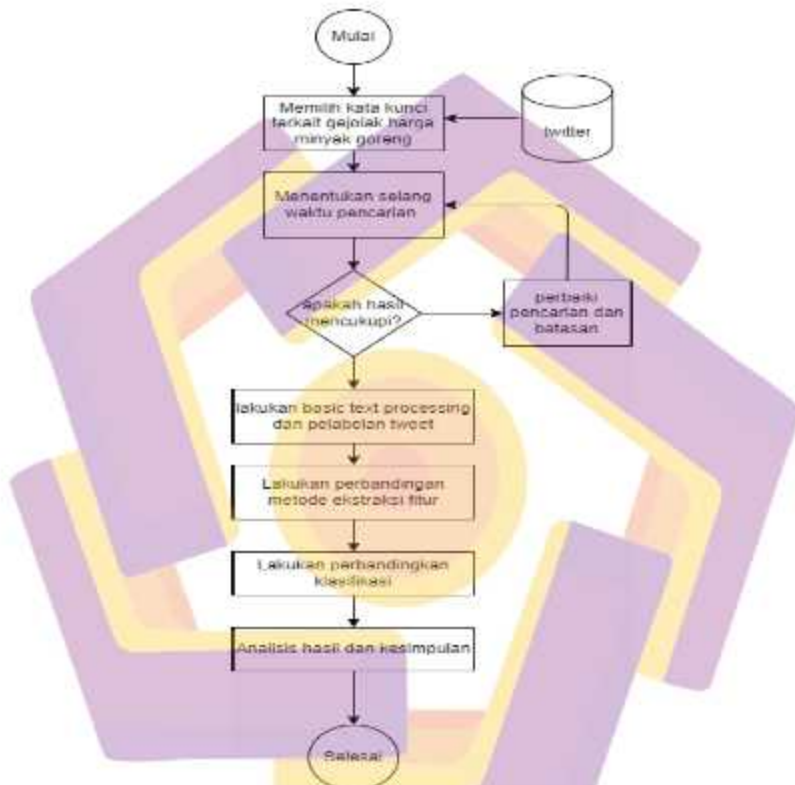
Metode analisis data didasarkan pada beberapa teknik prerprocessing yang digunakan termasuk pada proses pelabelan sentiment baik secara manual maupun secara otomatis menggunakan kamus kata positif dan negative serta menggunakan text blob yang diterjemahkan dalam bahasa Indonesia. Analisis kuantitatif juga dilakukan untuk membandingkan tiga ekstraksi fitur yang digunakan yaitu BoW, TF-IDF serta word2 vec serta membandingkan beberapa algoritma klasifikasi baik yang konvensional maupun menggunakan teknik deep learning.

3.2.2 Alur Penelitian

Alur penelitian yang digunakan pada penelitian ini ditunjukkan pada Gambar 1. Tahapan pada penelitian terbagi menjadi beberapa tahapan yaitu pertama adalah melakukan crawling data dari twitter menggunakan kata kunci tertentu seperti dapat dilihat pada Gambar 9. Pengumpulan data dilakukan dengan menggunakan fitur API yang telah disediakan oleh Twitter Developer. Sebelum dapat mengakses API, diperlukan pendaftaran terlebih dahulu di Twitter Developer. Setelah berhasil terdaftar, Twitter Developer akan memberikan akses API Key, API Key Secret, Access Token dan Access Token Secret.

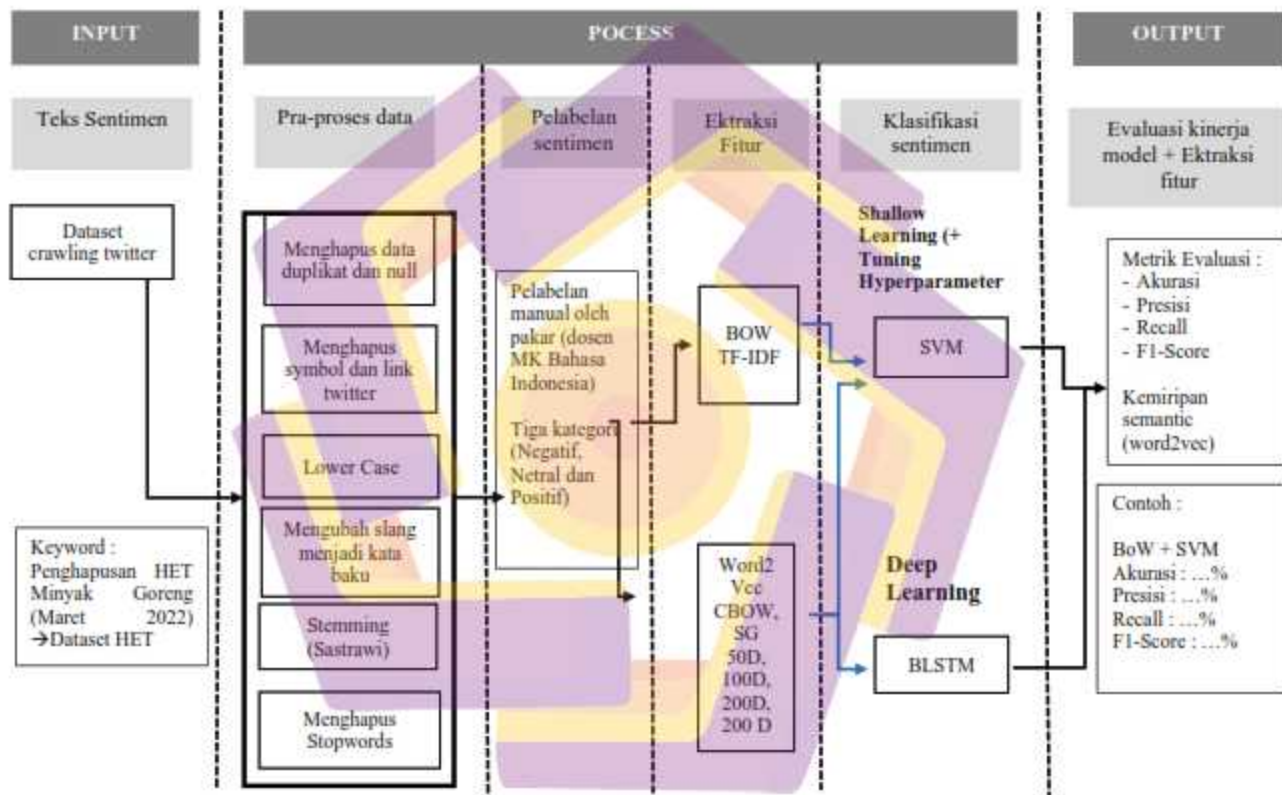
Eksperimen yang kedua dilakukan dengan mengkombinasikan beberapa metode ekstraksi fitur dengan metode klasifier. Ekstaksi fitur yang dibandingkan adalah TF-IDF dan BOW (Bag Of Word). Klasifier yang dibandingkan adalah K-Nearest Neighbors dan SVM yang berasal dari

teknik non deep learning serta teknik ANN, LSTM dan BiLSTM yang berasal dari teknik deep learning. Rencana solusi dapat dilihat pada Gambar 3.1.

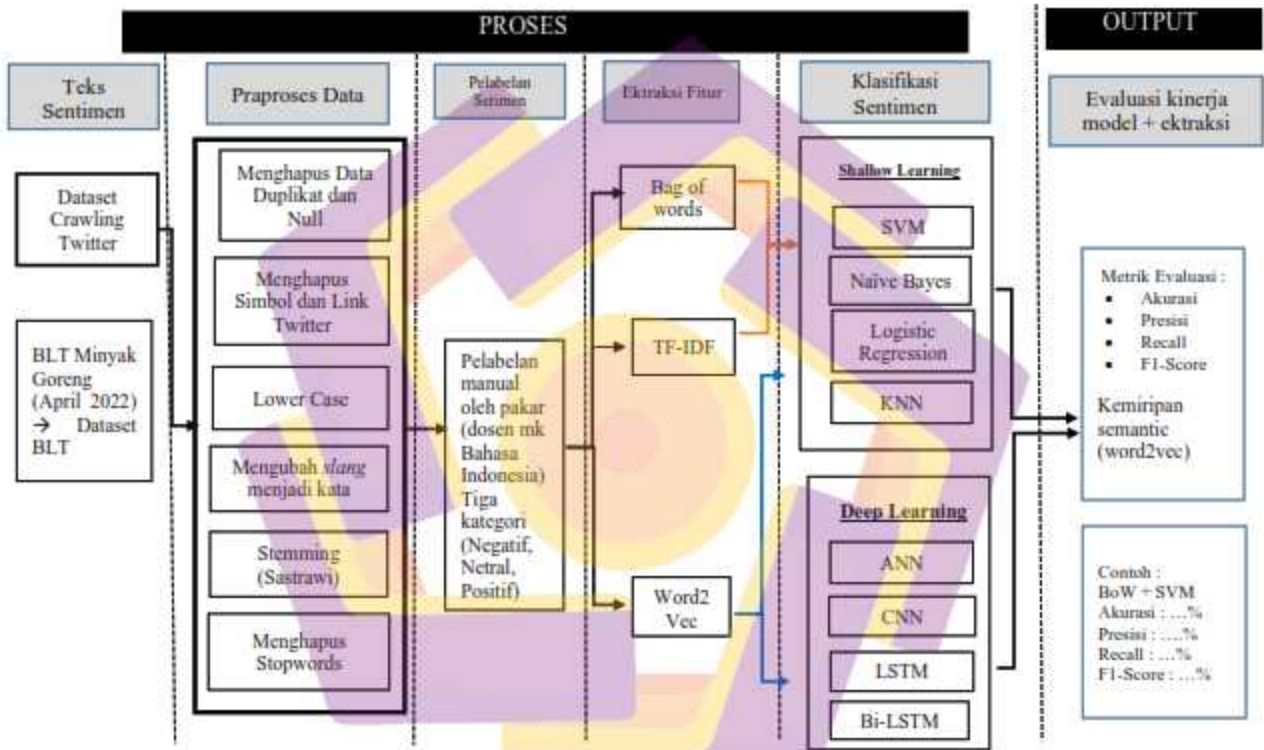


Gambar 3.1. Diagram Alur Penelitian

Rancangan eksperimen terbagi atas 2 rancangan seperti dapat dilihat pada Gambar 3.2 dan 3.3.



Gambar 3.2 Rancangan 1



Gambar 3.3 Rancangan 2

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pra-pemrosesan Dataset Penghapusan Harga Eceran Tertinggi (HET)

Proses crawling menghasilkan data tweet sebanyak 15001 tweet (Gambar 4.1) dengan kata kunci minyak goreng dan setelah melalui tahapan pra-pemrosesan tersisa menjadi 12408 tweet (Gambar 4.2). Contoh pelabelan secara manual dapat dilihat pada Gambar 4.3 dengan persentase yang dapat dilihat pada Gambar 4.4.



	tweet
0	@u_satsy lapi di tali minyak goreng gak langka om
1	@BangkitRoses Minyak goreng dibuat bahan bakar? 🤔
2	@Helmi_Fahri @Jokowi Kebutuhan negara hancur... pertahankan minyak goreng saja gk mampus. oh ya coba tanya mnti pertahatan si @rebrown
3	@geloraco Pak purni bawa sekalian minyak goreng ke Indonesia, ajarin tata cara mengolah negara ig 🇲🇵🇲🇵
4	Sudah dibombardir dgn berbagai sanksi: Kenaikan BPJS, Listrik, LPG, Tol, BBM, PPN, kenaikan harga Sembako, kenaikan harga Minyak goreng sampe 70%inMasih akan dibeban juga untuk urusan dana IKN, 77%inFakir dong yang 11 ribut dikandung in/n%Danabatakanin/n/n%Danabatakanin/n https://t.co/VtdE166ase
...	...
14996	Polisi mengorek gudang minyak gorengin#AjukanKebutuhanPanganinBangkit Tumbuh Ekonomi https://t.co/z9SLVMDUKA
14997	Ketersediaan minyak goreng harus terus dipantau di pasaranin#AjukanKebutuhanPanganinBangkit Tumbuh Ekonomi https://t.co/96AuSBdpl
14998	Antisipasi Kelangkaan Minyak GorenginPolres Minahasa melalui Polsek jajaran melaksanakan monitoring ketersediaan minyak goreng di pasaran Pedagang Mikro hingga Sosialayin/n%Humaspolresminahasa in%polresminahasa https://t.co/Cin8LYR0CK
14999	Urut mengolah penjualan di pasar Kapoli instruksikan kapolda dan jajarannya terkait ketersediaan minyak goreng #AjukanKebutuhanPanganinBangkit Tumbuh Ekonomi https://t.co/0xUYTsqJv
15000	Jokowi Selalu Dhaharinhttps://t.co/yD4gqkph/n/nMinyak goreng bersubsidi telah tersedia

