TESIS

ANALISIS PERBANDINGAN METODE BAG OF WORDS, TF-IDF, WORD2VEC DAN DOC2VEC PADA KLASIFIKASI TEKS SENTIMEN MASYARAKAT TERHADAP PRODUK LOKAL DI INDONESIA

Disusun oleh:

Nama : **Ivan Rifky Hendrawan**
NIM : **21.51.1002**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023

TESIS

ANALISIS PERBANDINGAN METODE BAG OF WORDS, TF-IDF, WORD2VEC DAN DOC2VEC PADA KLASIFIKASI TEKS SENTIMEN MASYARAKAT TERHADAP PRODUK LOKAL DI INDONESIA

COMPARATIVE ANALYSIS OF BAG OF WORDS, TF-IDF,WORD2VEC AND DOC2VEC METHODS ON CLASSIFICATION OF COMMUNITY SENTIMENT TEXTS TOWARDS LOCAL PRODUCTS IN INDONESIA

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Ivan Rifky Hendrawan**
NIM	**: 21.51.1002**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

HALAMAN PENGESAHAN

ANALISIS PERBANDINGAN METODE BAG OF WORDS, TF-IDF, WORD2VEC DAN DOC2VEC PADA KLASIFIKASI TEKS SENTIMEN MASYARAKAT TERHADAP PRODUK LOKAL DI INDONESIA

COMPARATIVE ANALYSIS OF BAG OF WORDS, TF-IDF,WORD2VEC AND DOC2VEC METHODS ON CLASSIFICATION OF COMMUNITY SENTIMENT TEXTS TOWARDS LOCAL PRODUCTS IN INDONESIA

Dipersiapkan dan Disusun oleh

Ivan Rifky Hendrawan
21.51.1002

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

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

Yogyakarta, 6 Desember 2022
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISIS PERBANDINGAN METODE BAG OF WORDS, TF-IDF, WORD2VEC DAN DOC2VEC PADA KLASIFIKASI TEKS SENTIMEN MASYARAKAT TERHADAP PRODUK LOKAL DI INDONESIA

COMPARATIVE ANALYSIS OF BAG OF WORDS, TF-IDF,WORD2VEC AND DOC2VEC METHODS ON CLASSIFICATION OF COMMUNITY SENTIMENT TEXTS TOWARDS LOCAL PRODUCTS IN INDONESIA

Dipersiapkan dan Disusun oleh

Iyan Rifky Hendrawan
21.51.1002

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

Pembimbing Utama

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

Anggota Tim Penguji

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

Pembimbing Pendamping

Alva Hendi Muhammad,S.T., M.Eng.,Ph.D.
NIK. 190302493

Anggit Dwi Hartanto, M.Kom
NIK. 190302163

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 Desember 2022
Direktur Program Pascasarjana

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

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Ivan Rifky Hendrawan
NIM : 21.51.1002
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Analisis Perbandingan Metode Bag Of Words, Tf-Idf, Word2vec Dan Doc2vec Pada Klasifikasi Teks Sentimen Masyarakat Terhadap Produk Lokal Di Indonesia

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Anggit Dwi Hartanto., 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 ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 13 Februari 2023
Yang Menyatakan,

Ivan Rifky Hendrawan

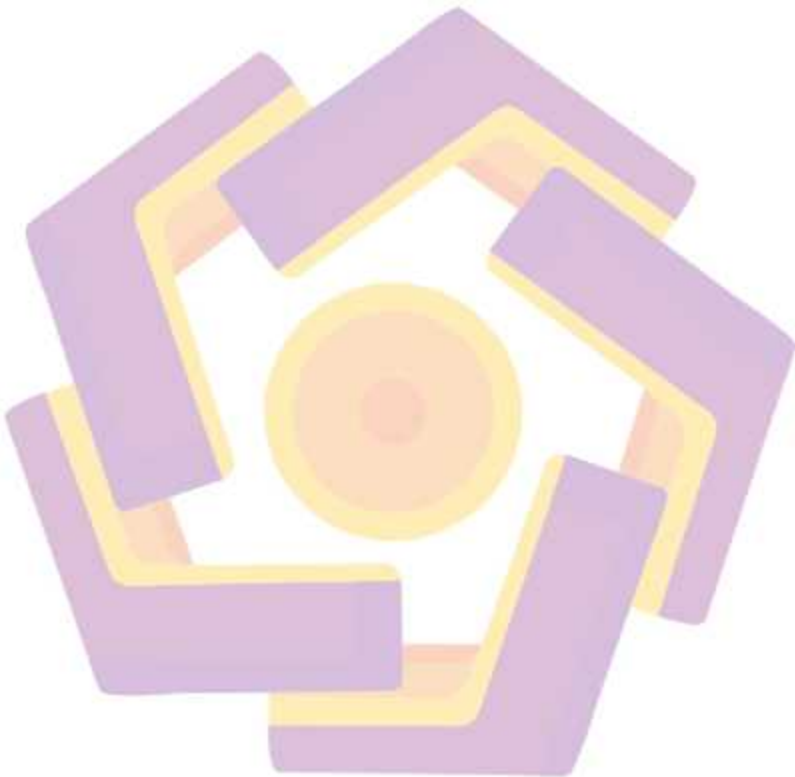
HALAMAN PERSEMBAHAN

Pertama dan paling utama, saya ucapkan syukur kepada Allah SWT yang telah memberikan kemudahan dan kelancaran dalam proses pembuatan tesis ini. Tesis ini saya persembahkan untuk :

1. Kedua orang tua dan adik (Bapak Arifin , Ibu Sri H. , Tiffani Anggi dan Azka Dian) yang senantiasa memberikan semangat dan doa, semoga selalu dalam lindungan-NYA.
2. Ibu Prof. Ema Utami, S.Si., M.Kom dan Bapak Anggit Dwi Hartanto., M.Kom, yang telah memberikan bimbingan aktif selama pelaksanaan penelitian, semoga mendapatkan banyak keberkahan dan dilancarkan segala urusannya.
3. Keluarga besar kelas MTI26A yang selalu membantu dan memberi saran demi kelancaran pengerjaan tesis, semoga selalu semangat dan sukses.
4. Keluarga besar yang selalu mendukung dan memberikan semangat dalam keadaan apapun. Serta semua pihak yang tidak dapat saya sebut satu persatu.

HALAMAN MOTTO

" Learning by Experience "



KATA PENGANTAR

Alhamdulillah puji syukur penulis panjatkan kehadirat Alloh SWT atas nikmat, karunia, taufik dan hidayah-Nya penulis dapat menyelesaikan tesis dengan judul : **Analisis Perbandingan Metode Bag Of Words, Tf-Idf, Word2vec Dan Doc2vec Pada Klasifikasi Teks Sentimen Masyarakat Terhadap Produk Lokal Di Indonesia**. Penulis menyampaikan terimakasih dan penghargaan yang sebesar-besarnya atas segala bantuan, dukungan dan doa kepada :

1. Bapak Prof. Dr. M. Suyanto, MM., selaku Rektor Universitas AMIKOM Yogyakarta.
2. Ibu Prof. Dr. Ema Utami, S.Si, M.Kom, selaku pembimbing utama dalam penulisan tesis saya, yang selalu mampu mengubah hal yang tidak mungkin menjadi mungkin bagi saya, memberikan banyak ilmu dan motivasi serta banyak hal luar biasa lainnya yang beliau berikan kepada saya.
3. Bapak Anggit Dwi Hartanto., M.Kom, selaku pembimbing pendamping dalam penulisan tesis saya, yang telah banyak memberikan arahan.
4. Orang tua dan adik yang selalu memberi support dan doa bagi saya dalam menyelesaikan segala masalah.

Yogyakarta, 6 Desember 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Review online sangat penting dalam mendukung keputusan pembelian karena dengan berkembangnya e-commerce, semakin banyak ulasan palsu, sehingga semakin banyak konsumen yang khawatir tertipu belanja online. Penelitian ini bertujuan untuk membandingkan performa (accuracy, presicion, recall dan f1-score) yang dihasilkan oleh model Algoritma XGBoost saat menggunakan Bag of Word, TF-IDF, Word2vec, dan Doc2vec pada dataset teks ulasan produk local di Shopee berbahasa Indonesia. Metode yang digunakan dalam penelitian ini adalah pengumpulan data, pembersihan data, pelabelan data, pra-pemrosesan data, klasifikasi dan evaluasi.

Proses pengikisan data menghasilkan 22.624 data yang terbagi menjadi 80% data latih dan 20% data uji. Data tersebut dibagi menjadi dua kelas, yaitu sentimen baik dan sentimen buruk. Hasil dari penelitian didapatkan metode Bag of Words menghasilkan F1 Score 0.932, TFIDF menghasilkan F1 Score 0.932, Word2vec menghasilkan F1 Score 0.934 dan untuk metode Doc2vec menghasilkan nilai F1-Score 0.933. Pada keempat model ektrasi tersebut diketahui bahwa Word2vec memilik nilai F1-Score tertinggi. Pada kasus penelitian ini untuk uji coba menggunakan empat metode tidak terlalu signifikan pada XGBoost, hal ini dikarenakan setiap metode menghasilkan nilai F1-Score yang tidak terlalu jauh untuk jarak perbedaan.

Penelitian yang akan datang penulis menyarankan untuk memaksimalkan peforma Doc2vec jika ingin meneliti permasalahan yang berkaitan dengan ulasan produk pada penelitian yang akan datang diperluhkan dataset yang lebih beragam agar hal ini dikarenakan metode Doc2vec ini masih bisa optimal jika didukung oleh jumlah kosakata yang beragam dengan cara mengubah vector_size dari 100 menjadi sesuatu yang lebih kecil atau lebih besar.

Kata kunci: Bag of Words, TFIDF, Word2vec, Doc2vec, XGBoost

ABSTRACT

Online reviews are very important in supporting purchasing decisions because with the development of e-commerce, there are more and more fake reviews, so more consumers are worried about being deceived by online shopping. This study aims to compare the performance (accuracy, precision, recall and f1-score) generated by the XGBoost Algorithm model when using Bag of Word, TF-IDF, Word2vec, and Doc2vec on a local product review text dataset at Shopee in Indonesian. The methods used in this research are data collection, data cleaning, data labeling, data pre-processing, classification and evaluation. The data scraping process produces 22,624 data which is divided into 80% training data and 20% test data. The data is divided into two classes, namely good sentiment and bad sentiment. The results of the research show that the Bag of Words method produces an F1 Score of 0.932, TFIDF produces an F1 Score of 0.932, Word2vec produces an F1 Score of 0.934 and the Doc2vec method produces an F1-Score value of 0.933. In the four extraction models, it is known that Word2vec has the highest F1-Score value. In the case of this study, for testing using four methods, XGBoost is not too significant, this is because each method produces an F1-Score value that is not too far away for the distance difference. In future research, the author suggests maximizing the performance of Doc2vec if you want to examine problems related to product reviews in future research, a more diverse dataset is needed so that this is because the Doc2vec method can still be optimal if it is supported by a variety of vocabulary numbers by changing the vector_size from 100 to something smaller or bigger.

Keyword: Bag of Words, TFIDF, Word2vec, Doc2vec, XGBoost

BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Penggunaan internet di Indonesia menurut Analisis Kepios telah mengalami kenaikan yang signifikan sebesar 2,1 juta (+1,0 persen) antara tahun 2021 dan 2022(Simon Kemp, 2022). Pertumbuhan pengguna internet yang semakin pesat menjadikan negara Indonesia sebagai pasar yang sesuai untuk *marketplace*. *Marketplace* merupakan aplikasi online yang memberikan fasilitas untuk kegiatan jual beli dari berbagai macam toko(Rohman et al., 2020). Berbagai fitur yang dimiliki oleh *marketplace* menjadikanya media transaksi online populer, seperti melakukan transaksi, mencari produk dan memberikan ulasan.

Sebuah proses menghasilkan ulasan produk pada dasarnya berkaitan dengan rating, yang membuat pengguna memberikan komentar yang bias, sebagai contoh untuk pengguna yang toleran, meskipun pengguna sangat tidak puas dengan produk, namun perangkingan tetap membuatnya memberikan komentar netral yang tidak dapat menunjukkan kualitas produk.(X. Wang et al., 2022).

Ulasan produk biasanya berisi testimoni buruk atau baik dari pengguna yang telah membeli produk. Dalam membeli sebuah produk, ulasan juga dapat digunakan pengguna sebagai bahan pertimbangan dan sebagai sumber data untuk membuat berbagai keputusan manajemen(Bi et al., 2019). Ulasan online menjadi factor penting dalam mendukung pengambilan keputusan pembelian online karena dengan berkembangnya *e-commerce*, semakin banyak ulasan palsu yang muncul di platform *e-commerce* untuk menyesatkan konsumen, dan semakin banyak

konsumen yang khawatir akan tertipu dalam belanja online(Q. Wang et al., 2022). Hal ini tidak bisa dipungkiri dikarenakan dari *review* pelanggan dapat diketahui tingkat kepuasan pelanggan terhadap produk yang telah dibeli(Kevin et al., 2020). Analisis sentimen adalah salah satu dari beberapa teknik yang dapat digunakan untuk mengolah dan mengklasifikasikan ulasan pengguna. Analisis sentimen juga dapat digunakan untuk mempelajari perasaan individu, pandangan, dan perilaku terhadap orang lain baik diri sendiri atau permasalahan dalam sebuah kegiatan yang dilakukan (Sistem et al., 2021).

Penelitian terkait sentimen analisis ulasan produk beberapa telah banyak dilakukan dengan metode dan hasil yang beragam. Seperti penelitian yang dilakukan oleh (Rohman et al., 2020)dan(Yennimar & Rizal, 2019) menggunakan algoritma KNN dan *Naive Bayes*, kedua metode ini juga pernah dilakukan pada analisis sentimen dalam mengklasifikasikan ulasan teks di *marketplace* Shopee(Sihombing et al., 2021). Pada penelitian (Basani et al., 2019) menggunakan metode *Support Vector Machine* (SVM) dan TF-IDF dapat dimanfaatkan sebagai penunjang dalam analisis sentimen saat pengambilan keputusan terhadap suatu produk. Kombinasi algoritma SVM dan TF-IDF juga pernah digunakan oleh (JAYADI, 2022) untuk menentukan klasifikasi pada ulasan produk pada lima *e-commerce* di Indonesia. Menurut penelitian(Yennimar & Rizal, 2019) *Naïve Bayes* menghasilkan nilai *accuracy* lebih baik dibanding *KNN* dalam mengklasifikasikan ulasan produk di *e-commerece*. Perbedaan *accuracy* antar metode tentunya juga dipengaruhi oleh tahapan *preprocessing*, dimana menurut(Hakim, 2021) *preprocessing* memiliki tujuan untuk mengubah

data mentah atau biasa dikenal dengan *raw* data yang dikumpulkan dari berbagai sumber menjadi informasi yang lebih bersih dan bisa digunakan untuk pengolahan selanjutnya. Selain *preprocessing*, *volume* data tentunya juga akan mempengaruhi pemilihan metode yang sesuai. Beberapa penelitian terdahulu hanya menggunakan data yang memiliki kelas data seimbang dan relative sedikit, padahal yang ditemui dilapangan tidak semua meiliki kelas data seimbang.

Permasalahan *data inbalance* ini juga dialami oleh penelitian(Amien et al., 2021) sehingga berpengaruh pada nilai *recall, prescision,* dan *f1-score*. Saat ini metode *Xtreeme Gradient Boost* sangat populer digunakan untuk memproses *data imbalance*, penelitian(Afifah et al., 2021) telah membuktikan asumsi bahwa XGBoost mampu menangani data tidak seimbang dengan nilai akurasi 96,24% dalam mengklasifikasikan ulasan pengguna pada aplikasi *Google Playstore*.

Menurut (Muslim et al., 2020) *Xtreeme Gradient Boost* adalah algoritma klasifikasi dan regresi dengan metode *ensemble* yang merupakan suatu varian dari algoritma *Tree Gradient Boosting* yang telah dimodifikasi dengan optimasi 10 kali lebih cepat dibandingkan *Gradient Boosting* lainnya. Penelitian (Bhoi & Joshi, 2018)mencoba komparasi enam algoritma *machine learning*, yaitu *Naïve Bayes*, SVM, DT, Random Forest *Extreme Gradient Boosting* (XGBoost), dan *Extra Trees Clasifier* untuk dua domain dataset bahasa Inggris. Algoritma XGBoost terbukti memiliki nilai tertinggi dengan akurasi 65,11 % untuk dataset teknologi dan 63,33% untuk dataset makanan. Hal ini didukung oleh penelitian selanjutnya(Akter et al., 2021) membahas perbandingan performa TF-IDF pada bahasa Bangla dengan menggunakan dua algoritma , yaitu XGBoost dan *Logistic*

Regression. Kombinasi TF-IDF dan XGBoost menghasilkan akurasi lebih tinggi dibanding dengan TF-IDF dan *Logistic Regression*. Menurut penelitian(Ramadhan Al-Mubaraq et al., 2021) fitur ekstraksi merupakan faktor penting yang dapat mempengaruhi tingkat akurasi pada tahap klasifikasi.

Ekstraksi fitur memiliki pengertian sama dengan penyisipan kata. *Word embedding* mulai dikembangkan sekitar tahun 2000 (Khattak et al., 2019). *Word embedding* menyematkan setiap kata dalam vektor padat, di mana setiap vektor mewakili pengucapan kata dalam ruang vektor (Kowsari et al., 2019). Posisi kata tersebut disimpulkan dari teks atau berdasarkan kata-kata terdekat. Penyematan kata dapat menangkap arti dan sintaks kata. Penyematan kata juga dapat digunakan untuk menyorot kata-kata serupa dalam kalimat., seperti *information retrieval*. Dalam penelitian yang akan dilakukan menggunakan *model Bag of Word* (BoW), *Term Frequency Inverse Document Frequency* (TF-IDF), *Word2vec* dan *Doc2vec*.

Metode TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (term) terhadap dokumen. TF-IDF pada penelitian sebelumnya memiliki kelebihan dalam memberikan frekuensi kemunculan kata pada sebuah dokumen, menurut (Flores & Jasa, 2020) TF-IDF menganggap semakin sedikit tingkat frekuensi yang muncul maka kata itu unik dan penting, TF-IDF juga dapat digunakan untuk mengukur tingkat similaritas dokumen dengan kata kunci. Selain meenggunakan TF-IDF metode *Bag of Words* juga dapat digunakan pada pembobotan kata. Fitur *Bag of Words* pada penelitian(Rozy et al., 2018) hanya mengenali ejaan bukan makna sehingga menimbulkan fitur

yang tidak relevan dan menyebabkan nilai evaluasi terutama *accuracy* menjadi tidak optimal sehingga diperluhkan metode lain yang cocok untuk merepresentasikan relasi *semantic* antar kata.

Penelitian(Agustiningsih et al., 2022) menguji penggunaan *word embedding* pada klasifikasi sentimen menggunakan model LSTM dua arah. Penelitian (Nurdin et al., 2020) membahas kinerja CNN dalam mengklasifikasikan teks ulasan menggunakan *Word2vec embedding* menggunakan ukuran *F-Measure*. Dalam mempresentasikan suatu kata, *Word2vec* mengimplementasi *neural network* untuk menghitung *contextual and semantic similarity* dari setiap kata yang berbentuk *one-hot encoded vectors*. Menurut (Nurdin, Seno aji, et al., 2020) *Word2vec* mampu merepresentasikan relasi *semantic* antar kata lebih baik dibanding dengan *TF-IDF*. Penelitian terkait *word embedding* juga telah dilakukan oleh (Jaya Hidayat et al., 2022) dalam mengklasifikasikan teks tentang respon masyarakat terhadap pembangunan di Pulau Rinca di media sosial twitter dengan menggunakan dua algoritma yaitu SVM dan *Logistic Regression* dengan *fitur extraction Doc2vec*, tidak seperti *Word2vec* yang dapat membuat representasi vektor dari kata-kata sambil mempertimbangkan konteks akun, Doc2vec dapat membuat representasi vektor dari dokumen(Edy et al., 2021). Model *Doc2vec* mampu merepresentasikan fitur sebagai vektor padat daripada representasi renggang konvensional yang umumnya mampu mengatasi masalah sinonim dan homonim yang sering dijumpai pada tugas NLP(Nawangsari et al., 2019). Penelitian yang dilakukan saat ini juga akan mencoba membuktikan asumsi bahwa metode *Word2vec* dan *Doc2vec* tidak begitu optimal ketika

memproses data yang *relative* sedikit dibanding metode tradisional TF-IDF dan *Bag of Words*. Pada tahun 2022 penelitian (Efrizoni et al., 2022) mengkomparasi ekstrasi fitur dalam klasifikasi teks menggunakan algoritma *machine learing* menggunakan 1000 ulasan. Untuk ekstraksi fitur *Doc2vec*, akurasi tertinggi pada algoritma SVM hanya sebesar 81% lebih rendah dibanding dengan TF-IDF 86% dan *Bag of Words* 82%. Sedangkan untuk metode *Word2vec* memiliki akurasi model dibawah 50% dari keenam algoritma *machine learning* yang digunakan.