15001 rows • 1 columns

Gambar 4.1. Contoh tweet hasil crawling dataset HET

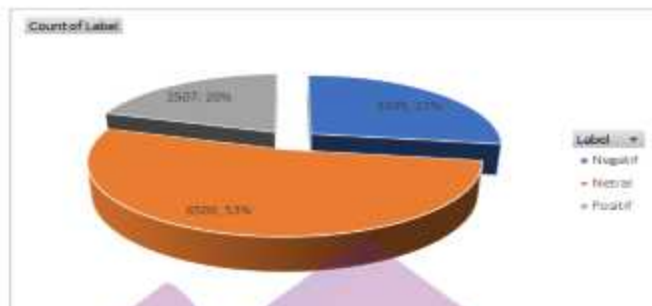
	Label	comment
0	Negatif	lali minyak goreng gak langka an
1	Netral	minyak goreng bahan bakar
2	Negatif	daulat negara hancur lahan minyak goreng gk nya coba di lahan si
3	Negatif	putri bawa minyak goreng Indonesia ajah tata kelola negara
4	Negatif	bombardir serang bpjs listrik lg tol bmn dgn harga semblan bahan pokok harga minyak goreng diobarin unun dana iku palak karling
...
12403	Netral	polisi menggerek gudang minyak goreng bangkit tumbuh ekonomi
12404	Netral	sedia minyak goreng panto pasar bangkit tumbuh ekonomi
12405	Netral	ardiansah bilang minyak goreng pntes minahasa pntek jejer laksana monitoring sedia minyak goreng kalang daging mikro selayan
12406	Positif	bagah jual pasar kapoh instruksi kapoda jejer kal sedia minyak goreng bangkit tumbuh ekonomi
12407	Positif	pakewi hat minyak goreng subsidi sedia
12408 row = 2 columns		

Gambar 4.2. Contoh tweet yang telah melewati tahap pra-proses pada dataset HET

index	Label	comment
0	Negatif	lali minyak goreng gak langka an
1	Netral	minyak goreng bahan bakar
2	Negatif	daulat negara hancur lahan minyak goreng gk nya coba di lahan si
3	Negatif	putri bawa minyak goreng Indonesia ajah tata kelola negara
4	Negatif	bombardir serang bpjs listrik lg tol bmn dgn harga semblan bahan pokok harga minyak goreng diobarin unun dana iku palak karling
5	Negatif	minyak goreng lunas harga jagung melambung
6	Netral	pc nandani ulama ngada dusung bjak pemerintah subsidi minyak goreng curah
7	Netral	pemntah subsidi minyak goreng curah Indonesia esang
8	Netral	jalang nansahan kapoh cak sedia hat minyak goreng curah pasar eshang
9	Netral	bahuk nantun kupa mudi ac langit atagfukahi na utublahimzash

Gambar 4.3. Contoh hasil pelabelan sentiment dataset HET

Terlihat pada Gambar 4.4 bahwa data bersifat tidak seimbang (imbalance dataset) dimana terbesar adalah kategori netral (53%) disusul oleh kategori negative (27%) dan kategori positif (20%).



Gambar 4.4. Diagram lingkaran kategori sentiment pada dataset HET

4.2. Pra-pemrosesan Dataset Pemberian Bantuan Langsung Tunai (BLT)

Proses crawling menghasilkan tweet sebanyak 5058 dan setelah melewati tahap pra proses menjadi tersisa 1335 tweet seperti dapat dilihat pada Gambar 4.5 dan Gambar 4.6. Hal ini karena telah dilakukan pembersihan data yang duplikat atau tidak memiliki nilai (null). Terlihat pada Gambar 18 bahwa data teks sudah dalam huruf kecil dan sudah dalam bentuk kata dasar. Sebagai contoh kata kata “habiskan” dalam baris pertama di Gambar 4.5 sudah berubah menjadi kata “habis” pada Gambar 4.6. Begitu pula kata “menyalurkan” pada baris kedua di Gambar 4.5 sudah berubah menjadi kata “salur” pada Gambar 4.6. Berikutnya adalah pelabelan sentiment menjadi 3 kategori yang dilakukan secara manual dengan hasil seperti dapat dilihat pada Gambar 4.7

	comment
0	b'RT @iputan6dotcom: BLT Minyak Goreng Habiskan Rp 5,9 Triliun, Kenapa Tak Bentuk BUMN CPO? https://t.co/kfhzds56N
1	b'ke2v00v09cKami mengapresiasi langkah pemerintah untuk menyalurkan BLT minyak goreng, terutama di Bulan Ramadan dan menjelang Lebaran, di mana kebutuhan masyarakat terhadap berbagai komoditas, termasuk minyak goreng, cukup tinggi.ve2v00v09d –Puan Maharani'
2	b'Ketua DPR Puan Maharani mengapresiasi langkah pemerintah menyalurkan Bantuan Langsung Tunai (BLT) minyak goreng.'
3	b'RT @iputan6dotcom: BLT Minyak Goreng Habiskan Rp 5,9 Triliun, Kenapa Tak Bentuk BUMN CPO? https://t.co/kfhzds56N
4	b'RT @iputan6dotcom: BLT Minyak Goreng Habiskan Rp 5,9 Triliun, Kenapa Tak Bentuk BUMN CPO? https://t.co/kfhzds56N
...	...
5053	b'RT @democracymedia: PKS Bongkar Sumber Dana BLT Minyak Goreng Jokowi. Ternyata Uangnya Dari Anggaran Bansos! https://t.co/RbLAz0EvMS
5064	b'RT @Ransor: Presiden @jokowi Serahkan BLT Minyak Goreng di Pasar Angko Duo, Jambi, 7 April 2022. https://t.co/jokowi Gaspol pemulihan Ekonomi https://t.co/7xe2v00v09d
5065	b'RT @hunnwahid: @FPKSDPRI Bongkar Sumber Dana BLT Minyak Goreng Jokowi. Ternyata Uangnya Dari Anggaran Bansos DEMOCRACY News Indonesia/ve2v00v09d
5066	b'RT @WingDewa: Saat Jokowi ke Pasar Angko Duo Kota Jambi, ulk mberikan BLT Minyak Goreng Kemas 7-4-22. seorang pria membawa poster dan ingkve2v00v09d
5067	b'RT @maspiyujaja: Serikat Buruh Tolak BLT Minyak Goreng: Yang Dibutuhkan Harga Minyak Goreng Turun, BLT Sama Saja Tunduk Pada Taipen Pemilik! ve2v00v09d

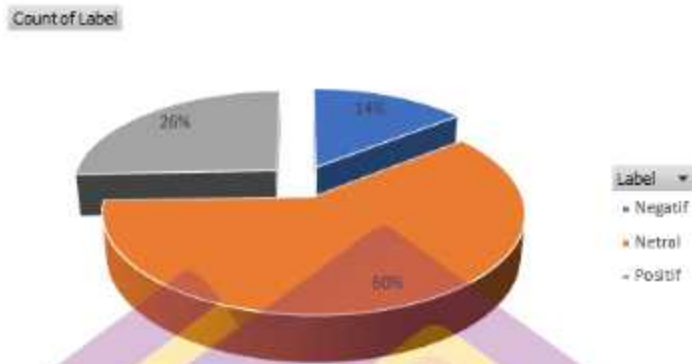
5056 rows x 1 columns

Gambar 4.5. Contoh tweet hasil crawling dataset BLT

unnamed:	comment
0	blt minyak goreng habis rp triliun bentuk bumncpo
1	apresiasi langkah perintah salur blt minyak goreng utama ramadan jelang lebaran butuh masyarakat komoditas minyak goreng puan maharani
2	ketua dpr puan maharani apresiasi langkah perintah salur bantu tunai blt minyak goreng
3	blt minyak goreng rakyat oligarki lengkap aliansi usaha muslim
4	puan maharani apresiasi kali langkah perintah salur blt minyak goreng
...	...
1331	bantu pantengin hari blt minyak goreng kali lan perintah part
1332	blt minyak goreng wow muncul bantu bantu periode
1333	bantu moga ringan subsidi masyarakat utama dagang kaki jual goreng tulis jokowi laman instagram pribadi kemas
1334	presiden joko widodo bantu tunai blt minyak goreng dagang ferima pa
1336	desember gada solusi blt minyak goreng bajak gin rya memprotek krj kartel

1336 rows x 2 columns

Gambar 4.6. Contoh tweet yang telah melewati tahap pra-proses dataset BLT



Gambar 4.10. Diagram lingkaran kategori sentiment pada dataset HET 2

	bit	selur	bantu	tunai	apresiasi	tunai
0	1	0	1	1	0	1
1	1	0	0	0	0	0
2	1	1	0	0	0	0
3	1	0	0	0	0	0
4	1	0	0	0	0	0
...						
1196	1	0	1	1	0	1
1197	1	1	1	1	0	1
1198	1	0	0	0	0	0
1199	1	0	0	0	0	0
1200	2	1	1	1	1	1

Gambar 4.11. Contoh nilai ekstraksi fitur BOW

Hasil ekstraksi fitur untuk data latih (train) dengan komposisi 90% dari dataset dapat dilihat pada Gambar 4.11 untuk model BOW dan Gambar 4.12 untuk model TF-IDF. Dapat dilihat pada Gambar 4.11 untuk kata kunci "BLT" sebagian besar muncul sekali dalam dokumen tweet namun pada contoh dokumen ke-1200 tercatat frekuensi kemunculannya sebanyak dua kali. Selanjutnya kedua matriks vector dari data train yang didapatkan menggunakan model BOW dan TF-IDF akan

dimasukkan ke dalam algoritma klasifikasi shallow learning untuk melatih model klasifikasi.

	blt	salur	bentu	tunai	apresiasi	tunai
0	0.119112	0.000000	0.271250	0.333895	0.000000	0.333895
1	0.098110	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.038499	0.097493	0.000000	0.000000	0.000000	0.000000
3	0.089007	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.086537	0.000000	0.000000	0.000000	0.000000	0.000000
...
1196	0.112804	0.000000	0.256883	0.316211	0.000000	0.316211
1197	0.086206	0.218304	0.196313	0.241652	0.000000	0.241652
1198	0.179248	0.000000	0.000000	0.000000	0.000000	0.000000
1199	0.110921	0.000000	0.000000	0.000000	0.000000	0.000000
1200	0.108833	0.137802	0.123920	0.152540	0.262625	0.152540

Gambar 4.12. Contoh nilai ekstraksi fitur TF-IDF

4.3. Ekstraksi Fitur Word2vec pada Dataset HET

Langkah selanjutnya setelah melakukan pelabelan secara manual adalah memotong jumlah kata dalam kalimat. Pemotongan ini berfungsi untuk menyeragamkan jumlah kata pada setiap kalimat agar sesuai dengan ukuran input model machine learning. Jumlah kata yang ditentukan diperoleh dari rata-rata jumlah kata dalam semua kalimat. Dalam hal ini, angka rata-rata adalah 10,64 kata dan dibulatkan menjadi 11 kata. Ukuran kamus sebanyak 12238 kata unik. Contoh kata dengan indeks sampel 1 sampai 15 dapat dilihat pada Tabel 4.1.

Tabel 4.1. Contoh kamus data pada dataset HET

Index	Word
1	goreng
2	minyak

Tabel 4.1. Contoh kamus data pada dataset HET (lanjutan)

3	barga
4	langka
5	mafia
6	mahal
7	curah
8	pasar
9	iya
10	perintah
11	rakyat
12	mendag
13	beli
14	stok
15	Sedia

Tweet pada setiap dokumen akan dikonversi menjadi nilai indeks dengan contoh seperti yang dapat dilihat pada Tabel 4.2. Terlihat bahwa karena jumlah kata kurang dari 11 maka diberikan *padding value* 0 di awal. Terlihat bahwa pada kalimat 'pks tagih janji mendag nama mafia minyak goreng via' terdapat kata 'mendag' yang memiliki indeks 12, "mafia" dengan indeks 5, "minyak" dengan indeks 2 dan "goreng" memiliki nilai indeks 1 yang sesuai dengan nilai pada Tabel 4.1.

Tabel 4.2. Contoh kamus dan nilai index

Tweet	Index
pks tagih janji mendag nama mafia minyak goreng via	[0 0 401 760 128 12 13 8 5 2 1 240]
cari uang beli minyak goreng bung	[0 0 0 0 0 44 161 13 2 1 213]
menperin wajib sedia minyak goreng curah masyarakat umkm menteri industri kemenperin atur menteri industri	[434 104 21 79 431 629 15 2 1 7 30]

Tabel 4.2. Contoh kamus dan nilai index (lanjutan)

Tweet	Index
permenperin nomor sedia minyak goreng curah butuh	

Dua contoh vector yang didapatkan dari arsitektur atau model CBOW dengan dimensi 50 dan *window size* 5 dapat dilihat pada Tabel 4.3. Vektor berukuran 50 dan didapatkan dengan menggunakan model CBOW. Kata 'mafia' dan 'kartel' yang terdapat dalam dokumen tweet dapat ditangkap oleh model CBOW dengan similaritas yang cukup tinggi yaitu sebesar 0.642 (Gambar 4.13). Gambar 4.14 memperlihatkan nilai similaritas untuk window size 5 dengan dimensi yang bergerak dari 50, 100, 200 dan 300. Terlihat bahwa model SG memiliki nilai similaritas yang lebih tinggi dibandingkan CBOW pada dimensi yang rendah. Hal ini sejalan dengan penelitian dari (Jatnika et al., 2019) dan (Khomsah, 2021) yang menyebutkan bahwa nilai kemiripan juga akan semakin rendah atau semakin lemah dengan bertambahnya dimensi vector (Gambar 4.15).