Komparasi atau analisis perbandingan yang dilakukan pada penelitian sebelumnya maksimum hanya pada dua algoritma *machine learning* yang dikombinasikan dengan satu atau dua model *feature extraction* atau *word embedding*. Sementara pada penelitian ini, melakukan komparasi dengan empat model *feature extraction* (BoW, TF-IDf, Word2vec dan Doc2vec) dengan algoritma klasifikasi XGBoost. Dengan dilakukannya penelitian ini, diharapkan dapat menemukan fitur ekstraksi yang terbaik untuk klasifikasi ulasan teks pengguna berbahasa Indonesia di aplikasi *e-commerce*.

1.2. Rumusan masalah

Latar belakang diatas menghasilkan rumusan masalah sebagai berikut:

a. Bagaimana sentimen masyarakat pengguna aplikasi e-commerce terhadap produk lokal di Indonesia?
b. Berapa nilai performa (*accuracy, presicion, recall* dan *f1-score*) yang dihasilkan oleh model Algoritma *XGBoost* saat menggunakan *Bag of Word*, TF-IDF, *Word2vec*, dan *Doc2vec* pada dataset yang sama?

c. *Feature extraction* yang manakah yang menghasilkan *F1-Score* tertinggi dari model Algoritma XGBoost?

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

a. *Dataset* yang digunakan adalah dataset berbahasa Indonesia yang diambil dari aplikasi Shopee dengan kategori apparel terlaris dari 4 *brand local* di Indonesia yaitu Erigo Apparel, Major Minor, Minimal dan Nah Project
b. *Dataset* merupakan data berformat teks yang diambil menggunakan metode *scrapping*.
c. *Pre-processing* data dilakukan melalui teknik *word normalizer, remove punctuation, remove number, case folding, , stopword removal, tokenization* ,dan *stemming*.
d. *Platform* penelitian menggunakan Google Colaboratory.
e. Sentimen analisis diklasifikasikan menjadi 2, yaitu baik dan buruk.
f. Metode yang digunakan untuk melakukan sentiment analisis adalah metode *XGBoost* dengan beberapa kombinasi *feature extraction* yaitu *Word2vec, Doc2vec* , *Bag of Word,* TF-IDF.
g. Algoritma yang akan digunakan dalam melakukan klasifikasi adalah XGBoost+*Bag of Words,* XGBoost+TF-IDF, XGBoost+*Word2vec* dan XGBoost+*Doc2vec*.
h. Membandingkan tingkat performa (*accuracy, presicion, recall* dan *f1-score*) dari algoritma XGBoost + *Bag of Words,* XGBoost + TF-IDF, XGBoost +

Word2vec dan XGBoost + *Doc2vec* dalam studi kasus sentimen analisis respon masyarakat terhadap produk lokal di Indonesia.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

a. Mengetahui sentiment masyarakat pengguna aplikasi *e-commerce* terhadap produk lokal di Indonesia
b. Mengetahui nilai performa (*accuracy, presicion, recall* dan *f1-score*) yang dihasilkan oleh model Algoritma XGBoost saat menggunakan *Bag of Word*, TF-IDF, *Word2vec*, dan *Doc2vec* pada dataset yang sama.
c. Mengetahui *feature extraction* mana yang menghasilkan performa terbaik

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah:

a. Dapat menjadi pedoman pengembangan penelitian dalam menganalisis sentiment ulasan terhadap produk local di Indonesia pada aplikasi *ecommerce*.
b. Berkontribusi secara ilmiah terhadap penggunaan *feature extraction* (Bag of Word, TF-IDF,Word2vec, Doc2vec) dalam pengolahan data ulasan online berbahasa Indonesia pada klasifikasi sentimen menggunakan algoritma *XGBoost*.
c. Hasil klasifikasi sentiment diharapkan dapat menjadi bahan referensi untuk pihak terkait, khususnya perusahaan atau brand owner dalam membuat produk yang diminati dan dibutuhkan bagi masyarakat di Indonesia.

BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian terdahulu sangat penting bagi penulis sebagai bahan ajar untuk mengetahui hubungan antara penelitian masa lalu dengan penelitian yang akan datang. Tinjauan pustaka juga bertujuan untuk menunjukkan bahwa penelitian yang dilakukan oleh penulis sangat bermanfaat dan memiliki implikasi yang signifikan sebagai kontribusi penelitian terhadap ilmu pengetahuan. Di bawah ini adalah review jurnal penelitian-penelitian sebelumnya tentang metode yang digunakan penulis sebagai referensi.

Penelitian(Rohman et al., 2020) melakukan penelitian terkait sentimen analisis terhadap produk dengan mengambil data sejumlah 3341 ulasan di platform ecommerce menggunakan teknik *scrapping*, data diambil berdasarkan kategori fashion, yaitu kategori dengan penjualan terbanyak dalam rentan Mei 2019 sampai April 2020. Data kemudian melewati proses *preprocessing* agar optimal sebelum diolah. Untuk klasifikasi menggunakan dua metode yaitu *Naïve Bayes* dan KNN dengan hasil klasifikasi metode *Naïve Bayes* dengan *Unigram* akurasi 52,4% lebih rendah dibanding dengan KNN dengan *Unigra*m dengan nilai akurasi 76.2%. Untuk hasil akurasi kurang begitu optimal dikarenakan adanya ulasan yang tidak relevan terhadap suatu produk. Hal ini bisa dijadikan referensi bagi penulis saat ini yang akan meneliti topik yang berkaitan dengan ulasan produk dengan menambahkan proses *cleaning data* berupa menghapus setiap ada ulasan tidak relevan dan terdeteksi spam.

Hasil berbeda dikemukakan oleh penelitian(Yennimar & Rizal, 2019) dengan mengunakan algoritma sama bahwa *Naïve bayes* menghasilkan akurasi 89.00% lebih tinggi dibanding KNN dengan nilai akurasi sebesar 67.00%. Perbedaan cukup signifikan terletak pada metode *preprocessing* yang digunakan dimana pada penelitian(Rohman et al., 2020) menggunakan penambahan proses *normalizer* kata. *Normalizer* kata sangat penting dilakukan di fase *preprocessing* yang memungkinkan penelitian saat ini untuk mendeteksi dan mengoreksi kesalahan ketik, singkatan, dan kata-kata yang tidak baku sehingga model dapat memprosesnya lebih optimal.

Preprocessing pada analisis sentimen merupakan salah satu tahapan penting untuk data dalam proses penambangan karena data yang digunakan dalam proses penambangan tidak selalu dalam kondisi ideal untuk diproses. Penelitian(Sihombing et al., 2021) melakukan klasifikasi pada ulasan online terhadap produk Xiaomi Redmi Note 9 yang dijual pada website Shopee Indonesia. Data *review customer* dilakukan menggunakan teknik *web scraping* yang nantinya akan diklasifikasikan menggunakan algortima *Naïve Bayes*, akurasi yang dihasilkan sebesar 85%, hasil ini belum bisa dikatakan optimal karena masih terdapat banyak data yang tidak terklasifikasi dengan baik atau bisa disebut dengan data *misclassified*. Hal ini bisa dijadikan acuan pada penelitian yang akan dilakukan perlu memperhatikan struktur dan metode pada tahap *preprocessing* agar kerusakan data seperti *missing value, data redundant*, ataupun format data yang tidak sesuai dengan sistem bisa diatasi.

Penelitian (Nurdin, Seno aji, et al., 2020) membahas kinerja CNN dalam mengklasifikasikan teks menggunakan *Word2vec, GloVe*, dan *FastText*. *Dataset* yang digunakan dalam survei ini terdiri dari 20 newsgroup (Arsip KDD UCI, 1999a), sekitar 18.846 artikel, 134.142 kosakata, dengan 20 topik yang dibagi menjadi 11.314 data latih dan 7.532 data uji. Hasil klasifikasi di ukur menggunakan *F-Measure* secara berturut-turut adalah 0.925, 0.958, dan 0.979. *Word2vec* di penelitian sebelumnya tergolong mempunyai *F-Measure* yang lebih rendah dibanding *Fastext* dan *Glove*. Hal ini membuat penulis terkait untuk mencoba memaksimalkan performa *Word2vec* dengan kombinasi metode XGBoost dimana menurut (Afifah et al., 2021) algoritma ini cukup baik dalam menangani data yang tidak seimbang. Hal ini didukung dengan penelitian yang telah dilakukan sebelumnya XGBoost menghasilkan akurasi 96,24% dengan *dataset* yang dikumpulkan dari *Google Play Store* menggunakan pustaka scraper *Google Play* di *Python* sejumlah 12.969 data ulasan dari 1 Januari – 30 September 2021.

Penelitian ini juga akan mencoba memaksimalkan performa dari *Doc2vec Embedding* dengan XGBoost dimana pada penelitian(Jaya Hidayat et al., 2022) dua model *Doc2vec* digunakan yaitu distributed model, dan *Bag of Words* yang dikombinasikan dengan SVM dan *Logistic Regression* hanya menghasilkan tingkat akurasi diatas 75%. Penelitian selanjutnya (Akter et al., 2021) membahas perbandingan kinerja TF-IDF pada dua algoritma, yaitu XGBoost dan *Logistic Regression*. *Dataset* yang gunakan adalah ulasan produk yang berbeda dari bahasa Bangla sejumlah 7905 data. Hasil penelitian ini membuktikan bahwa dari data

pembelajaran mesin dengan TF-IDF XGBoost mencapai akurasi 90.56% lebih tinggi dibandingan dengan TF-IDF *Logistic Regression* yang mencapai akurasi 90.33%, hal ini menjadikan alasan utama penulis menggunakan algoritma ini karena XGBoost merupakan salah satu algoritma yang paling populer dan paling banyak digunakan karena algoritma ini termasuk algoritma yang *powerful*.

Penelitian (JAYADI, 2022) membandingkan beberapa algoritma pembelajaran mesin untuk menentukan cara terbaik untuk analisis sentimen pada ulasan produk terkandung dalam lima *e-commerce* di Indonesia dengan data sejumlah 14742 reviews. Dari hasil klasifikasi algoritma SVM mendapat akurasi paling tinggi sebesar 95.875% dibanding algoritma *Decission Tree* 95.762%, *Random Forest* 95.838%, dan *Gradien Boost* 95.801%. Perbedaan akurasi setiap algoritma juga dipengaruhi oleh tingkat kompleksitas sebuah data, untuk memaksimalkan sebuah model di penelitian ini data yang akan digunakan lebih banyak dari penelitian sebelumnya sehingga hasilnya lebih konkrit.

Berdasarkan beberapa penelitian di atas, maka pada penelitian ini akan melakukan klasifikasi ulasan terhadap produk lokal di *e-commerce* menggunakan pendekatan sentimen analisis yang hampir sama dengan (Rohman et al., 2020) dan (JAYADI, 2022). Perbedaan yang akan dilakukan terletak pada *preprocessing*, fitur ekstraksi dan model pembelajaran mesin yang berbeda. Dalam *preprocessing* juga akan mengoptimalkan normalisasi kata mengapdatasi dari (Yennimar & Rizal, 2019) agar ulasan yang terdapat singkatan, salah ketik dan tidak baku bisa diolah oleh model dengan optimal.

Fitur ekstrasi yang akan digunakan TF-IDF mengadaptasi dari (Sihombing et al., 2021) serta tambahan *Bag of Words* dari peneliti. Selain itu juga akan membandingkan model *embedding* (Word2vec dan Doc2vec) yang sama dengan penelitian (Nurdin, Seno aji, et al., 2020) dan (Jaya Hidayat et al., 2022). Selanjutnya untuk fitur *Bag of Word,* TF-IDF, dan *word embedding* itu akan diuji coba menggunakan model yang sama dengan (Afifah et al., 2021) dan (Akter et al., 2021) yaitu XGBoost.

2.2. Keaslian Penelitian

Tabel 2. 1Matriks literatur review dan posisi penelitian

Analisis Perbandingan Metode *Bag of Words*, TF-IDF, Word2vec Dan Doc2vec pada Klasifikasi Teks Sentimen Masyarakat Terhadap Produk Lokal Di Indonesia

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atauKelemahan	Perbandingan
1	*Natural Language Processing on Marketplace Product Review Sentiment Analysis*	Arif Nur Rohman, Ema Utami, Rizqa Luviana Musyarofah, Suwanto Raharjo ICORIS (2020)	Mengetahui perbandingan dua algoritma dengan mengklasifikasikan ulasan online dari produk *e-commerce* untuk diketahui sentiment positif atau *negative* dengan algoritma KNN dan *Naïve bayes*	Sentimen analisis diklasifikasikan menjadi positi dan negatif dengan menggunakan *metode Naïve Bayes* dan KNN dengan hasil klasifikasi metode *Naïve Bayes* dengan *Unigram* akurasi 52,4% lebih rendah dibanding dengan KNN dengan *Unigram* dengan nilai akurasi 76.2%.	Di peneltian selanjutnya disarankan menerapkan deteksi spam untuk menemukan *review* yang tidak relevan tentang suatu produk sehingga dapat meningkatkan kualitas analisis sentimen.	Metode *Naïve Bayes* dan KNN tidak dilakukan dalam penelitian selanjutnya. Untuk scrapping data dilakukan dengan kategori *most relevan* agar lebih optimal dan menghindari *review* spam.
2	*Comparison of Machine Learning Classification Algorithms*	Yennimar, Reyhan Achmad Rizal SinkrOn (2019)	Membandingkan hasil klasifikasi product review menggunakan algoritma *Naïve Bayes* dengan KNN dengan pembobotan TF-IDF	Berdasarkan hasil penelitian yang telah dilakukan, *Naïve Bayes* menghasilkan akurasi 89.00% lebih tinggi dibanding KNN	Pada penelitian ini tidak adanya normalisasi kata, jadi beberapa kata singkatan yang seharusnya ada arti jadi tidak bisa diproses	Penelitian selanjutnya juga menggunakan pembobotan TF IDF dengan menambahkan *normalize* kata pada preprocessing

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atauKelemahan	Perbandingan
3	*Sentimen Analisis Customer Review Produk Shopee Indonesia Menggunakan Algortima Naïve Bayes Classifier Loemongga*	Loemongga Oktaria Sihombing , Hannie, Budi Arif Dermawan, Edumatic: Jurnal Pendidikan Informatika (2021)	Untuk mengklasifikasikan Ulasan customer dari produk Xiaomi Redmi Note 9 yang dijual pada website Shopee Indonesia untuk diketahui kelas sentimennya.	disimpulkan menggunakan kombinasi Naïve *Bayes Classifier* dan TF-IDF. Algoritma *Naïve Bayes* pada penelitian ini diketahui memiliki tingkat akurasi 85%.	banyak data yang tidak terklasifikasi dengan baik atau disebut dengan misclassified data. Ada total 436 data yang tidak sepenuhnya diklasifikasikan	Penelitian selanjutnya akan lebih mengoptimalkan ke *preprocessing* agar kerusakan data seperti *missing value, data redundant*, ataupun format data yang tidak sesuai dengan sistem bisa diatasi.
4	*Sentiment Analysis of COVID-19 Vaccines in Indonesia on Twitter Using Pre-Trained and Self-Training Word Embeddings*	Kartikasari Kusuma Agustiningsih, Erna Utami, and Omar Muhammad Altoumi Alsyaiban, Jurnal Ilmu Komputer dan Informasi (2022)	menguji penggunaan *word embedding* pada klasifikasi sentiment vaksin di Indonesia menggunakan Model LSTM dua arah	Hasil dari penelitian ini adalah jarak akurasi antara *Fastext* dan kata *GloVe* embedding sangat kecil di mana kata *GloVe embedding* menghasilkan akurasi 92.55% yang sedikit lebih tinggi daripada *Fastext* 92.33%	Saran untuk penelitian diberikan dataset yang berisi lebih banyak tweet komprehensif.	Pada penelitian yang akan dilakukan metode LSTM tidak akan dilakukan akan tetapi akan membandingkan dua word embedding lain yaitu Word2vec dan Doc2vec dengan algoritma XG Boost dengan data teks yang diambil dari ulasan produk yang bermacam macam strukturnya.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atauKelemahan	Perbandingan
5	Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks	Arliyanti Nurdin , Bernadus Anggo Seno Aji, Anugrayani Bustamin , Zaenal, Jurnal Tekno Kompak Abidin(2020)	Mengetahui kinerja CNN dalam mengklasifikasikan teks menggunakan *word embedding* Word2vec, GloVe, dan FastText	Hasil penelitian diukur menggunakan *FMeasure* secara berturut-turut untuk *dataset* 20 *newsgroup* adalah *Word2vec* 0.925, *Glove* 0.958, dan Fastext 0.979. *Word2vec* dan *GloVe* tidak dapat merender vektor untuk kata-kata yang tidak ada dalam korpus (tanpa kosakata).	Meskipun kinerja eksperimental terbaik dicapai dengan menggunakan FastText dan penyisipan kata. Namun, perbedaan kecil dalam kinerja ini menunjukkan bahwa kinerja penyematan tiga kata ini bersaing, dan juga diperiksa untuk data seimbang atau tidak seimbang dalam penelitian ini.	Penelitian yang akan dilakukan akan mencoba mengoptimalkan kinerja *Word2vec* yang nantinya akan dikombinasikan dengan algoritma XGBoost dimana, algoritma ini mempunyai asumsi mampu mengklasifikasikan data yang tidak seimbang
6	*Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier*	K. Afifah, I. N. Yulita, and I. Sarathan, International Conference on Artificial Intelligence and Big Data Analytics (2021)	Mengklasfikasikan sentiment positif dan *negative* pada ulasan aplikasi Telemedicine dan untuk membuktikan asumsi lain tentang XGBoost yang mengatakan XGBoost cukup baik dalam menangani data yang tidak seimbang.	kombinasi TF-IDF dengan Algoritma XGBoost menghasilkan akurasi 96,24%, studi ini membuktikan bahwa XGBoost cukup baik dalam menangani data yang tidak seimbang	Dalam penelitian ini menyarankan menggunakan salah satu teknik untuk menangani data yang tidak seimbang dan membandingkannya dengan penelitian ini.	Di penelitian sebelumnya untuk menguji model hanya menggunakan metode akurasi dimana akurasi sangat cocok jika datasetnya seimbang, di penelitian selanjutnya akan menggunakan teknik *uji F-Score*, *F-Score* sangat cocok digunakan untuk mengukur data yang tidak seimbang

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atauKelemahan	Perbandingan
7	Bengali Sentiment Analysis of E-commerce Product Reviews using K-Nearest Neighbors	M. T. Akter, M. Begum, and R. Mustafa, ICICT4SD (2021)	Untuk mengetahui perbandingan kinerja TF-IDF pada dua algoritma, yaitu XGBoost dan *Logistic Regression* pada klasifikasi text sentiment	Hasil penelitian ini membuktikan bahwa data machine learning mencapai akurasi 90.56% lebih tinggi pada TF-IDF XGBoost disbanding TF-IDF *Logistic Regression* yang mencapai akurasi 90.33%	Di penelitian selanjutnya perlu menggunakan lebih banyak kumpulan data untuk melatih model.	Di penelitian ini tetap menggunakan kombinasi TF-IDF dan Algoritma XGBoost, namun akan menggunakan data ulasan lebih banyak dibanding dengan penelitian sebelumnya
8	Application of Sentiment Analysis on Product Review E-Commerce	Yuniarta Basani, Harris V. Sibuea, Sinta Ida Patona Sianipar and Jen Presly Samosir, Journal of Physics: Conference Series (2019)	Penelitian ini bertujuan untuk mengklasifikasikan setiap *review* produk menjadi orientasi positif, negatif atau netral dan juga menghasilkan ringkasan ulasan produk berdasarkan fitur produk untuk membantu proses membaca,pengambilan keputusan.	Klasifikasi pada penelitian ini menggunakan dua algoritma untuk dibandingkan, hasil akurasi masing masing diketuhi bahwa algoritma SVM memiliki nilai 85% lebih tinggi dibanding *Naïve Bayes* hanya menghasilkan nilai 79%.	Tahap *pre-processing* pada dataset yang digunakan terdiri dari *Lemmatization* dan *stemming* akan menggunakan *library Stanford Core* NLP.	Pada penelitian selanjutnya, akan menggunakan *case folding, filtering, stemming,* tokenisasi dan normalisasi kata .