Tabel 4.3. Contoh vector kata dari kata 'mafia' dan 'kartel'

Word	Word Vector CBOW 50D
mafia	array([-0.56892875, 2.0289217, -1.1227585, -1.1525702, 0.8419746, 0.5586357, -0.6068408, 2.1427064, 2.1188006, 1.313243, -0.04043027, 1.5766977, 3.1851795, -1.9543221, -1.7411065, 0.41647932, -2.2227454, 2.048298, 2.319479, -0.6260029, 0.2707395, 1.0605536, -3.5908034, 1.5332314, -1.0886542, -2.1656687, 0.16630426, 1.9131241, 2.0570612, -1.1507548, 0.19240206, 0.8448768, 0.6710651, -2.8331344, -1.4188331, 1.8492148, -0.14567924, -1.3153963, 1.554626, 1.6846476, -0.6650871, 0.1740221, -0.71855783, 2.0238554, -

Tabel 4.3. Contoh vector kata dari kata 'mafia' dan 'kartel' (lanjutan)

Word	Word Vector CBOW 50D
	0.26927564, 1.6553223, 1.03337701, 0.21720444, -1.7952678, -1.2687485], dtype=float32)
kartel	array([-2.5225072, 0.58414376, 1.6165864, -2.9735773, 1.1047269, -0.07021898, - 1.597327, 2.5603216, -1.3518994, 2.8366055, -0.5908015, -1.0138676, 4.9474783, -4.0722523, -2.1096187, 0.12271009, -1.6168385, -0.74996346, 0.36694336, -0.203116, -2.2861948, 1.1278254, -6.615916, -1.8414071, 0.5299412, 0.19319774, -2.0714629, 1.8133492, 3.2557642, -1.2094542, 2.1729934, 0.48804253, -0.40780009, - 1.1330132, -2.0840626, 1.2450163, 0.1762628, -2.58235, 1.7960441, 1.7053685, -1.8262835, -1.0867745, 1.2517663, 0.2910628, -0.4897558, 3.586528, 0.45648718, -0.9132699, - 0.7286088, -0.22194196], dtype=float32)

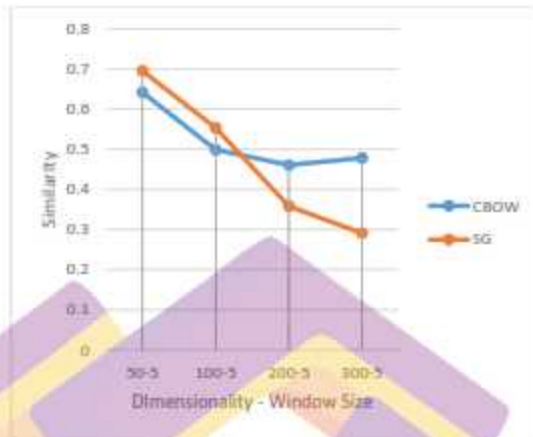
```
[.] word_model_50_cbow.wv.similarity('mafia', 'kartel')
```

```
0.6416664
```

```
[4] # compute cosine similarity
from numpy.linalg import norm
cosine = np.dot(x,y)/(norm(x)*norm(y))
print("Cosine Similarity:", cosine)
```

```
Cosine Similarity: 0.6416664171605831
```

Gambar 4.13. Hasil perhitungan cosine similarity kata 'mafia' dan 'kartel'



Gambar 4.14. Cosine Similarity berdasarkan arsitektur word2vec dan dimensi dari kata 'Mafia' dan 'kartel'

Gambar 4.14 memperlihatkan nilai kemiripan dari beberapa pasangan kata seperti kata 'susah' dan 'sulit', kata 'antri' dan 'antre' serta kata 'mafia' dan 'mapia'. Terlihat bahwa kelebihan dari word2vec adalah kemampuannya untuk menangkap makna yang sama terdapat dalam kata 'susah' dan 'sulit' serta kata yang 'typo' dimana hanya berbeda satu huruf. Pasangan kata 'mafia' dengan beberapa kata seperti 'bongkar', 'tangkap' dan 'oligarki' juga memiliki kemiripan yang cukup tinggi. Ternyata juga kata 'mafia' diasosiasikan dengan kata 'mendag' dalam penelitian ini yang memang gejala minyak goreng serta kelangkaan yang terjadi terkait dengan kebijakan yang dikeluarkan oleh Kementerian Perdagangan.


```
[ ] word_model_50_cbow.wv.similarity('susah', 'sulit')
0.6072103

word_model_50_cbow.wv.similarity('antri', 'antre')
0.5918728

[ ] word_model_50_cbow.wv.similarity('mafia', 'mapia')
0.4965331

[ ] word_model_50_cbow.wv.similarity('mafia', 'bongkar')
0.5725606

[ ] word_model_50_cbow.wv.similarity('mafia', 'tangkap')
0.63314044

[ ] word_model_50_cbow.wv.similarity('mafia', 'oligarki')
0.5170306

[ ] word_model_50_cbow.wv.similarity('mafia', 'mendang')
0.44317773
```

Gambar 4.15. Nilai Cosine Similarity dari beberapa pasangan kata

1.2 Perbandingan Kinerja Klasifikasi pada Dataset HET (Rancangan 1)

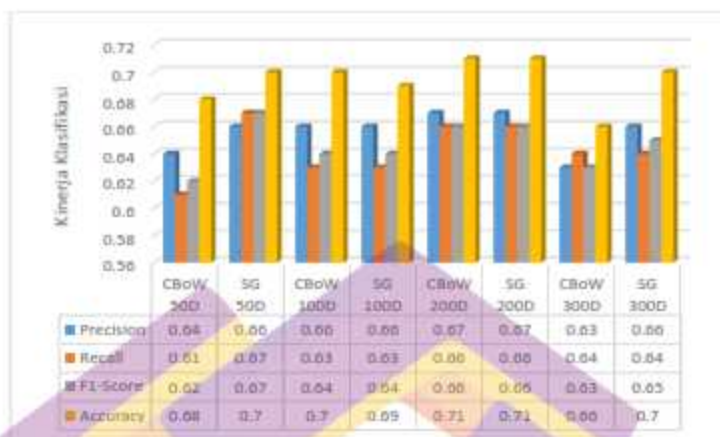
Perbandingan kinerja klasifikasi pada dataset HET dilakukan dengan beberapa metode ekstraksi fitur baik yang konvensional seperti BOW dan TF-IDF maupun yang cukup baru yaitu menggunakan word2vec serta metode klasifikasi BLSTM serta SVM. Kode python yang digunakan dalam model BLSTM dapat dilihat pada Lampiran 1 dan model SVM pada Lampiran 2. Struktur lapisan BLSTM dengan model SG 50D window size

5 yang terbentuk dapat dilihat pada Gambar 4.16. Sesuai dengan rata-rata jumlah kata dalam setiap dokumen yaitu 11 akan masuk ke dalam lapisan embedding bersama representasi vektor dari masing-masing kata tersebut yang berukuran 50. Kemudian vektor dihitung dengan menggunakan LSTM sebanyak dua layer. Lapisan LSTM pertama memiliki 10 node dan melakukan urutan kembali, diikuti oleh lapisan dropout 0.6. Kemudian, layer LSTM kedua memiliki 5 node tanpa urutan kembali, diikuti dengan dropout layer 0.4. Hasil akhir dari semua LSTM akan dimasukkan ke dalam lapisan padat dengan 5 node. Kemudian, sukseskan di lapisan output dengan 3 node dan gunakan fungsi aktivasi softmax untuk mengklasifikasikan sentimen negatif, netral, atau positif.

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 11, 50)	611950
bidirectional_2 (Bidirectional)	(None, 11, 20)	4880
dropout_2 (Dropout)	(None, 11, 20)	0
bidirectional_3 (Bidirectional)	(None, 10)	1040
dropout_3 (Dropout)	(None, 10)	0
dense_2 (Dense)	(None, 5)	55
dense_3 (Dense)	(None, 3)	16

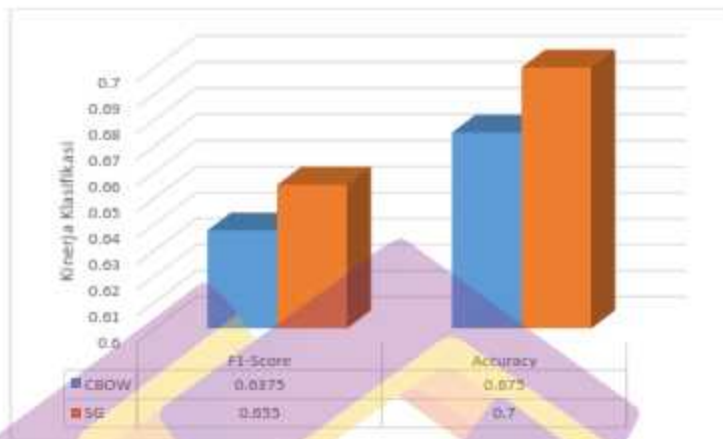
Total params: 617,943
 Trainable params: 617,943
 Non-trainable params: 0

Gambar 4.16. Struktur Lapisan BLSTM pada Data HET



Gambar 4.17. Kinerja Klasifikasi Dataset HET menggunakan BLSTM dengan Window Size 5

Gambar 4.17 memperlihatkan kinerja klasifikasi dari BLSTM pada berbagai arsitektur Word2Vec dan dimensi. Terlihat bahwa akurasi terbesar terjadi pada CBOW 200D atau SG 200D. Bila diperbandingkan untuk dua model Word2Vec dapat diketahui bahwa rata-rata akurasi dan F1-score pada model SG cenderung lebih tinggi dibandingkan dengan model CBOW (Gambar 4.18). Hal ini sejalan dengan penelitian (Khomsah, 2021) dan (Nawangsari et al., 2019). Sedangkan pengaruh ukuran dimensi terhadap kinerja klasifikasi dapat dilihat pada Gambar 4.19. Terlihat pola yang didapatkan tidak menunjukkan adanya pengaruh hubungan positif atau negative peningkatan kinerja sejalan dengan peningkatan dimensi vector. Hal ini berbeda dengan penelitian (Khomsah, 2021) yang menunjukkan bahwa peningkatan dimensi akan meningkatkan kinerja klasifikasi serta (Nawangsari et al., 2019) yang menunjukkan peningkatan dimensi akan menurunkan kinerja klasifikasi.



Gambar 4.18. Perbandingan Rata-rata Kinerja Klasifikasi BLSTM dengan Word2vec CBOW dan SG



Gambar 4.19. Perbandingan Rata-rata Kinerja Klasifikasi BLSTM dengan Word2vec Berdasarkan Dimensi Vektor

```

> t.test(df$F1.Score~df$Model)

Welch Two Sample t-test

data: df$F1.Score by df$Model
t = -1.6348, df = 5.5846, p-value = 0.1569
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.044172206  0.009172206
sample estimates:
mean in group CBOW   mean in group SG
           0.6375             0.6550

> t.test(df$Accuracy~df$Model)

Welch Two Sample t-test

data: df$Accuracy by df$Model
t = -1.8058, df = 1.5663, p-value = 0.1339
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.06535379  0.01535379
sample estimates:
mean in group CBOW   mean in group SG
           0.675             0.700

> anova_dim=aov(df$F1.Score~df$Dimension)
> summary(anova_dim)
          Df Sum Sq Mean Sq F value Pr(>F)
df$Dimension  1  0.0000019  0.0000019  0.006  0.942
Residuals    6  0.0019856  0.0003309

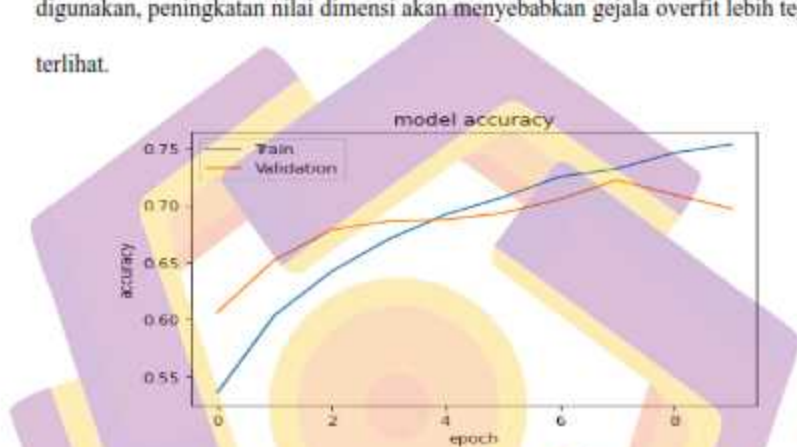
> anova_dim2=aov(df$Accuracy~df$Dimension)
> summary(anova_dim2)
          Df Sum Sq Mean Sq F value Pr(>F)
df$Dimension  1  0.000021  0.0000212  0.036  0.856
Residuals    6  0.003529  0.0005881