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atauKelemahan	Perbandingan
9	*Sentiment Analysis on Tokopedia Product Online Reviews Using Random Forest Method*	B. Warsito and A. Prahutama, ICENIS (2020)	Penelitian ini bertujuan mengklasifikasikan respon pelanggan Tokopedia terhadap kualitas produk dan keramahan penjual	Dalam mengevaluasi model, ulasan dikelompokkan sebagai *sentiment* positif dan negatif menggunakan algoritma *Random Forest* yang dikombinasikan dengan TF-IDF. Data yang digunakan berjumlah 14432 sebagai data *training* dan 6186 sebagai data *testing*. Akurasi yang dihasilkan pada model klasifikasi ini sebesar 97.05%.	Dalam paper ini sudah dijelaskan menggunakan *preprocessing* hanya saja tidak dijelaskan secara jelas berapa tahapan preprocessing yag dilakukan, serta tidak digambarkannya alur proses dan metodologi penelitian.	Pada penelitian selanjutnya akan di gambarkan seluruh tahapan *preprocessing* beserta alur proses dari metode penelitian
10	*Sentiment Analysis Of Indonesian E-Commerce Product Reviews Using Support Vector Machine Based*	Siti Fidyanti Nurfadila Hadju, Riyanto, Jayadi, Journal of Theoretical and Applied Information Technology (2022)	membandingkan beberapa algoritma pembelajaran mesin untuk menentukan cara terbaik untuk analisis sentimen pada ulasan produk terkandung dalam lima *E-Commerce* di Indonesia	Dari hasil klasifikasi algoritma SVM mendapat akurasi paling tinggi sebesar 95.875% dibanding algoritma *Decission Tree* 95.762%, *Random Forest* 95.838%, dan *Gradien Boost* 95.801%.	Dalam, penelitian lebih lanjut, diharapkan dapat menggunakan algoritma yang lebih maju. Selain itu, kumpulan data yang digunakan bisa lebih banyak sehingga hasilnya lebih konkrit, dan terakhir model ini dapat diterapkan pada bidang lain yang relevan.	Penelitian selanjuntya akan fokus ke pengoptimalan algoritma XGBoost dengan beberapa scenario uji coba model.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atauKelemahan	Perbandingan
11	Sentiment analysis of twitter data related to Rinca Island development using Doc2vec and SVM and logistic regression as classifier	Tirta Hema Jaya Hidayat, Yova Ruldeviyani, Achmad Rizki Aditama, Gusti Raditia Madya, Ade Wija Nugraha, Muhammad Wijaya Adisaputra, Procedia Computer Science (2021)	melakukan klasifikasi *sentiment* terhadap respon masyarakat terhadap pembangunan di Pulau Rinca oleh Pemerintah Indonesia di media sosial twitter	Penelitian ini dilakukan untuk menganalisis yang terbagi menjadi tiga kategori: pro, kontra, dan netral. Ada dua model Doc2vec digunakan dalam penelitian ini, yaitu distributed model, dan bag of words dan menggunakan algoritma klasifikasi SVM dan logistic regression. Setiap kombinasi model dan classifier memiliki tingkat akurasi di atas 75% dan menunjukkan bahwa hampir semua menentang pembangunan Pulau Rinca.	Penelitian di masa depan juga harus memastikan bahwa memiliki dataset yang seimbang antara label. Selain itu, penelitian lebih lanjut harus mencoba mengklasifikasikan berdasarkan topik, tidak hanya berdasarkan sentiment	Penelitian selanjutnya Doc2vec dan Bag of Words akan menggunakan model XGBoost bukan dengan algoritma SVM maupun logistic regression. Klasifikasi sentimen hanya ada dua yaitu baik dan buruk.

2.3. Landasan Teori

2.3.1 Natural Languange Processing

NLP adalah aplikasi pembelajaran mesin dan teknik komputasi yang dapat memahami dan merepresentasikan teks lisan dan tulisan (Bhattacharjee, 2018). Menurut(Farzindar, A. A., & Inkpen, 2020) model NLP dilatih menggunakan bahasa yang formal dan sesuai kaidah. Terdapat dua pendekatan yang dapat dilakukan untuk mengolah teks dari sosial media yang pertama melatihnya secara berulang atau yang kedua melakukan normalisasi teks.

Lebih lanjut (Bhattacharjee, 2018) menguraikan beberapa hal yang dapat dipecahkan menggunakan NLP, antara lain:

1. Pemodelan topik: Secara umum, teks berhubungan dengan topik membantu pembaca untuk mendapatkan inti dari dokumen dan deskripsi tingkat tinggi tentang apa yang terjadi.
2. Klasifikasi kalimat: mengklasifikasikan teks ke dalam label yang berbeda.
3. *Machine Translator*, merupakan model terjemahan pembelajaran mesin yang mampu melatih dari beberapa sumber dengan tingkat daya prediksi yang tinggi.
4. Sistem tanyajawab (QA): Fokus di sini adalah membangun sistem yang secara otomatis menjawab pertanyaan berdasarkan pertanyaan yang diajukan orang dalam bahasa alami.
5. Analisis sentimen: tentang memahami kebutuhan dan maksud yang dibagikan pengguna saat membicarakan sesuatu.

6. Deteksi nama entitas: untuk mengekstrak dan mengklasifikasikan entitas atau informasi spesifik sesuai dengan beberapa kategori yang telah ditentukan sebelumnya, seperti orang, organisasi, geografi, dan sebagainya.

2.3.2 Analisis Sentimen

Analisis sentimen merupakan salah satu bidang ilmu yang digunakan untuk mengetahui persepsi masyarakat baik positif , negatif atau netral terhadap tokoh, organisasi maupun isu yang sedang terjadi. Menurut(Farzindar, A. A., & Inkpen, 2020) analisis sentimen yang menggunakan data yang bersumber dari sosial media mempunyai tantangan tersendiri karena pengguna di sosial media terkadang menggunakan susunan kalimat yang tidak baku, penulisan kata yang tidak sesuai kamus, maupun gaya bahasa yang tidak formal.

Selanjutnya (Hidayatullah et al., 2021) menjelaskan bahwa pekerjaan dasar dalam sebuah analisis sentimen yaitu mengklasifikasikan *polarity* dari sebuah dokumen, kalimat atau teks, apakah pendapat yang diekspresikan pada teks tersebut positif, negatif atau netral. Menurut (Rambocas dan Pacheco 2018) analisis sentimen dalam lingkup perusahaan berguna untuk mendapatkan pemahaman yang lebih baik tentang agregat pelanggan pendapat dan sikap terhadap produk.

2.3.3 Text Preprocessing

Preprocessing merupakan salah satu tahapan dalam mempersiapkan data sebelum digunakan dalam suatu model. Pra-pemrosesan analisis sentimen

merupakan salah satu fase penting dari data dalam proses penambangan karena data yang digunakan dalam proses penambangan tidak selalu dalam keadaan ideal untuk diproses(Sihombing et al., 2021). Teknik pre-processing yang dapat digunakan pada teks yang bersumber dari sosial media, diantaranya menghapus *username*, *hashtag*, URL, *punctuation*, huruf yang terulang dalam sebuah kata, tanda spasi yang berlebihan, stopwords dan tweet yang duplikat.

Menurut(Zhang et al., 2019) tahap *preprocessing* dimulai dengan normalisasi, di mana kata atau frasa direduksi menjadi bahasa standar. Proses selanjutnya adalah sanitasi data, yaitu menghilangkan angka, tanda baca, dan tanda unik. Langkah selanjutnya adalah huruf kecil. Di sini semua kata dinormalisasi menjadi huruf kecil. (Kusumaningrum et al., 2021). Setelah semua kata diubah menjadi huruf kecil, proses *stopword removal* dimulai. Proses ini adalah proses menghilangkan kata penghubung atau konjungsi. Setelah menghapus konjungsi, proses stemming berikut yang mengubah kata-kata kembali ke kata aslinya. (Amin et al., 2021).

Tahap akhir dari proses akhir preprocessing adalah tokenization, yang mengubah kalimat atau paragraf menjadi token dengan banyak arti(Septian et al., 2019). Penelitian (Jaya Hidayat et al., 2022) tokenisasi adalah proses memecah kalimat menjadi bagian-bagian yang lebih kecil (kata-kata). Proses pemecahan ini bervariasi di masing-masing Bahasa. Misalnya, dalam bahasa Inggris dan Indonesia, proses penyelesaian dapat dipecah berdasarkan spasi atau tanda baca.

2.3.4 Fitur Extraction

Ekstraksi fitur memiliki pengertian sama dengan penyisipan kata. *Word embedding* mulai dikembangkan sekitar tahun 2000 (Khattak et al., 2019). *Word* embedding menyematkan setiap kata dalam vektor padat, di mana setiap vektor mewakili pengucapan kata dalam ruang vektor (Kowsari et al., 2019). Penyematan kata juga dapat digunakan untuk menyorot kata-kata serupa dalam kalimat, seperti *information retrieval*. Dalam mengubah kata menjadi vector terdapat beberapa metode yaitu untuk metode tradisional dapat menggunakan *Bag-of-Words* dan TF-IDF. Sementara untuk yang popular ada *Word2vec*, *Word2vec* memiliki beberapa pengembangan diantaranya seperti *Doc2vec* yang dapat mengubah dokumen atau kalimat menjadi *vector*.

Kemudian ada *FastText* adalah metode penyisipan kata lain yang merupakan perpanjangan dari model *Word2vec*. Alih-alih mempelajari vektor untuk kata-kata secara langsung, *FastText* mewakili setiap kata sebagai n-gram karakter(Nurdin, Anggo Seno Aji, et al., 2020). Jadi, misalnya, ambil kata, "Artificial" dengan n=3, representasi fastText dari kata ini adalah <ar, art, rti, tif, ifi, fic, ici, ial, al>, di mana tanda kurung siku menunjukkan awal dan akhir kata sedangkan *GloVe* merupakan representasi kata untuk menghasilkan *word embedding* untuk yang dapat digunakan untuk word *similarity, word analogy*. Pada prinsipnya *GloVe* memperoleh hubungan semantik antar kata berdasarkan *co-occurrence matrix*.

2.3.4.1 Bag- of -Words

Menurut(Ishihara, 2021) BoW atau *Bag of Words* merupakan salah satu metode paling sederhana dalam mengubah data teks menjadi vektor yang dapat dipahami oleh komputer. Algoritma ini mendeklarasikan hubungan antara mendokumentasikan dan mengevaluasi kata-kata berdasarkan istilah frekuensi dalam dokumen. Menurut(Jaya Hidayat et al., 2022) suatu kalimat, paragraf, atau seluruh dokumen yang direpresentasikan sebagai kumpulan kata, tanpa memperhatikan tata bahasa atau urutan kemunculan kata.

Metode bag-of-word adalah teks dalam bentuk kalimat atau dokumen yang direpresentasikan sebagai kumpulan kata-kata multiset yang terkandung di dalamnya, tanpa memperhatikan urutan kata atau tata bahasa, tetapi tetap melestarikan keragamannya(Trisari et al., 2020). Pendekatannya sangat sederhana dan fleksibel, dan dapat digunakan dalam berbagai cara untuk mengekstrak fitur dari dokumen.

Model hanya memperhatikan apakah kata-kata yang diketahui muncul dalam dokumen, bukan di mana dalam dokumen. Dalam pendekatan ini, dapat dilihat histogram kata-kata dalam teks, yaitu mempertimbangkan setiap kata dihitung sebagai fitur(Jurafsky & Martin, 2008).

2.3.4.2 TF-IDF

Menurut(Sihombing et al., 2021) pembobotan TF-IDF (Term Frequency Inverse Document Frequency) merupakan suatu metode untuk mengubah data dari data teks menjadi data numerik dan pembobotan setiap kata atau fungsi. Nilai TF-

IDF dihasilkan dari perkalian antara TF dan IDF. Semakin tinggi nilai TF-IDF maka semakin rendah frekuensi munculnya sebuah istilah, dan sebaliknya semakin rendah nilai TF-IDF maka semakin tinggi frekuensi munculnya istilah tersebut. Penelitian (Flores & Jasa, 2020) mengatakan bahwa TF-IDF menganggap semakin sedikit tingkat frekuensi yang muncul maka kata itu unik dan penting, TF-IDF juga dapat digunakan untuk mengukur tingkat similaritas dokumen dengan kata kunci.