```

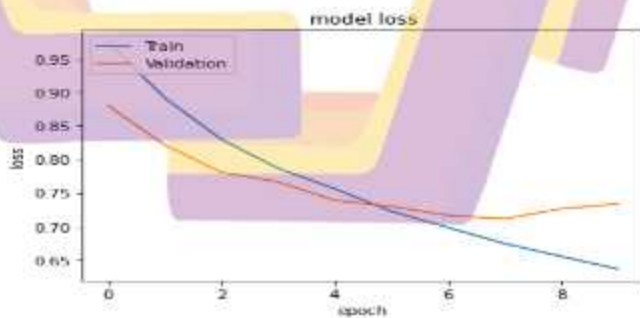
Gambar 4.20. Perbedaan Rata-rata Kinerja Klasifikasi BLSTM Berdasarkan Model CBOW/SG dan Dimensi Vektor

Hasil uji statistic perbedaan rata-rata kinerja menggunakan R-Studio dapat dilihat pada Gambar 4.20. Hasil uji t untuk uji perbedaan dua rata-rata untuk model CBOW dan SG menunjukkan hasil perbedaan rata-rata yang tidak signifikan yang ditandai oleh nilai p-value > 0.05. Begitu pula hasil uji perbedaan rata-rata lebih dari dua grup menggunakan ANOVA yang menunjukkan tidak adanya perbedaan yang signifikan rata-rata kinerja klasifikasi antar ukuran dimensi vector yang ditandai oleh nilai $Pr(>F) > 0.05$. Model SG 50D terlihat lebih rendah sedikit nilai akurasinya namun bila diperhatikan pada Gambar 4.21 dan 4.22 yang menampilkan

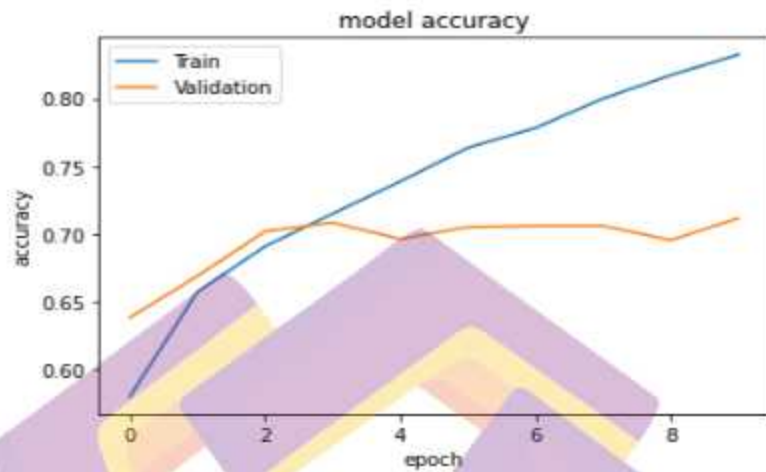
model akurasi dan loss, pola akurasi dan loss antara data latih dan data uji walaupun tetap terlihat adanya gejala overfit namun tidak terlalu signifikan. Hal ini jauh berbeda bila dilihat pada Gambar 4.23 dan 4.24 yang secara jelas menampilkan adanya gejala overfit. Gejala ini menunjukkan bahwa pada arsitektur yang digunakan, peningkatan nilai dimensi akan menyebabkan gejala overfit lebih tegas terlihat.



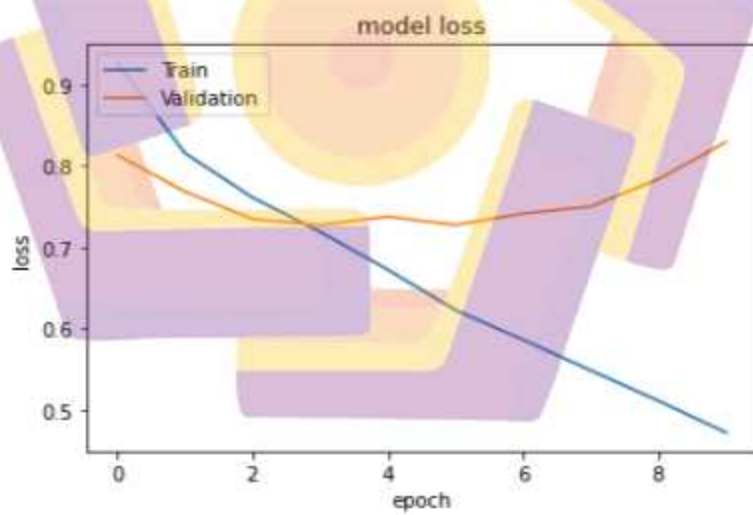
Gambar 4.21. Model Akurasi Berdasarkan BLSTM SG 50D



Gambar 4.22. Model Loss Berdasarkan BLSTM SG 50D



Gambar 4.23. Model Akurasi Berdasarkan BLSTM CBOW 200D



Gambar 4.24. Model Loss Berdasarkan BLSTM CBOW 200D



Gambar 4.25. Kinerja klasifikasi Dataset HET menggunakan SVM dengan Word2Vec Window Size 5, TF-IDF dan BOW

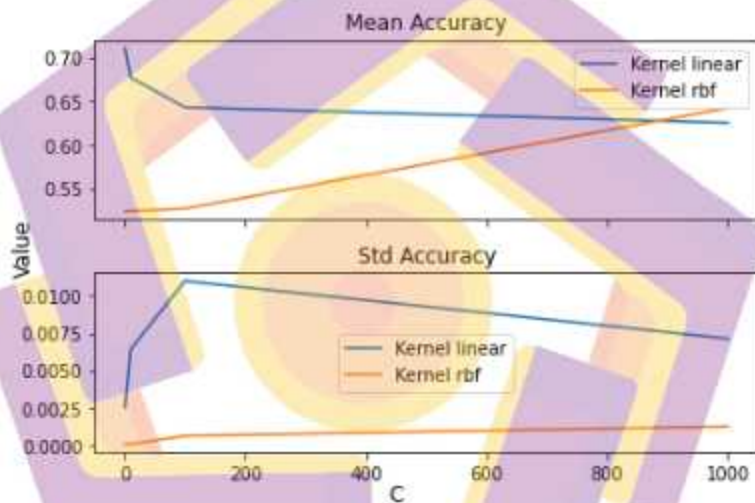
Gambar 4.25 memperlihatkan perbandingan kinerja klasifikasi dari SVM baik dengan menggunakan ekstraksi fitur Word2vec maupun dengan TF-IDF dan BOW. Terlihat bahwa model word2vec memiliki kinerja yang jauh lebih rendah dibandingkan dengan TF-IDF dan BOW. Hal ini diduga word2vec kurang bagus atau terjadi penurunan kinerja untuk *shallow learning*. Hal ini diduga word2vec memerlukan pelatihan lebih banyak untuk dapat menangkap makna kata secara lebih baik terlebih lagi bila dilanjutkan dengan mesin klasifikasi yang bersifat shallow learning. Hal ini juga sejalan dengan penelitian dari (Fauzi, 2019), (Cahyani & Patasik, 2021) dan (Oyeon Park et al., 2020) yang menyimpulkan model Word2vec memiliki nilai akurasi paling rendah dibandingkan metode lainnya seperti TF-IDF dan BOW pada dataset penelitiannya. Hal ini diduga karena dataset berukuran relatif kecil untuk melatih model Word2Vec. Word2Vec

membutuhkan contoh besar untuk mempelajari representasi kata dan menempatkan kata-kata serupa ke posisi yang lebih dekat.

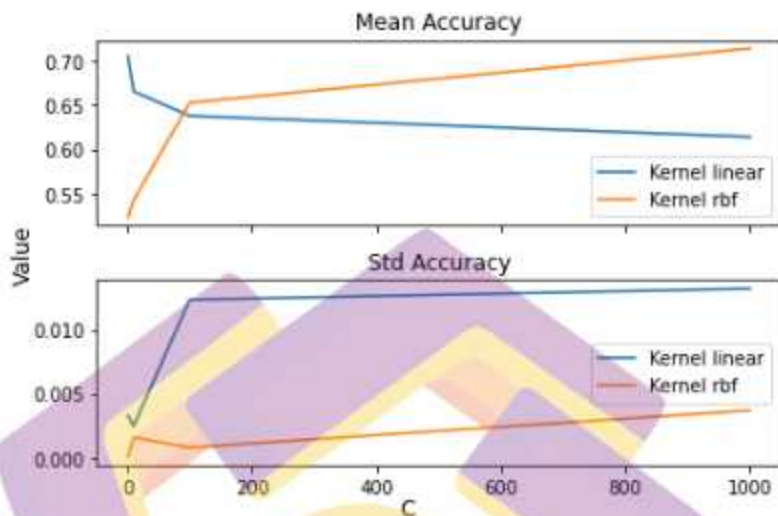
Penyetelan parameter SVM dilakukan pada jenis kernel serta nilai C (regularisasi) atau parameter penalty menggunakan *cross validation grid search* atau Grid Search CV. Nilai parameter grid yang diuji coba adalah pada rentang nilai C [1, 10, 100, 1000] dan kernel ['linear', 'rbf']. Pencarian grid dapat digambarkan sebagai eksplorasi lengkap yang menguji semua kombinasi hyperparameter yang diberikan pada konfigurasi grid dan dalam sebuah penelitian digunakan nilai C dari 6 sampai dengan 13 dan nilai kernel linier dan radial basis function (rbf) (Belete & Huchaiah, 2021). Penelitian lain menggunakan parameter "C" dengan nilai 0.001, 0.01, 0.1, 1, 10, dan 100 dan kernel "linear" (Michael, 2022) sedangkan (Toha et al., 2022) menggunakan tiga jenis kernel yaitu 'poly', 'linear' dan 'rbf'. Kernel mentransformasi ruang vector pada asli yang berdimensi rendah dan mengubahnya menjadi ruang berdimensi lebih tinggi supaya data dapat dipisahkan secara linier dengan mudah. Hal biasanya berkaitan dengan pemisahan non linier (Cristianini & Shawe-taylor, 2000). Parameter C digunakan untuk mengendalikan aspek untung-rugi (trade-off) antara batas keputusan dan kesalahan klasifikasi. Parameter C dapat digunakan untuk menentukan seberapa besar ingin menghindari salah pada data latih. Untuk nilai C yang besar, pengoptimalan akan memilih hyperplane dengan margin lebih kecil jika itu hyperplane melakukan pekerjaan yang lebih baik untuk mendapatkan semua poin pelatihan diklasifikasikan dengan benar (Swapna, 2022). Bila nilai C tinggi, algoritma akan

mengklasifikasikan semua titik data dengan benar namun bisa memungkinkan terjadinya overfit.

Penyetelan parameter SVM pada model TF-IDF menghasilkan nilai akurasi tertinggi pada nilai $C=1$ dan kernel linier (Gambar 4.26). Sedangkan pada model BOW nilai akurasi tertinggi didapatkan pada $C=100$ dan kernel rbf (Gambar 4.27).