Bobot kata tinggi jika sering muncul dalam satu dokumen dan rendah jika sering muncul di banyak dokumen (Septian et al., 2019). Pendekatan TF-IDF dapat dilihat pada Persamaan (1):

$W_{ij} = TF_{ij} \times \log\left(\frac{D_i}{df_i}\right)$ ..... Persamaan (1)

(Willy et al., 2019)

Dimana

TF_{ij} = jumlah kata ke-i yang muncul pada dokumen ke-j.

D_i = jumlah dokumen atau data,

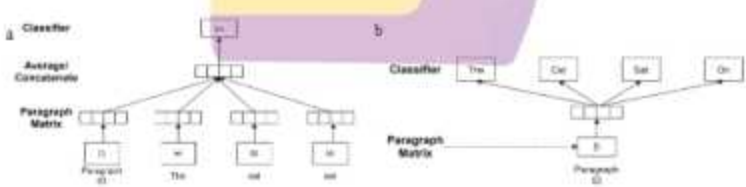
df_i = jumlah dokumen yang mengandung kata ke-i

2.3.4.3 DO2VEC

Word embedding merupakan teknik NLP yang mengubah sebuah kata dasar menjadi vector bernilai *real*(Kurniawan & Maharani, 2020). Doc2vec adalah model untuk mewakili nilai numerik dari dokumen, tidak peduli berapa lama dokumen itu. Penelitian (Jaya Hidayat et al., 2022) mengungkapkan alasan menggunakan model ini adalah bahwa dokumen tidak seperti kata-kata dengan

struktur logis. *Vektor* yang *Doc2vec* dapat digunakan untuk beberapa tujuan, seperti menemukan kesamaan antar kalimat/paragraf(Shuai et al., 2018). Doc2vec dapat membuat representasi *vector* dari dokumen(Edy et al., 2021). Model *Doc2vec* mampu merepresentasikan fitur sebagai vektor padat daripada representasi renggang konvensional yang umumnya mampu mengatasi masalah sinonim dan homonim yang sering dijumpai pada tugas NLP(Nawangsari et al., 2019).

Menurut (Purnama, 2021) *Doc2vec* memiliki dua model berbeda yang dapat digunakan diantaranya versi memori terdistribusi dari vektor paragraf (PV-DM) dan versi kata dari vektor paragraf (PV-DBOW). PV-DM memiliki kemiripan dengan model *Continuous Bag of Words* dari model Word2vec, tetapi menambahkan vektor tambahan untuk menggambarkan keunikan dokumen. Di sisi lain, versi *Skip-Gram* dari *Word2vec* menggunakan versi *words* dari *Paragraph Vector* (PV-DBOW). Model Skip-Gram mencoba memprediksi konteks di sekitar kata target yang diberikan. Berikut adalah arsitektur penelitian *Doc2vec* dapat dilihat pada gambar 2.1



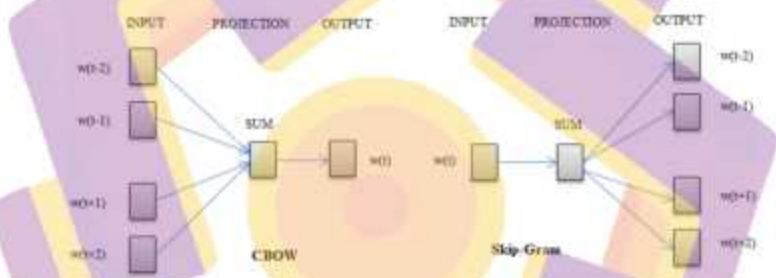
Gambar 2.1 (a) PV-DM; (b) PV-DBOW (Jaya Hidayat et al., 2022)

Representasi PV-DM ditunjukkan pada Gambar 1 a. Di sisi lain, PVDBOW hanya menggunakan ID paragraf untuk memprediksi kata-kata yang dipilih dari sampel acak. Seperti yang ditunjukkan pada Gambar 1 b di sana tidak ada kata-kata yang digunakan untuk melatih model. Ada beberapa perbedaan antara PV-DM dan PV-DBOW. PV-DM memprediksi satu kata dari 4 input. Di sisi lain, PV-DBOW memprediksi empat kata dari 1 input, seperti yang terlihat pada Gambar 1 a dan Gambar 2 b. Perbedaan kedua adalah undian PV-DM kata-kata dari sekitar target, dan PV-DBOW menarik kata dari seluruh paragraf. Terakhir, PV-DBOW cenderung menyimpan lebih sedikit data daripada PV-DM karena hanya bobot *SoftMax* yang disimpan(Jaya Hidayat et al., 2022).

2.3.4.4 Word2vec

Word2vec adalah salah satu metode *embedding word* yang berguna untuk merepresentasikan kata menjadi sebuah *vector*. Sejak kemunculannya, model word embedding ini banyak digunakan dalam penelitian NLP (Nurdin, Anggo Seno Aji, et al., 2020). *Word2vec* merepresentasikan kata-kata ke dalam vektor yang dapat membawa arti semantik kata. Model penyisipan kata ini adalah aplikasi pembelajaran tanpa pengawasan yang menggunakan jaringan saraf yang terdiri dari lapisan tersembunyi dan terhubung penuh(Nurdin, Seno aji, et al., 2020). Dimensi dari matriks bobot pada setiap layer adalah jumlah dengan kata dalam korpus dikalikan dengan jumlah hidden *neuron* pada *hidden layer*-nya. Matriks bobot pada *hidden layer* dari model yang telah dilatih digunakan untuk mentransformasikan kata ke dalam *vector*.

Menurut (Kurniawan & Maharani, 2020) metode *Word2vec* dibagi menjadi dua algoritma penyisipan kata utama, *Continuous Bag of Words* (CBOW) dan *Skip Gram*. Algoritma CBOW biasanya mengenali panjang tertentu dalam dokumen input sebagai satu kata. Algoritma *Skip-Gram* dapat memprediksi konteks suatu kata dengan melihat seberapa dekat kata tersebut dengan kata lain, tetapi sebelum dan sesudah kata tersebut. Arsitektur CBOW dan *Skip gram* bisa dilihat pada gambar 2.2



Gambar 2.2 Arsitektur Cbow dan Skip Gram (Eligüzel et al., 2022)

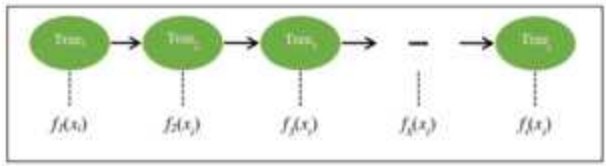
Dalam gambar 2.2 dapat dijelaskan bahwa model CBOW, *Word2vec* menggunakan kata-kata yang ada di sebelah kiri dan kanan kata target dan dibatasi dengan window untuk memprediksi kata target tersebut. Sedangkan *skip-gram* menggunakan sebuah kata untuk memprediksi kata-kata yang ada di sebelah kiri dan kanan kata tersebut yang dibatasi oleh window(Eligüzel et al., 2022). Masing-masing kata yang digunakan sebagai input di-encode ke dalam one-hot vector. Perbedaan dari kedua model tersebut adalah model prediksi kata(Ay et al., 2018). Pada CBOW, terdapat intermediate layer yang akan melakukan kalkulasi

average pada vector kata-kata input karena CBOW menerima sejumlah n kata sebagai input.

2.4. Algoritma XGBoost

Extreme Gradient Boosting (XGBoost) adalah teknik pembelajaran mesin untuk analisis dan klasifikasi regresi berdasarkan Gradient Boosting Decision Tree (GBDT). XGBoost metode pertama kali diperkenalkan oleh (Friedman et al., 2000), dalam penelitiannya Friedman menghubungkan antara *boosting* dan optimasi dalam membangun *Gradient Boosting Machine* (GBM). Penelitian lain (Afifah et al., 2021) *Extreme Gradient Boosting* adalah algoritma pembelajaran mesin *ansambel* berbasis pohon keputusan yang menggunakan kerangka kerja penambah *gradien*. Pendekatan *ensemble classifers* mengadopsi beberapa algoritma pembelajaran untuk mendapatkan kinerja yang lebih baik (Behera et al. 2016). XGBoost juga dapat digunakan untuk time series(Jurafsky & Martin, 2008).

Metode *boosting* digunakan untuk membangun model baru yang memprediksi kesalahan dari model sebelumnya. Penambahan model baru dilakukan sampai koreksi kesalahan tidak memungkinkan lagi. Dengan menggunakan penurunan gradien untuk meminimalkan kesalahan saat membangun model baru, algoritme ini disebut peningkatan gradien. Gambar 2.3 menunjukkan proses perhitungan algoritma XGBoost. (Mo et al., 2019)



Gambar 2.3. Diagram Skema dari Algoritma XGBoost

Nilai prediksi pada langkah t diumpamakan $y_i^{(t)}$ dengan:

$$\hat{y}_i^{(t)} \sum_{k=1}^{t} f_k(x_i) \tag{3.1}$$

$f_k(x_i)$ menggambarkan model pohon. Untuk y_i diperoleh dari perhitungan berikut:

$$\hat{y}_i^{(0)} = 0$$

$$\hat{y}_i^{(1)} = f_i(x_1) = \hat{y}_i^{(0)} + f_i(x_1)$$

$$\hat{y}_i^{(2)} = f_i(x_1) + f_2(x_2) = \hat{y}_i^{(1)} + f_2(x_2)$$

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_1)$$

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) \tag{3.2}$$

Dimana

$\hat{y}_i^{(t)}$ = Final tree model

$\hat{y}_i^{(t-1)}$ = Model pohon yang dihasilkan sebelumnya

$f_t(x_1)$ = Model baru yang dibangun

t = Jumlah total model dari *base tree models*

Untuk algoritma XGBoost, dalam menentukan jumlah pohon dan *depth* merupakan hal penting. Permasalahan dalam menentukan algoritma yang optimum dapat diubah dengan pencarian klasifikasi baru yang dapat mengurangi *loss function,* dengan target fungsi kerugian ditunjukan pada persamaan (3.3) berikut

$$Obj^{(t)} = \sum_{k=1}^{t} l(\ _{i}\hat{y}_i^{(t)}) + \sum_{i=i}^{t} \dot{U}(f_i) \qquad (3.3)$$

Dimana:

-$\hat{y}_i^{(t)}$ = Nilai Prediksi

-$y_{i.}$ = Nilai Aktual

- $l(y_{i.}\hat{y}_i^{(t)}$ = lost function

- $\dot{U}(f_i)$ = istilah regularisasi

Karena model ensemble tree pada persamaan (3.3) merupakan fungsi sebagai parameter dan tidak dapat dioptimalkan menggunakan metode pengoptimalan tradisional pada ruang Euclidean. Sehingga digantikan dengan model dilatih dengan cara aditif, dengan menggunakan $\hat{y}_{gg}^{(t)}$ pada prediksi ke-I dan iterasi ke-t. Dalam meminimalkan loss function maka ditambahkan f_t sehingga didapatkan persamaan (3.4) sebagai berikut

$$Obj^{(t)} = \sum_{i=1}^{t} l(y_{i.}\hat{y}_i^{(t-1)}) + (f_t(x_i)) + \dot{U}(f_t) + Contant \qquad (3.4)$$

Selanjutnya target akhir dari *loss function* diubah menjadi persamaan (3.5), kemudian dilatih sesuai dengan target *loss function* berikut:

$$Obj^{(t)} = \sum_{i=1}^{t} \left[(g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \dot{U}(f_t) \qquad (3.5)$$

Di mana

$$g_i = \partial_{y_i^{t-1}} l(y_i, \hat{y}_i^{(t-1)})$$

$$h_i = \partial^2_{y_i^{t-1}} l(y_i, \hat{y}_i^{(t-1)})$$

gi dan hi merupakan urutan pertama dan kedua *statistic gradient* pada *loss function*.