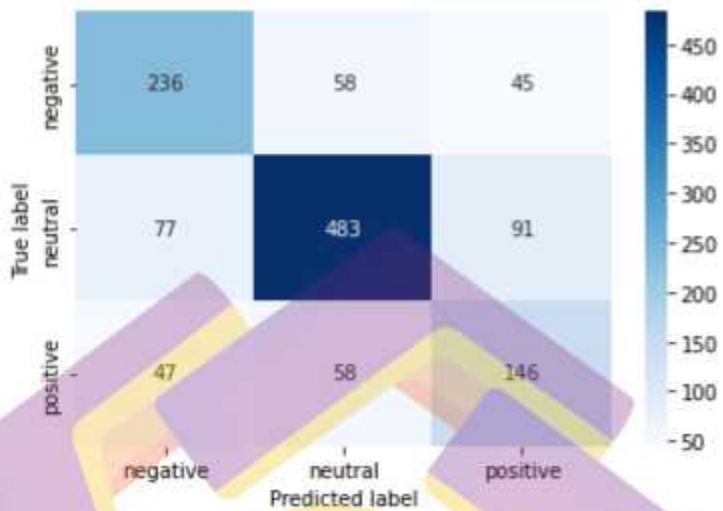


Gambar 4.26. Hasil Penyetelan Parameter C dan Kernel pada Klasifikasi SVM-TF-IDF

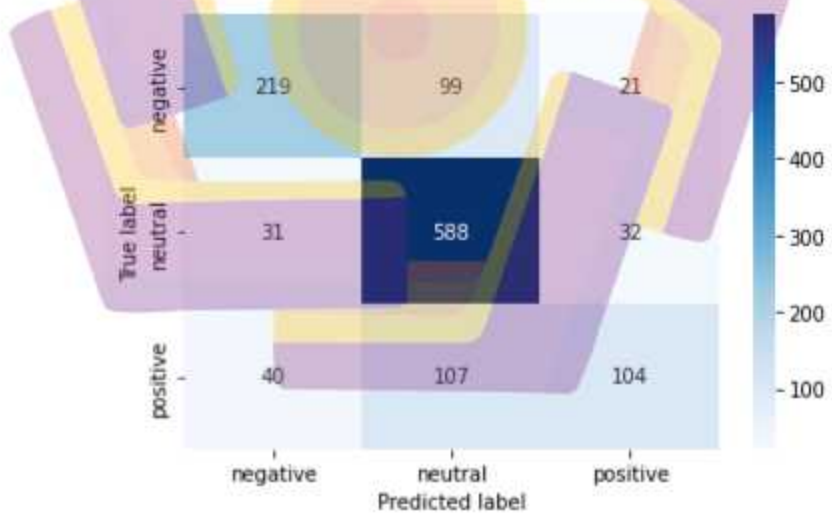


Gambar 4.27. Hasil Penyetelan Parameter C dan Kernel pada Nilai Akurasi Klasifikasi SVM-BOW

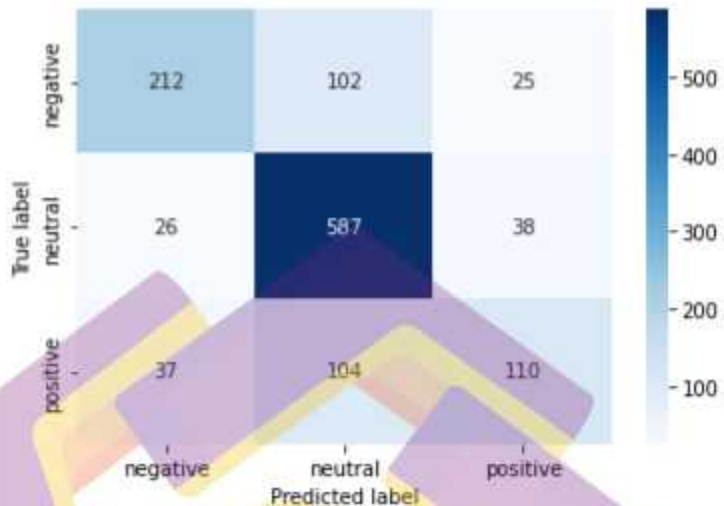
Gambar 4.28 sampai 4.30 menampilkan confusion matrix secara berturut-turut mulai dari klasifikasi BLSTM dengan model SG 50D, klasifikasi SVM dengan TF-IDF dan klasifikasi SVM dengan BOW. Nilai akurasi tertinggi didapatkan pada klasifikasi SVM dengan ekstraksi fitur BOW dengan nilai akurasi sebesar 0.73 yang sama dengan ekstraksi fitur TF-IDF. BOW sedikit lebih unggul dalam nilai recall dibandingkan TF-IDF. Nilai akurasi dari BLSTM masih lebih rendah yaitu sebesar 0.70. Namun demikian model BLSTM dengan SD 50D memiliki nilai F1-score yang sama dengan SVM BOW dan SVM TF-IDF yaitu sebesar 0.67.



Gambar 4.28. Confusion Matrix dari BLSTM dengan model SG 50D



Gambar 4.29. Confusion Matrix dari SVM dengan model TF-IDF



Gambar 4.30. Confusion Matrix dari SVM dengan model BOW

1.3 Ekstraksi Fitur Word2vec pada Dataset BLT

Hasil perhitungan rata-rata jumlah kata dalam setiap dokumen tweet sebesar 11,39 yang dibulatkan menjadi 11 yang didapatkan dari total kata sebesar 15207 kata dengan tweet yang sudah bersih sebanyak 1335 tweet. Ukuran kamus sebesar 1702 kata unik. Indeks untuk tiap kata dapat dilihat pada Tabel 4.4. Tweet pada setiap dokumen akan dikonversi menjadi nilai indeks dengan contoh seperti yang dapat dilihat pada Tabel 4.5. Terlihat bahwa karena jumlah kata kurang dari 11 maka diberikan padding value 0 di awal.

Tabel 4.4. Contoh Kamus Kata pada Dataset BLT

Index	Word
1	minyak
2	goreng

Tabel 4.4. Contoh Kamus Kata pada Dataset BLT (lanjutan)

Index	Word
3	blt
4	bantu
5	perintah
6	salur
7	masyarakat
8	terima
9	rp
10	presiden
11	jokowi
12	tunai
13	harga
14	jambi
15	rakyat

Tabel 4.5. Contoh Tweet dan Nilai Indeks pada Dataset BLT

Tweet	Index
blt minyak goreng bantu pangan non tunai cair	[0 0 0 3 1 2 4 53 127 12 22]
blt solusi blt olah olah perintah tunduk mafia minyak goreng ketawa penuh menang	[3 343 343 5 368 39 1 2 1164 213 209]
pkh blt minyak goreng cair april cek catat terima	[0 0 49 3 1 2 22 15 71 283 8]

Dua contoh vector yang didapatkan dari arsitektur atau model SG dengan dimensi 50 dan window size 5 dapat dilihat pada Tabel 4.6. Vektor berukuran 50 dan didapatkan dengan menggunakan model SG. Kata 'presiden' dan 'joko' yang terdapat dalam dokumen tweet dapat ditangkap oleh model SG dengan similaritas yang cukup tinggi yaitu sebesar 0.684 (Gambar 4.31).

Tabel 4.6. Contoh Vector kata dari kata 'presiden' dan 'joko'

Word	Word Vector SG 50D
presiden	array([-0.22173268, 0.36107484, -0.08529703, 0.1057525, -0.392858, 0.03457518, -0.546452, 0.54573405, -0.1932228, 1.2761835, -0.43019134, -0.16131677, -0.8147112, -0.00925156, -1.3289199, 0.6944458, 0.5573534, -0.16991602, 0.20886894, 0.67323965, 0.8073816, 0.04965236, -0.3990146, -0.25982538, 0.24226318, -0.3892284, 0.35927245, -0.67797405, 1.1518921, 0.04892143, -0.6247293, 0.1476876, 0.01023157, -0.7480781, -0.26503626, -0.11624528, 1.0477937, -0.8452864, -0.05298308, 1.0749058, 0.11871333, -0.2883212, -0.39787537, -0.41106627, 0.31537968, -0.70030713, 0.24489087, 0.25631073, 0.2716412, 0.4815451], dtype=float32)
joko	array([-0.27754948, 0.58492726, 0.14888279, -0.17205203, 0.36261854, -0.24013019, 0.19406804, -0.01833916, -0.12742971, 2.1871378, 0.17445543, 0.10854571, -0.83840984, -0.08076847, -1.2348242, 0.18427463, -0.00812461, -0.36875832, -0.53437966, 0.26182213, -0.01699635, 1.0243934, -0.2910422, 0.04876749, -0.30017376, -0.41287935, 0.48128134, 0.70924985, 0.8522213, -0.3040137, -0.3242471, 0.5393866, 0.61788714, -0.01364699, -0.64141786, 0.3057178, 0.8216766, -1.4027721, 0.6111982, 0.23286587, -0.5175387, -0.62267655, -0.77327365, 0.71586883, 0.45408905, -0.78942615, 0.2842239, 0.5764373, 0.4092003, 0.79542947], dtype=float32)

```
[ ] word_model.wv.most_similar('presiden', topn=3)
```

```
[('joko', 0.6841455698013306),
 ('gor', 0.6330770254135132),
 ('apri', 0.6255810260772705)]
```

```
[ ] # compute cosine similarity
from numpy.linalg import norm
cosine = np.dot(x,y)/(norm(x)*norm(y))
print("Cosine Similarity:", cosine)
```

```
Cosine Similarity: 0.6841454917517441
```

Gambar 4.31. Hasil perhitungan cosine similarity kata 'presiden' dan 'joko'

Gambar 4.32 memperlihatkan tiga kata yang paling mirip dengan kata yang ditanyakan. Dalam hal ini ditanyakan kata 'goreng' yang memiliki kemiripan tertinggi dengan kata 'blt', 'minyak' dan kata 'bantu'. Kata 'minyak' yang memiliki kemiripan dengan kata 'blt', 'goreng' dan kata 'bantu' serta kata 'blt' yang memiliki kemiripan tertinggi dengan kata 'minyak', 'goreng' dan 'perintah'. Kata 'blt' ini ternyata lebih tinggi kemiripannya dengan kata 'perintah' dibandingkan dengan kata 'bantu'.

```
[ ] word_model.wv.most_similar('goreng', topn=3)
[('blt', 0.8322471380233765),
 ('minyak', 0.791400671005249),
 ('bantu', 0.5526338815689087)]

[ ] word_model.wv.most_similar('minyak', topn=3)
[('blt', 0.8696589469909668),
 ('goreng', 0.791400671005249),
 ('bantu', 0.5349310981170654)]

[ ] word_model.wv.most_similar('blt', topn=3)
[('minyak', 0.8696589469909668),
 ('goreng', 0.8322471380233765),
 ('perintah', 0.5318762063900103)]
```

Gambar 4.32. Nilai Cosine Similarity dari beberapa pasangan kata pada dataset BLT

1.4 Perbandingan Kinerja Klasifikasi pada Dataset BLT (Rancangan

2)

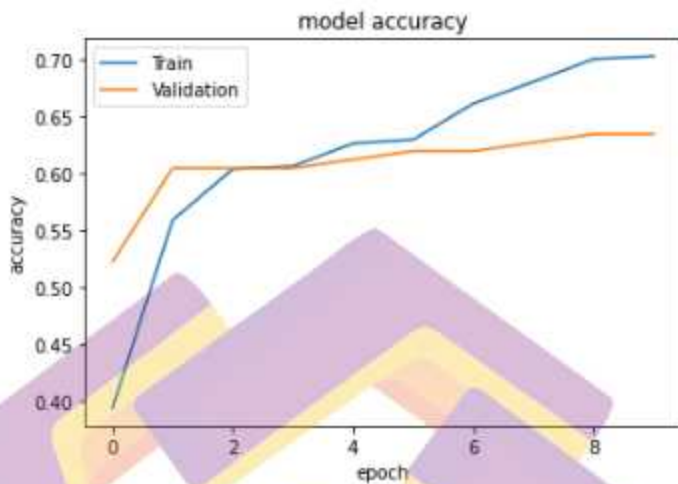
Ekstraksi fitur word2vec diuji baik pada algoritma klasifikasi deep learning (ANN, CNN, LSTM, BLSTM) maupun shallow learning (SVM, NB, KNN, LR). Kode python yang digunakan pada masing-masing arsitektur deep learning dapat

dilihat pada Lampiran 3 sampai 6. Layer embedding berisi vector dengan ukuran dimensi 50 yang mengkonversi setiap kata dalam dokumen tweet dimana vector tersebut didapatkan dari arsitektur word2vec dengan model skip-gram (SG). Fungsi aktivasi yang digunakan dalam layer tersebut ANN adalah relu yang merupakan singkatan dari *Rectified Linear Unit*, sedangkan untuk layer output menggunakan fungsi softmax. Regularisasi ditetapkan dengan menggunakan dropout sebesar 20% atau 0.2. Optimalisasi model menggunakan algoritma *adam* dengan ukuran metrik adalah akurasi.

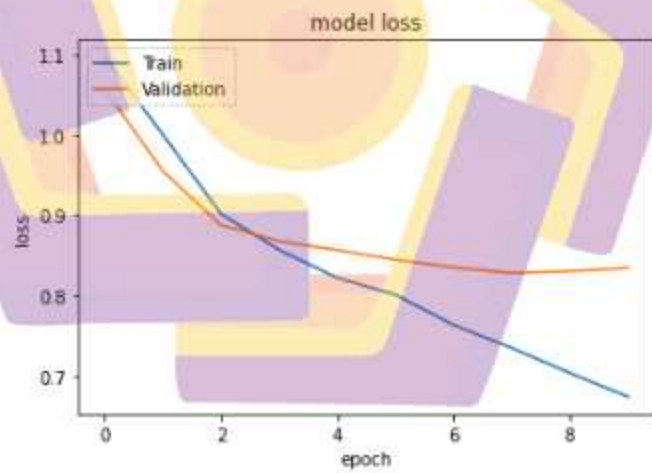
Layer (type)	Output Shape	Param #
embedding_16 (Embedding)	(None, 11, 50)	85150
dense_31 (Dense)	(None, 11, 10)	510
dropout_31 (Dropout)	(None, 11, 10)	0
dense_32 (Dense)	(None, 11, 5)	55
flatten_2 (Flatten)	(None, 55)	0
dense_33 (Dense)	(None, 3)	168

Total params:	85,883	
Trainable params:	85,883	
Non-trainable params:	0	

Gambar 4.33. Struktur Lapisan ANN pada Data BLT



Gambar 4.34. Model Accuracy dari ANN pada Data BLT



Gambar 4.35. Model loss dari ANN pada Data BLT

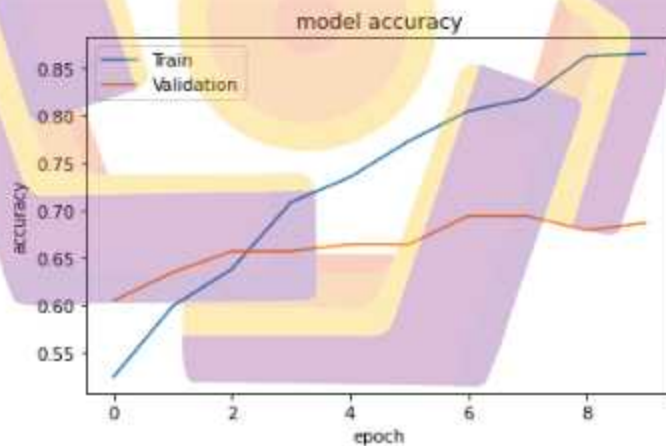
Gambar 4.34 dan 4.35 memperlihatkan hasil pengujian nilai akurasi dan *loss* dengan nilai epoch 10. Terlihat bahwa walaupun sudah ditambahkan *dropout* untuk mengatasi *overfitting* namun tetap menunjukkan adanya gejala *overfitting*. Kondisi ini dapat terjadi karena keterbatasan data training baik dari segi ukuran maupun jumlah noise (Ying, 2019).

Kode python untuk arsitektur CNN yang digunakan dapat dilihat pada Lampiran 4. Sebagai input adalah lapisan *embedding* dengan ukuran dimensi vektor 50. Lapisan *Conv1D* berfungsi menerima masukan dari lapisan *embedding* dengan ukuran kernel sebesar 5 dan fungsi aktivasi *relu*. *Dropout* sebesar 0.2 digunakan untuk mengatasi adanya kendala *over fitting*. Struktur lapisan CNN yang terbentuk dapat dilihat pada Gambar 4.36. Gambar 4.37 dan 4.38 memperlihatkan tren nilai akurasi dan *loss* baik pada data latih maupun data tes dan belum menunjukkan tren yang sama sehingga hal ini menunjukkan adanya pola *overfitting*.

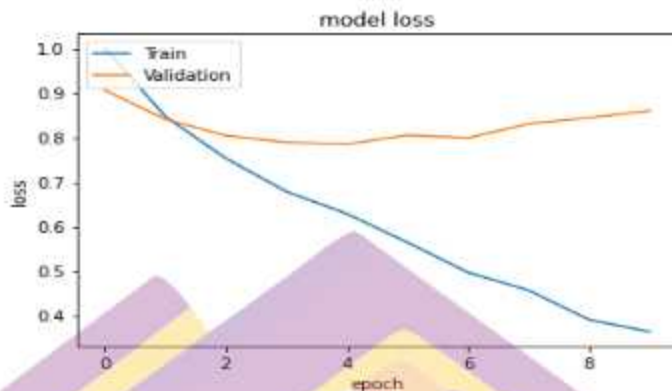
Layer (type)	Output Shape	Param #
embedding_17 (Embedding)	(None, 11, 50)	85150
conv1d_1 (Conv1D)	(None, 7, 32)	8032
max_pooling1d_1 (MaxPooling1D)	(None, 3, 32)	0
flatten_3 (Flatten)	(None, 96)	0
dense_34 (Dense)	(None, 10)	970
dropout_32 (Dropout)	(None, 10)	0
dense_35 (Dense)	(None, 3)	33

Total params: 94,185		
Trainable params: 94,185		
Non-trainable params: 0		

Gambar 4.36. Struktur Lapisan CNN pada Data BLT



Gambar 4.37. Model Accuracy dari CNN pada Data BLT



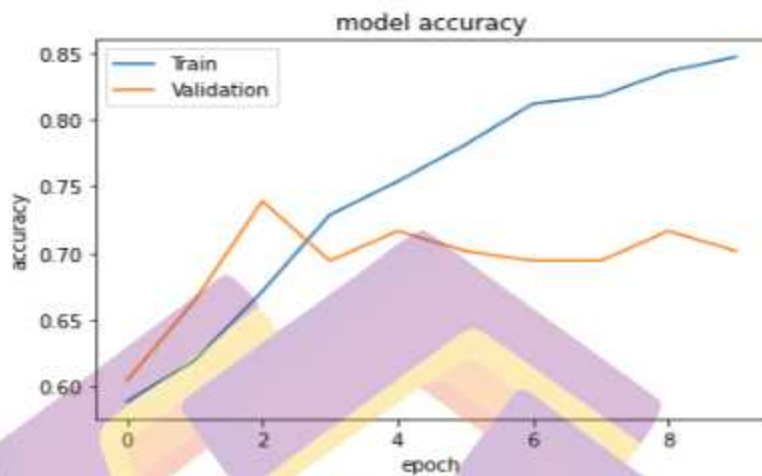
Gambar 4.38. Model Loss dari CNN pada Data BLT

Struktur lapisan LSTM yang digunakan pada data BLT dapat dilihat pada Gambar 4.39. Jumlah lapisan hidden layer yang digunakan sebanyak dua buah dan dilengkapi dengan proses regularisasi dengan menggunakan dropout. Klasifikasi LSTM yang digunakan masih menunjukkan adanya pola overfitting sebagaimana ditunjukkan pada Gambar 4.40 dan 4.41.

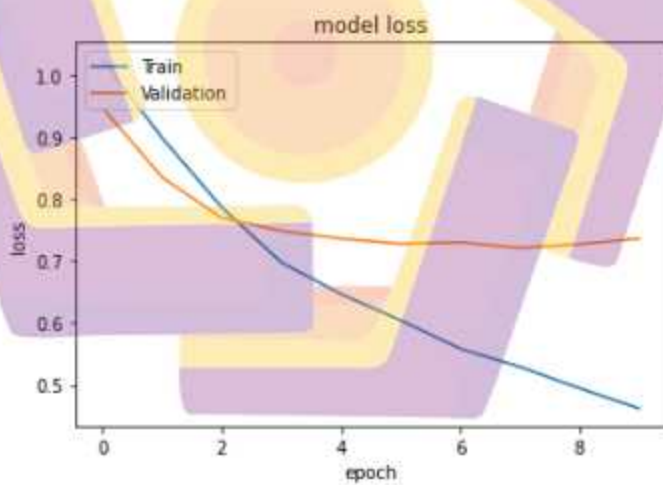
Layer (type)	Output Shape	Param #
embedding_18 (Embedding)	(None, 11, 50)	85150
lstm_26 (LSTM)	(None, 11, 10)	2440
lstm_27 (LSTM)	(None, 5)	320
dropout_33 (Dropout)	(None, 5)	0
dense_36 (Dense)	(None, 5)	30
dense_37 (Dense)	(None, 3)	18

Total params: 87,958
 Trainable params: 87,958
 Non-trainable params: 0

Gambar 4.39. Struktur Lapisan LSTM pada Data BLT



Gambar 4.40. Model Accuracy dari LSTM pada Data BLT



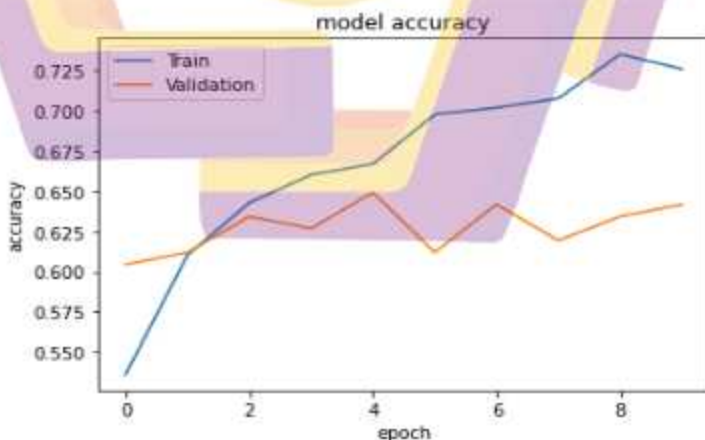
Gambar 4.41. Model loss dari LSTM pada Data BLT

Struktur lapisan yang terbentuk untuk klasifikasi menggunakan BLSTM dapat dilihat pada Gambar 4.42 dengan jumlah parameter sebanyak 91143. Terdapat dua layer tersembunyi yang digunakan dan dua lapisan dropout.

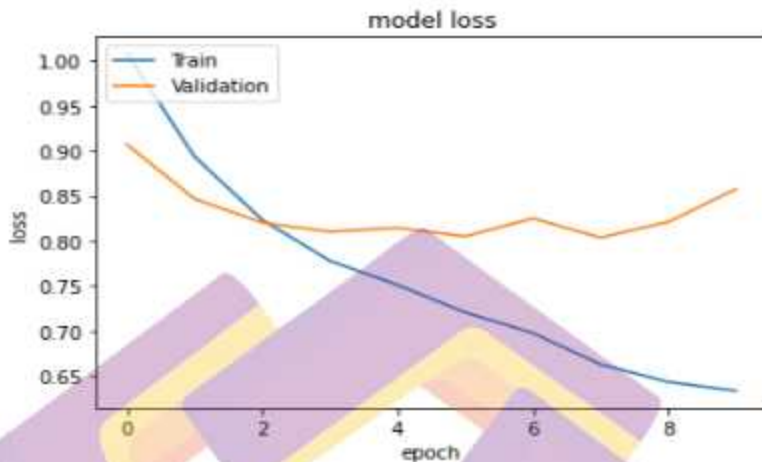
Layer (type)	Output Shape	Param #
embedding_25 (Embedding)	(None, 11, 50)	85150
bidirectional_36 (Bidirectional)	(None, 11, 20)	4880
dropout_46 (Dropout)	(None, 11, 20)	0
bidirectional_37 (Bidirectional)	(None, 10)	1040
dropout_47 (Dropout)	(None, 10)	0
dense_50 (Dense)	(None, 3)	55
dense_51 (Dense)	(None, 3)	18

Total params: 91,143
 Trainable params: 91,143
 Non-trainable params: 0

Gambar 4.42. Struktur Lapisan BLSTM pada Data BLT



Gambar 4.43. Model Accuracy dari BLSTM pada Data BLT



Gambar 4.44. Model loss dari BSTM pada Data BLT

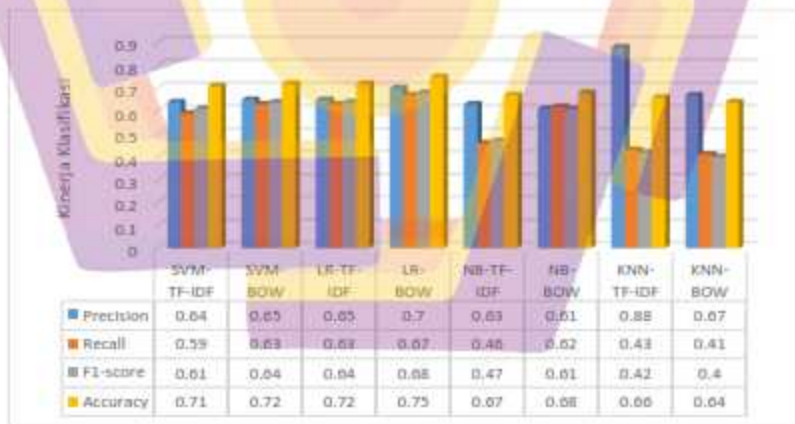
Gambar 4.43 dan 4.44 menunjukkan masih adanya pola overfitting pada klasifikasi BLSTM untuk data BLT. Lampiran 5 dan 6 memperlihatkan kode python untuk membangun arsitektur atau struktur lapisan LSTM dan BLSTM. Terlihat terdapat kesamaan yaitu sama-sama menggunakan dua lapisan namun perbedaannya dalam jumlah dropout serta algoritma optimalisasi yang digunakan.

Ekstraksi fitur word2vec digunakan baik untuk klasifikasi deep learning maupun shallow learning dengan hasil seperti dapat dilihat pada Gambar 4.45. Sedangkan TF-IDF dan BOW hanya diuji pada klasifikasi shallow learning. Terlihat bahwa LSTM memiliki kinerja terbaik untuk klasifikasi yang menggunakan word2vec dengan nilai akurasi sebesar 0.7 dan f1-score sebesar 0.62. Sebagaimana halnya pada dataset HET, algoritma shallow learning juga menunjukkan kinerja yang jauh lebih rendah bila menggunakan word2vec terlebih

lagi dataset BLT memiliki ukuran data yang jauh lebih kecil yaitu sebanyak 1335 tweet yang sudah bersih.