2.5. Confusion Matrix

Menurut (Kotu & Deshpande, 2019) *Confusion matrix* adalah tabel perhitungan yang didasarkan pada evaluasi kinerja model klasifikasi berdasarkan jumlah item studi yang diprediksi dengan benar dan salah. Dengan kata lain, matriks konfusi memberikan rincian kesalahan klasifikasi (Gorunescu, 2010).

Pada klasifikasi biner terdapat beberapa nilai evaluasi yang sering digunakan. Dapat dilihat berdasarkan nilai *Confusion Matrix* (Sokolova & Lapalme, 2009):

- *Accuracy* adalah perbandingan antara jumlah prediksi yang benar pada semua label dengan jumlah semua data dengan asumsi label pada semua kelas seimbang.
- *Precision* adalah rasio antara label yang diprediksi dengan benar dan sampel yang diprediksi menjadi data positif. Metrik ini mewakili tingkat presisi yang ditebak model terhadap label pada dataset.
- *Recall* didefinisikan sebagai rasio antara sampel positif benar dan sampel positif total dalam dataset. *Recall* mencerminkan kemampuan model untuk mengenali label negatif.

- *F1 Score* adalah *harmonic mean* atau antara *Precision* dan *Recall*. Matriks *F1 Score* sering digunakan pada kasus dimana dataset tidak seimbang. Keempat matriks ini dijelaskan pada formula di bawah ini.

$$Accuracy = \frac{TP+TN}{TP+\{P+FN+TN} \quad (3.6)$$

$$Precision = \frac{TP}{FP+TP} \quad (3.7)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.8)$$

$$F1 = 2\frac{Precision \ *Recall}{Precision \ +Recal} \quad (3.9)$$

BAB III
METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Adapun jenis, sifat dan pendekatan penelitian adalah sebagai berikut:

3.1.1 Jenis dan Sifat Penelitian

Jenis penelitian ini adalah penelitian eksperimental. Dimana penelitian ini melakukan pengujian tingkat *F1-Score* yang tertinggi menggunakan metode XGBoost + *Bag of Words*, XGBoost + TF-IDF, XGBoost + *Word2vec* dan XGBoost + *Doc2vec* dengan jumlah *dataset* yang sama. Pengujian ini dilakukan untuk mengetahui metode yang lebih akurat dan tepat dalam melakukan sentimen analisis pada data produk yang diambil di e-commerce dengan kategori apparel lokal terlaris pada 4 brand local di Indonesia.

Penelitian ini bersifat deskriptif, karena menggambarkan suatu objek yang akan diteliti dan menjabarkan hasil pengujian-pengujian yang dilakukan pada *dataset* yang ada untuk dapat diketahui metode mana yang memiliki akurasi, presisi, *recall, f1 score* terbaik.

3.1.2 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif. Dikarenakan penelitian ini jenis penelitian yang spesifikasinya adalah sistematis, terencana, dan terstruktur dengan jelas sejak awal hingga pembuatan desain penelitiannya. Penelitian yang akan dilakukan akan menghasilkan berupa angka dan grafik hasil eksperimen setiap model perbandingan ekstrasi fitur. Pada tahap kesimpulan hasil

penelitian ini akan ditampilkan dalam bentuk grafik yang menampilkan nilai *accuracy*, *prescision*, *recall*, dan *f-score* antar hasil klasifikasi dengan 4 metode perbandingan yaitu, *Bag of Words*, TF-IDF, *Word2vec* dan *Doc2vec* pada algoritma klasifikasi XGBoost.

3.2. Metode Pengumpulan Data

Untuk mendapatkan data yang dibutuhkan dalam penelitian ini, yaitu data ulasan pengguna, adapun metode yang dilakukan adalah sebagai berikut :

Data yang digunakan pada penelitian ini merupakan jenis data primer yang dikumpulkan langsung menggunakan teknik *scrapping* dari aplikasi *ecommerce*. Aplikasi yang akan dituju adalah Shopee yang merupakan situs jual beli pilihan terbaik pilihan pelanggan (https://www.topbrand-award.com/).

Gambar 3.1 Halaman Aplikasi Shopee

Scrapping dilakukan dengan cara mengambil beberapa brand lokal terkenal menurut (Kemenparekraf, 2022) yaitu Erigo Apparel, Major Minor, Minimal dan Nah Project. Pada penelitian ini *scrapping* digunakan untuk

mengumpulkan informasi yang ada dalam *ecommerce*. *Scrapping* bekerja secara otomatis, dimana informasi yang dikumpulkan berdasarkan kategori apparel terlaris tiap brand yang bersangkutan. Alat yang digunakan pada penelitian ini adalah Google Collaboratory dan bahasa pemrograman python. Dari data yang sudah dikumpulkan, dapat dianalisa dan diambil informasi yang akan di *mining* menggunakan teks *preprocessing* dan algoritma XGBoost. Dalam melakukan implementasi teknik *scrapping* pada aplikasi *e-commerce* ini menggunakan metode yang ditunjukan pada gambar 3.2 berikut.

Gambar 3.2 Alur Pengumpulan data

Pada gambar 3.2 tersebut dijelaskan secara bertahap tentang proses yang dilakukan oleh peneliti dalam mengimplementasikan teknik *scrapping*. Diawali dengan cara menentukan brand yang akan di ambil ulasan produknya, kemudian ditentukan untuk kategori produk apparel terlaris meliputi produk pakaian seperti kemeja, jaket, kaos, hoodie dan polo. Kemudian tahapan selanjutnya adalah menggunakan teknik scrapping menggunakan bahasa pemrograman python, proses *scrapping* dilakukan pada brand yang bersangkutan seperti Erigo Apparel, Major Minor, Minimal dan Nah Project. Setelah set data hasil pencarian ditampilkan maka hasil dari *scrapping* data berupa data teks yang berisi ulasan pengguna dan rating akan disimpan dalam bentuk file .csv yang nantinya akan

digabungkan membentuk kumpulan data. Penggabungan data dilakukan dengan cara *copy* dan *paste* file yang sudah disimpan dan dijadikan satu file. Kemudian data yang sudah digabungkan itu akan disimpan dalam format .csv. *Dataset* yang sudah ada kemudian akan dilakukan proses data cleaning dimana ulasan yang duplikat dan tidak relevan akan dihapus.

3.3. Metode Analisis Data

Metode analisis data yang akan digunakan dalam penelitian ini adalah menggunakan *Text Preprocessing*, diantaranya dengan melakukan:

1. Pengumpulan Data Awal

Data awal ini merupakan data yang didapat dari hasil *scrapping* dari ke empat *brand local* dengan kategori apparel terlaris masing-masing. Data yang digunakan adalah teks ulasan dan rating.

2. Data *Cleaning*

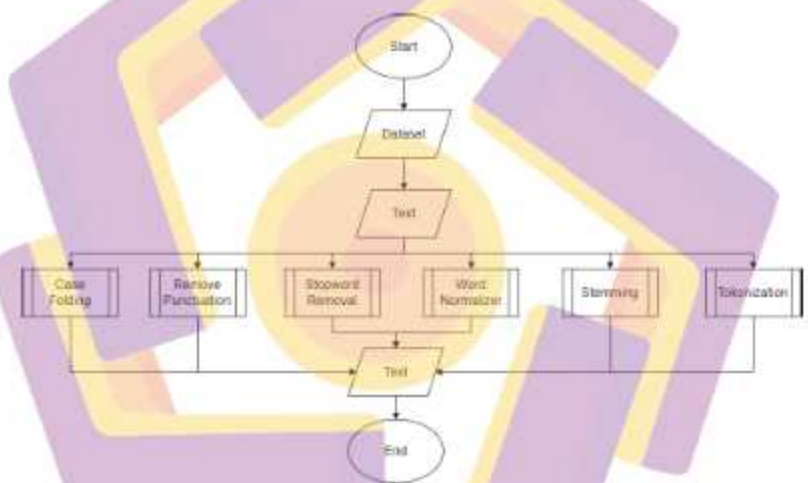
Data yang telah dikumpulkan pada langkah sebelumnya dibersihkan terlebih dahulu agar lebih optimal. Ulasan yang spam atau yang tidak relevan akan dihapus agar proses sentimen kedepanya nanti bisa optimal. Ulasan yang redundan atau duplikat juga akan dihapus.

3. Data *Labelling*

Data yang telah dibersihkan kemudian diberikan label baik dan buruk. Kriteria buruk adalah untuk rating 1 sampai 3, sedangkan kriteria baik rating 4 sampai 5.

4. Data *Pre-processing*

Preprocessing yang dilakukan pada penelitian ini bertujuan untuk menjadikan data mentah menjadi data yang siap pakai untuk dijalankan di proses selanjutnya. Pada tahap *preprocessing* menggunakan library NLTK, atau *Natural Language Toolkit*, NLTK berisi berbagai perpustakaan pemrosesan teks dengan banyak kumpulan data pengujian. *Preprocessing* terdiri dari beberapa sub-proses seperti pada gambar 3.3.



Gambar 3.3 Alur Preprocessing

Pada gambar 3.3 dapat dijelaskan bahwa pada penelitian ini menggunakan 6 tahapan diantaranya *Case Folding, Remove Punctuation, Stopword Removal, Word Normalizer*, Stemming, dan *Tokenization*. Tahapan pertama dimulai dari *case folding*, proses ini mengubah huruf kapital menjadi huruf kecil. Proses

selanjutnya adalah *remove punctuation*, dimana angka, tanda baca, dan tanda unik seperti "[^a-zA-Z#]" akan dihilangkan.