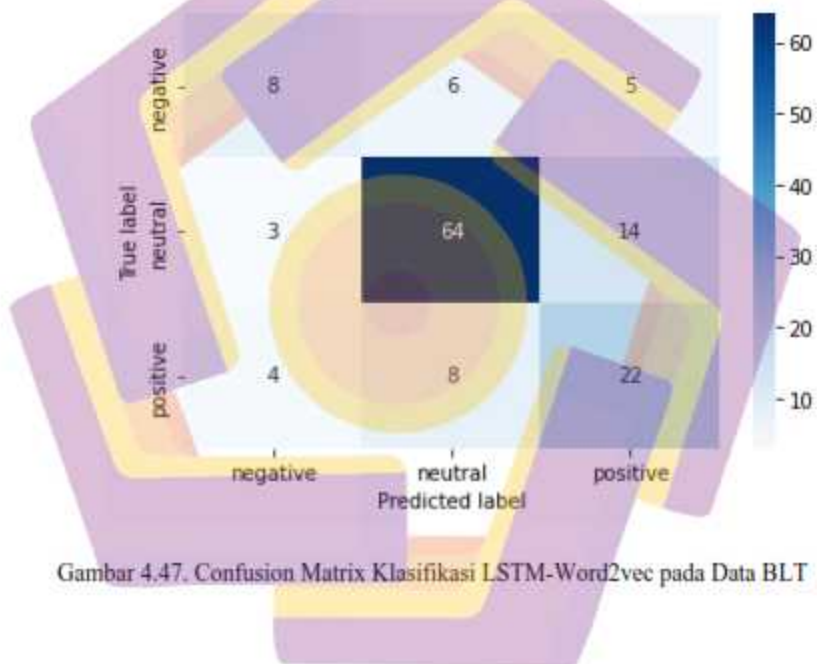


Gambar 4.45. Perbandingan kinerja Klasifikasi Menggunakan Ekstraksi Fitur Word2vec

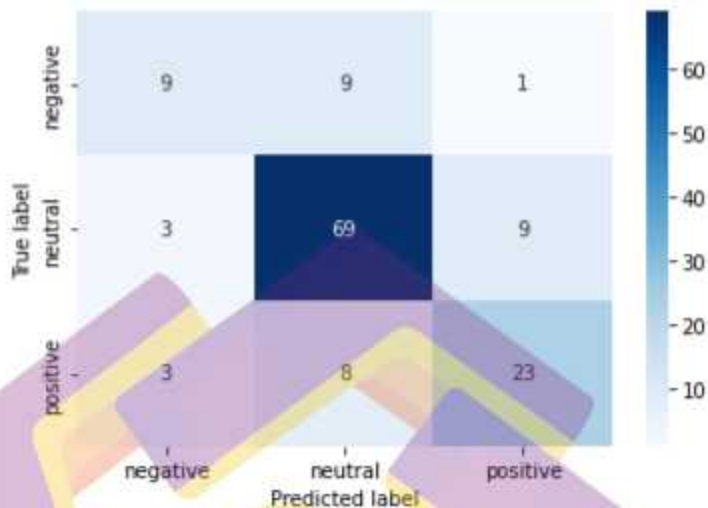


Gambar 4.46. Perbandingan Kinerja Klasifikasi Menggunakan Ekstraksi TF-IDF dan BOW

Gambar 4.46 memperlihatkan perbandingan kinerja klasifikasi shallow learning baik menggunakan TF-IDF maupun BOW. Nilai f1-score dan akurasi tertinggi didapatkan pada model LR-BOW dengan nilai berturut-turut sebesar 0.68 dan 0.75. Secara keseluruhan dari ketiga metode ekstraksi fitur yang digunakan pada dataset BLT ini dapat diketahui bahwa kombinasi LR dengan BOW memberikan kinerja terbaik disusul oleh LR – TF-IDF dan SVM-BOW.



Gambar 4.47. Confusion Matrix Klasifikasi LSTM-Word2vec pada Data BLT



Gambar 4.48. Confusion Matrix Klasifikasi LR-BOW pada Data BLT

Gambar 4.47 memperlihatkan hasil confusion matrix dari klasifikasi LSTM-Word2vec. Terlihat bahwa sepanjang diagonal utama telah menunjukkan warna yang lebih gelap dibandingkan sel lainnya yang menunjukkan true positive, true neutral dan true negative menunjukkan nilai yang lebih besar yang mempengaruhi pada perhitungan nilai akurasi. Persentase data uji yang digunakan adalah sebanyak 10%. Gambar 4.48 memperlihatkan hasil confusion matrix dari klasifikasi LR namun menggunakan ekstraksi fitur BOW yang menunjukkan hasil f1-score dan akurasi tertinggi dibandingkan klasifikasi lainnya. Nilai kinerja dari LR dan SVM yang lebih baik dibandingkan shallow learning lainnya sejalan dengan hasil penelitian dari (Sabri et al., 2021).

BAB V PENUTUP

5.1. Kesimpulan

Kesimpulan dari penelitian ini yaitu :

1. Rata-rata nilai akurasi dan F1-score pada model SG cenderung lebih tinggi dibandingkan dengan model CBOW pada data HET yang memiliki ukuran data lebih dari 10000 tweet tepatnya 12408. Nilai rata-rata akurasi pada model CBOW sebesar 0.675 dan SG 0,700 sedangkan rata-rata F1-score pada CBOW sebesar 0,6375 dan SG 0,6550. Namun berdasarkan uji-t didapatkan nilai p-value secara berturut-turut untuk akurasi sebesar 0.1539 dan F1-score sebesar 0.1569 dimana masih lebih besar dari taraf signifikan 0,05 sehingga disimpulkan tidak ada perbedaan rata-rata kinerja klasifikasi yang signifikan.
2. Pengaruh dari dimensi vector Word2Vec ukuran 50D, 100D, 200D dan 300D yang diuji menggunakan uji ANOVA menunjukkan hasil yang tidak signifikan baik pada kinerja akurasi maupun F1-Score dengan nilai p value lebih dari 0.05.
3. Berdasarkan pemeriksaan model akurasi dan loss dapat diketahui bahwa model BLSTM dengan SG 50D menunjukkan pola data latih dan data tes yang lebih mirip serta nilai akurasi 0.7 dan F1-Score 0.67. Hasil pemeriksaan pada dimensi yang lebih tinggi menunjukkan adanya gejala *overfitting*.

4. Model Word2Vec yang diterapkan pada klasifikasi SVM menunjukkan kegagalan dalam memprediksi kelas positif dan negative. Nilai F1-score cenderung rendah pada kisaran nilai 0.23 dan 0.25. Kinerja Word2Vec yang kurang bagus atau terjadi penurunan kinerja untuk *shallow learning* diduga membutuhkan data yang besar agar hasilnya bisa menangkap makna kata secara lebih baik.
5. Penyetelan hyperparameter pada *shallow learning* (SVM) dengan menggunakan TF-IDF pada data HET menunjukkan nilai akurasi tertinggi pada nilai $C=1$ dan kernel linier yaitu akurasi 0,73 dan F1-Score 0.67. Sedangkan pada model BOW nilai akurasi tertinggi didapatkan pada $C=100$ dan kernel rbf dengan nilai akurasi 0.73 dan F1-Score 0.67. Hal ini lebih tinggi kinerjanya dibandingkan model BLSTM dengan ekstraksi fitur Word2Vec.
6. Dalam kasus dataset BLT yang memiliki ukuran data yang jauh lebih kecil yaitu sebanyak 1335 tweet yang sudah bersih, diketahui bahwa LSTM dengan ekstraksi fitur Word2Vec memiliki kinerja terbaik untuk klasifikasi yang menggunakan word2vec dengan nilai akurasi sebesar 0.7 dan f1-score sebesar 0.62 dibandingkan metode *deep learning* yang lain (ANN, CNN dan BLSTM). Namun demikian sebagaimana halnya pada dataset HET, algoritma *shallow learning* juga menunjukkan kinerja yang jauh lebih rendah bila menggunakan word2vec.

7. Nilai *f1-score* dan akurasi tertinggi dengan menggunakan data BLT didapatkan pada model LR-BOW dengan nilai berturut-turut sebesar 0.68 dan 0.75 disusul oleh LR – TF-IDF dan SVM-BOW.
8. Model Word2Vec telah berhasil menangkap makna kata yang terkandung dalam data HET seperti kata 'mafia' memiliki nilai *cosine similarity* yang cukup tinggi dengan kata 'kartel', 'bongkar', 'tangkap', 'oligarki' dan 'mendag'. 'mahal', 'langka' dan kata 'mafia' yang memiliki kemiripan cukup tinggi dengan kata 'tangkap', 'bongkar', dan 'mendag' dimana hal ini bisa menjadi perhatian pemerintah untuk menangani harga minyak goreng. Nilai kemiripan dari pasangan kata akan semakin menurun dengan semakin besarnya nilai dimensi vektor.

5.2. Saran

Saran dari penelitian ini untuk menjadi lebih baik lagi yaitu :

1. Diperlukan penelitian lanjutan dengan melakukan perbaikan pada tahap pemrosesan data terutama melakukan pembersihan terhadap *noise* seperti data yang tidak memiliki makna atau tidak penting
2. Diperlukan pula penelitian lanjutan yang melakukan eksperimen berdasarkan pembagian data latih dan data uji yang lebih seimbang
3. Diperlukan penelitian lanjutan untuk menggunakan *random search* dalam *hyperparameter tuning*

DAFTAR PUSTAKA

- Agustiningasih, K. K., Utami, E., Muhammad, O., & Alsyabani, A. (2022). Sentiment Analysis of COVID-19 Vaccines in Indonesia on Twitter Using Pre-Trained and Self-Training Word Embeddings. *Jurnal Ilmu Komputer Dan Informasi*, 15(1), 39–46. <https://jiki.cs.ui.ac.id/index.php/jiki/article/view/1044>
- 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>
- Andoh, J., Asiedu, L., Lotsi, A., & Chapman-Wardy, C. (2021). Statistical Analysis of Public Sentiment on the Ghanaian Government: A Machine Learning Approach. *Advances in Human-Computer Interaction*, 2021. <https://doi.org/10.1155/2021/5561204>
- Belete, D. M., & Huchaiah, M. D. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of HIV / AIDS test results Grid search in hyperparameter optimization of machine learning models for prediction of HIV / AIDS test results. September. <https://doi.org/10.1080/1206212X.2021.1974663>
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). *Enriching Word Vectors with Subword Information*.
- Brownlee, J. (2017). Deep Learning for Natural Language Processing : Develop Deep Learning Models for Natural Language in Python. *Machine Learning Mastery*, 414. <http://web.stanford.edu/class/cs224n/readings/cs224n-2019->

notes06-NMT_seq2seq_attention.pdf

- Cahyani, D. E., & Patasik, I. (2021). Performance comparison of tf-idf and word2vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics*, 10(5), 2780–2788. <https://doi.org/10.11591/eei.v10i5.3157>
- Cristianini, N., & Shawe-taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*.
- Farhadloo, M., & Rolland, E. (2016). *Fundamentals of Sentiment Analysis and Its Applications*. August 2018, <https://doi.org/10.1007/978-3-319-30319-2>
- Fauzi, M. A. (2019). Word2Vec model for sentiment analysis of product reviews in Indonesian language. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(1), 525. <https://doi.org/10.11591/ijece.v9i1.pp525-530>
- Goldberg, Y. (2017). *Yoav Goldberg Book - Neural Network Methods in Natural Language Processing*.
- Goodfellow, I., Bengio, Yoshua, & Courville, Aaron. (2012). Deep Learning Ian. *Foreign Affairs*, 91(5), 1689–1699.
- Jang, B., Kim, M., Harerimana, G., Kang, S., & Kim, J. W. (2020). *applied sciences Bi-LSTM Model to Increase Accuracy in Text Classification : Combining Word2vec CNN and Attention Mechanism*.
- Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2vec model analysis for semantic similarities in English words. *Procedia Computer Science*, 157, 160–167. <https://doi.org/10.1016/j.procs.2019.08.153>

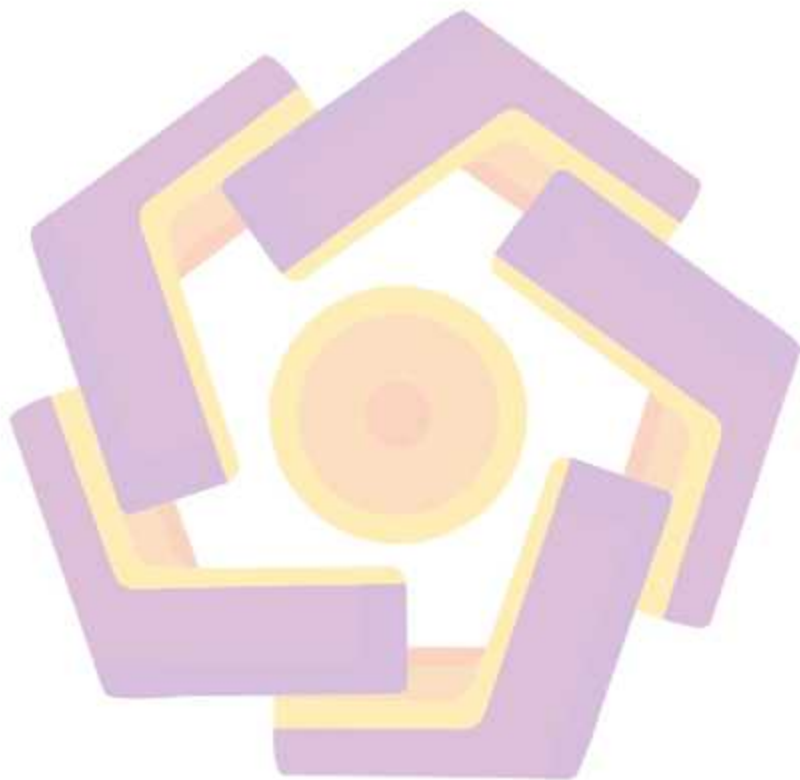
- Khomsah, S. (2021). Sentiment Analysis On YouTube Comments Using Word2Vec and Random Forest. *Telematika*, 18(1), 61. <https://doi.org/10.31315/telematika.v18i1.4493>
- Kubat, M. (2017). An Introduction to Machine Learning. In *Springer*. <https://doi.org/10.1007/978-3-319-63913-0>
- Ligthart, A., Catal, C., & Tekinerdogan, B. (2021). Systematic reviews in sentiment analysis : a tertiary study. In *Artificial Intelligence Review* (Vol. 54, Issue 7). Springer Netherlands. <https://doi.org/10.1007/s10462-021-09973-3>
- Liu, Z., Lin, Y., & Sun, M. (2020). Representation Learning for Natural Language Processing. In *Representation Learning for Natural Language Processing*. <https://doi.org/10.1007/978-981-15-5573-2>
- Michael, A. (2022). *Komparasi Kombinasi Pre-trained Model dengan SVM pada Klasifikasi Kematangan Kopi Berbasis Citra*. 7(1), 42–48.
- Nandini, R. A., Sari, Y. A., & Adikara, P. P. (2019). Analisis Sentimen Impor Beras 2018 Pada Twitter Menggunakan Metode Support Vector Machine dan Pembobotan Jumlah Retweet. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(4), 3396–3406.
- Nawang Sari, R. P., Kusumaningrum, R., & Wibowo, A. (2019). Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study. *Procedia Computer Science*, 157, 360–366. <https://doi.org/10.1016/j.procs.2019.08.178>
- Novák, J., Benda, P., Šilerová, E., Vaněk, J., & Kánská, E. (2021). Sentiment Analysis in Agriculture. *Agris On-Line Papers in Economics and Informatics*,

13(1), 121–130. <https://doi.org/10.7160/aol.2021.130109>

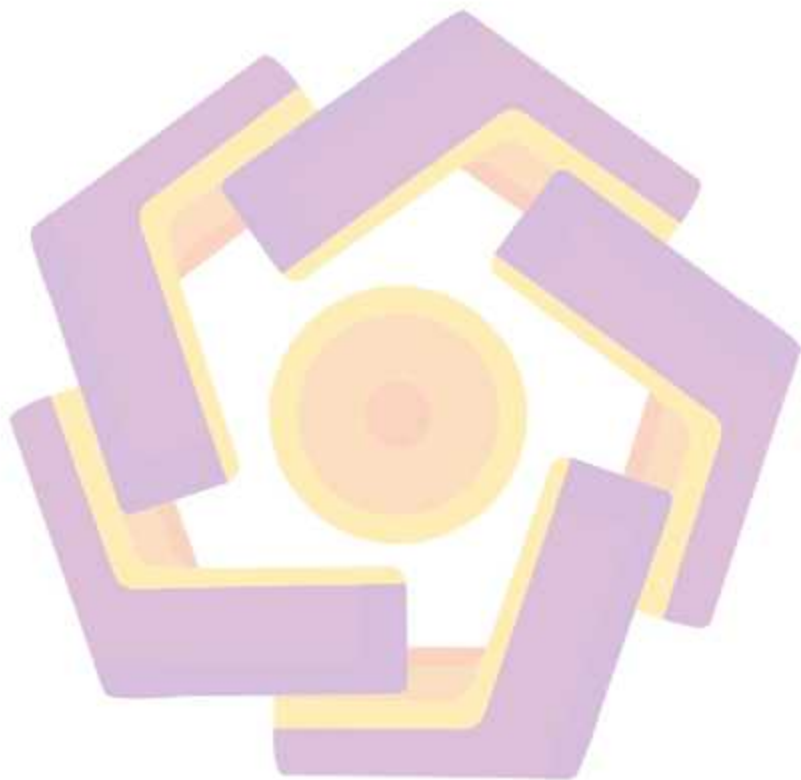
- Oyeon Park,) H. Kim, K.-J., Candidate, P. D., First, •, & Park, H. (2020). Impact of Word Embedding Methods on Performance of Sentiment Analysis with Machine Learning Techniques. *한국컴퓨터정보학회논문지 Journal of The Korea Society of Computer and Information*, 25(8), 181–188.
- Pasaribu, D. J. M., Kusriani, K., & Sudarmawan, S. (2020). Peningkatan Akurasi Klasifikasi Sentimen Ulasan Makanan Amazon dengan Bidirectional LSTM dan Bert Embedding. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, 10(1), 9–20. <https://doi.org/10.35585/inspir.v10i1.2568>
- Purwarianti, A., & Crisdayanti, I. A. P. A. (2019). Improving Bi-LSTM Performance for Indonesian Sentiment Analysis Using Paragraph Vector. *Proceedings - 2019 International Conference on Advanced Informatics: Concepts, Theory, and Applications, ICAICTA 2019*. <https://doi.org/10.1109/ICAICTA.2019.8904199>
- Rustam, F., Khalid, M., Aslam, W., Rupapara, V., Mehmood, A., & Choi, G. S. (2021). A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. *PLoS ONE*, 16(2), 1–23. <https://doi.org/10.1371/journal.pone.0245909>
- Sabri, T., Beggar, O. El, & Kissi, M. (2021). Comparative study of Arabic text classification using feature vectorization methods. *Procedia Computer Science*, 198(2021), 269–275. <https://doi.org/10.1016/j.procs.2021.12.239>
- Shahi, T. B., Sitaula, C., & Paudel, N. (2022). A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification. *Computational*

- Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/5681574>
- Skansi, S. (2020). Introduction to deep learning - From Logical Calculus to Artificial Intelligence. In *Springer*. Springer. https://doi.org/10.1007/978-981-15-4095-0_1
- Sulastri, K. (2020). *Klasifikasi Naïve Bayes pada Analisis Sentimen atas Penolakan Dibukanya Larangan Ekspor Benih Lobster*. 1(2), 68–75.
- Surjandari, I., Naffisnh, M. S., & Prawiradinata, M. I. (2014). Text Mining of Twitter Data for Public Sentiment Analysis of Staple Foods Price Changes. *Journal of Industrial and Intelligent Information*, 3(3), 253–257. <https://doi.org/10.12720/jiii.3.3.253-257>
- Swapna, A. (2022). *Autism Diagnosis Using Machine Learning and Neural Networks*. 11(6). <https://doi.org/10.15680/IJRSET.2022.1106092>
- Toha, A., Purwono, P., Gata, W., & Toha, A. (2022). *Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV*. 4(1), 12–21. <https://doi.org/10.12928/biste.v4i1.6079>
- Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. In *Artificial Intelligence Review* (Issue 0123456789). Springer Netherlands. <https://doi.org/10.1007/s10462-022-10144-1>
- Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 1168(2). <https://doi.org/10.1088/1742-6596/1168/2/022022>
- Zhang, F. (2022). A hybrid structured deep neural network with Word2Vec for

construction accident causes classification. *International Journal of Construction Management*, 22(6), 1120–1140.
<https://doi.org/10.1080/15623599.2019.1683692>



LAMPIRAN



Lampiran 1. Potongan Kode Python yang Digunakan dalam BLSTM pada Dataset HET

```
blstm_word2vec = Sequential()
blstm_word2vec.add(Embedding(nb_words, 50, input_length=max_seq_len, weights=[embedding_matrix_2_50_cbow]))
blstm_word2vec.add(Bidirectional(LSTM(10, return_sequences=True)))
blstm_word2vec.add(Dropout(0.6))
blstm_word2vec.add(Bidirectional(LSTM(5)))
blstm_word2vec.add(Dropout(0.4))
blstm_word2vec.add(Dense(3, activation="softmax"))
blstm_word2vec.summary()
blstm_word2vec.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['accuracy'])

history=blstm_word2vec.fit(word_seq_train, y_seq_train, batch_size=30, epochs=10, validation_data=(word_seq_test, y_seq_test))
```

Lampiran 2. Potongan Kode Python yang Digunakan dalam SVM

```
param_grid = {  
    'C': [1, 10, 100, 1000],  
    'kernel': ['linear', 'rbf']  
}  
  
#TF-IDF  
SVM = svm.SVC(gamma="auto")  
clf = GridSearchCV(SVM, param_grid, cv=3)  
clf.fit(tfidf_train,y_train)  
y_pred = clf.best_estimator_.predict(tfidf_test)  
  
#BOW  
SVM = svm.SVC(gamma="auto")  
clf = GridSearchCV(SVM, param_grid, cv=3)  
clf.fit(countword_train,y_train)  
y_pred = clf.best_estimator_.predict(countword_test)  
  
# Word2vec CBOW 50D  
SVM = svm.SVC(gamma="auto")  
clf = GridSearchCV(SVM, param_grid, cv=3)  
clf.fit(X_train_word2vec_scaled_50_cbow,y_train)  
y_pred = clf.best_estimator_.predict(X_test_word2vec_scaled_50_cbow)
```

Lampran 3. Potongan Kode Python yang Digunakan dalam ANN pada Data BLT

```
ann_word2vec = Sequential()
ann_word2vec.add(Embedding(nb_words_2, 50, input_length=max_seq_len, wei
ghts=[embedding_matrix_2])) #embedding_matrix_2 = word2vec
ann_word2vec.add(Dense(10, activation='relu'))
ann_word2vec.add(Dropout(0.2))
ann_word2vec.add(Dense(5, activation='relu'))
ann_word2vec.add(Flatten())
ann_word2vec.add(Dense(3, activation='softmax'))
ann_word2vec.compile(loss='categorical_crossentropy', optimizer='adam', metrics
=['accuracy'])
ann_word2vec.summary()
history=ann_word2vec.fit(word_seq_train, y_seq_train, epochs=10, validation_da
ta=(word_seq_test, y_seq_test))
```


Lampiran 4. Potongan Kode Python untuk Klasifikasi CNN pada Data BLT

```
cnn_word2vec = Sequential()
cnn_word2vec.add(Embedding(nb_words_2, 50, input_length=max_seq_len, weights=[embedding_matrix_2])) #embedding_matrix_2 = word2vec
cnn_word2vec.add(Conv1D(32, kernel_size=5, activation='relu'))
cnn_word2vec.add(MaxPooling1D(pool_size=2))
cnn_word2vec.add(Flatten())
cnn_word2vec.add(Dense(10, activation='relu'))
cnn_word2vec.add(Dropout(0.2))
cnn_word2vec.add(Dense(3, activation='softmax'))
cnn_word2vec.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
cnn_word2vec.summary()

history=cnn_word2vec.fit(word_seq_train, y_seq_train, epochs=10, validation_data=(word_seq_test, y_seq_test))
```

Lampiran 5. Potongan Kode Python untuk Klasifikasi LSTM pada Data BLT

```
lstm_word2vec = Sequential()
lstm_word2vec.add(Embedding(nb_words_2, 50, input_length=max_seq_len, weights=[embedding_matrix_2])) #embedding_matrix_2 = word2vec
lstm_word2vec.add(LSTM(10, return_sequences=True))
lstm_word2vec.add(LSTM(5))
lstm_word2vec.add(Dropout(0.2))
lstm_word2vec.add(Dense(5))
lstm_word2vec.add(Dense(3, activation="softmax"))
lstm_word2vec.summary()
lstm_word2vec.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

history=lstm_word2vec.fit(word_seq_train, y_seq_train, epochs=10, validation_data=(word_seq_test, y_seq_test))
```

Lampiran 6. Potongan Kode Python Klasifikasi BLSTM pada Data BLT

```
blstm_word2vec = Sequential()

blstm_word2vec.add(Embedding(nb_words_2, 50, input_length=max_seq_len, w
eights=[embedding_matrix_2])) #embedding_matrix_2 = word2vec

blstm_word2vec.add(Bidirectional(LSTM(10, return_sequences=True)))

blstm_word2vec.add(Dropout(0.6))

blstm_word2vec.add(Bidirectional(LSTM(5)))

blstm_word2vec.add(Dropout(0.4))

blstm_word2vec.add(Dense(5))

blstm_word2vec.add(Dense(3, activation="softmax"))

blstm_word2vec.summary()

blstm_word2vec.compile(loss='categorical_crossentropy', optimizer='rmsprop', m
etrics=['accuracy'])

history=blstm_word2vec.fit(word_seq_train, y_seq_train, epochs=10, validation_
data=(word_seq_test, y_seq_test))
```

Lampiran 7. Alamat Web Publikasi Ilmiah Terkait Tests dan Kuliah

<https://www.trijurnal.trisakti.ac.id/index.php/tekin/article/view/15669>

<http://jurnal.ubl.ac.id/index.php/explore/article/view/2348>

<http://jurnal.ubl.ac.id/index.php/explore/article/view/2197>

<https://ieeexplore.ieee.org/document/9563977>