Langkah selanjutnya adalah *Stopword Removal*, proses ini merupakan proses menghilangkan konjungsi. Setelah proses penghapusan *stopword*, kemudian proses *word normalize* atau normalisasi kata, hal ini dilakukan mengingat tidak semua ulasan tersaji dalam bahasa dengan EYD yang benar, kemudian adalah *stemming*, dimana kata dikembalikan ke kata dasar. Pada proses *stemming* menggunakan menggunakan library Sastrawi, library ini digunakan untuk mereduksi kata-kata infleksi dalam Bahasa Indonesia ke bentuk dasarnya (stem). Dan proses *preprocessing* terakhir adalah *tokenization*, dimana sebuah kalimat atau paragraf akan diubah menjadi sebuah token dengan arti tertentu.

5. Data *Splitting*

Data yang telah selesai di *pre-processing* akan dibagi menjadi dua bagian yaitu 80% data train, 20% data test. Data train digunakan untuk melatih model XGBoost pada klasifikasi ulasan teks, sedangkan data test digunakan untuk memprediksi ulasan teks.

6. Tahap Pelatihan dengan XGBoost

Model yang akan digunakan untuk training klasifikasi adalah XGBoost. Dalam tahap pelatihan ini akan menggunakan empat skenario yang pertama adalah menggunakan model *Bag of Words* + XGBoost, kemudian TFIDF + XGBoost, Word2vec+XGBoost, dan Doc2vec+XGBoost. Selanjutnya, model yang telah dilatih akan diuji di setiap skenario dengan menggunakan data yang *test* sebesar 20%. Komposisi jumlah data yang akan dilatih akan dibagi menjadi

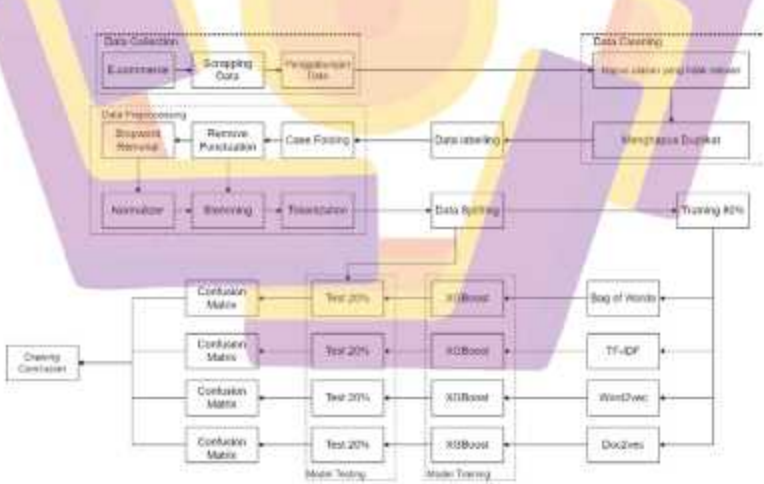
beberapa skenario meliputi data 1000 dan jumlah total dari data yang ada setelah selesai proses cleaning.

7. Tahap Conclusion

Conclusion merupakan tahap untuk melakukan evaluasi performa (akurasi, presisi, recall dan f1 score) yang dihasilkan oleh model Algoritma XGBoost saat menggunakan *Bag of Word*, TF-IDF, Word2vec, dan Doc2vec pada *dataset* yang sama. Dalam tahap ini akan dibandingan model mana yang memiliki tingkat *F1 Score* yang paling tinggi menggunakan *confusion matrix*.

3.4. Alur Penelitian

Alur penelitian yang akan dilakukan adalah sebagai berikut.



Gambar 3.4. Alur Penelitian

Pada gambar 3.4 dapat dijelaskan bahwa pada penelitian ini tahapan pertama dimulai dari *data collection*, dimana data ulasan dan rating dikumpulkan

pada aplikasi *ecommerce* menggunakan teknik Scrapping dengan Google Collaboratory platform dan Python sebagai bahasa eksekusinya. Kemudian setelah dataset yang sudah dikumpulkan melalui scrapping akan memasuki tahapan *data cleaning*, dimana pada tahap ini ada dua proses yang dilakukan yaitu menghapus ulasan yang tidak relevan dan menghapus ulasan yan duplikat. Sebeluh diolah dan dianalisa dataset akan memasuki proses Preprocessing, tahap ini berguna untuk memastikan kualitas data baik sebelum digunakan saat analisis data. Data yang telah selesai di *pre-processing* akan dibagi dua bagian yaitu 80% data train, 20% data test.

Tahap selanjuntya setelah pembagian data adalah ekstrasi fitur dimana metode yang digunakan ada empat yaitu Bag-of-Words, TF-IDF, Word2vec dan Doc2vec. Setelah itu adalah tahap klasifikasi, model yang akan digunakan untuk training klasifikasi adalah XGBoost. Dalam tahap pelatihan ini akan menggunakan empat skenario yang pertama adalah menggunakan model *Bag of Words* + XGBoost, kemudian TFIDF + XGBoost, Word2vec+XGBoost, dan Doc2vec+XGBoost. Selanjutnya, model yang telah dilatih akan diuji di setiap skenario dengan menggunakan data yang *test* sebesar 20%. Komposisi jumlah data yang akan dilatih akan dibagi menjadi beberapa skenario meliputi data 1000 dan jumlah total dari data yang ada setelah selesai proses cleaning yaitu 22.624 .

Tahap selanjutnya adalah *Confusion matrix*, pada tahapan ini dilakukan evaluasi kinerja model klasifikasi berdasarkan jumlah item studi yang diprediksi dengan benar dan salah. Pada tahapan inilah dapat diketahui untuk tingkat *accuracy, prescision, recall* dan *f1-score*. Tahap akhir dari penelitian ini adalah

drawing conclusion, dalam tahap ini akan dibandingan model mana yang memiliki tingkat *F1 Score* yang paling tinggi menggunakan *confusion matrix.*

BAB IV
HASIL DAN PEMBAHASAN

Bab ini memaparkan mengenai hasil penelitian, mencangkup penjelasan pengumpulan data, *text preprocessing*, fitur ekstrasi, pengklasifikasian data dan menguji akurasi, presisi, *recall* dan *f1-Score*.

4.1 Pengumpulan Data

Data yang digunakan pada penelitian ini berupa data hasil *scrapping* dari aplikasi *ecommerce*. Proses *scrapping* data menghasilkan 25.581 data. Dalam pengambilan data di Aplikasi Shopee membutuhkan API Shopee, pada implementasinya ke dalam sistem menggunakan python dengan kode sebagai berikut :

```
r = re.search(r'i\.(\d+)\.(\d+)', url)
shop_id, item_id = r[1], r[2]
ratings_url = 'https://shopee.co.id/api/v2/item/get_ratings?filter=0&flag=1&itemid={item
_id}&limit=20&offset={offset}&shopid={shop_id}&type=0'
```

Proses *scrapping* dilakukan secara otomatis pada masing – masing brand kategori apparel terlaris, dimana nantinya setelah di *execute* ulasan dan rating akan otomatis diambil sehingga data yang tersisa setelah proses *cleaning* sejumlah 22.624 data. Langkah selanjutnya beberapa data yang sudah dikumpulkan akan digabungkan menjadi satu *file* dengan format .csv. Berikut adalah contoh ulasan hasil *scrapping* yang dapat dilihat pada tabel 4.1

Tabel 4. 1Sample set data hasil scrapping dari aplikasi ecommerce Shopee

No	Ulasan	Rating
1	pakctnya sudah sampai tadi sore, ternyata lebih cepat dri estimasinya „ bahan nya halus adem nyaman dipakai ... bakalan order lgi nih next time 😉😉	5
2	orderan ku banyak.pegel kalo harus ngasih ulasan satu2 😁 pokonya jgn ragu belanja di toko miiiiii... 👍👍👍👍mantap dah	4
3	Niat Jualan Apa Ngga Seh, Jualan Ga Serius Sama sekali, udah lewat batas pengiriman barang ga datang juga...\nSaya kecewa sekali..\nDi tutup aja ini aplikasinya...	2
4	coba ya ganti model iklannya pake balita umur 2 thn.... kecewa 100%	1
5	Di foto kaos dewasa pas datng kaos ank 5thn kecewa	2
6	Paketnya sudah sampai tadi sore, ternyata lebih cepat dri estimasinya „ di bahan nya Halus adem nyaman dipakai ... bakalan order lgi nih next time ðŸ˜‰ðŸ˜‰	5

Data yang sudah dikumpulkan kemudian diberikan label baik dan buruk. Kriteria buruk adalah untuk rating satu sampai tiga, sedangkan kriteria baik rating empat sampai lima. Untuk kriteria baik akan diberikan label 1 dan untuk kriteria buruk akan diberikan label 0. Setelah proses pelabelan diketahui untuk sentimen baik berjumlah 18.427 data dan sentimen buruk berjumlah 4.197. Sehingga untuk lebih lengkapnya dapat dilihat pada tabel 4.2

Tabel 4. 2Labeling Data

No	Ulasan	Label
1	paketnya sudah sampai tadi sore, ternyata lebih cepat dri estimasinya ,, bahan nya halus adem nyaman dipakai ... bakalan order lgi nih next time 😉😉	1
2	orderan ku banyak.pegel kalo harus ngasih ulasan satu2 😄 pokonya jgn ragu belanja di toko iniiiiii... 👍👍👍👍mantap dah	1
3	Niat Jualan Apa Ngga Seh, Jualan Ga Serius Sama sekali, udah lewat batas pengiriman barang ga datang juga...\nSaya kecewa sekali..\nDi tutup aja ini aplikasinya...	0
4	coba ya ganti model iklannya pake balita umur 2 thn.... kecewa 100%	0
5	Di foto kaos dewasa pas datng kaos ank 5thn kecewa	0
6	Paketnya sudah sampai tadi sore, ternyata lebih cepat dri estimasinya ,, di bahan nya Halus adem nyaman dipakai ... bakalan order lgi nih next time ðŸ˜‰ðŸ˜‰	1

4.2 Preprocessing Text

Setelah menyelesaikan tahapan pengumpulan *dataset* yang berupa teks ulasan, selanjutnya akan dilakukan tahapan pengolahan data dengan tujuan memaksimalkan teks saat proses klasifikasi. Pada penjelasan tahap ini akan

menjelaskan proses *pre-processing* yang digunakan. Tahapan yang akan dilakukan adalah sebagai berikut:

4.2.1 Case Folding

Pada proses *case folding* ini dilakukan karena pada *dataset* tersebut tidak selalu memiliki struktur sehingga peran dari tahapan ini adalah menyamaratakan penggunaan huruf untuk proses klasifikasi teks, contoh hasil proses ini dapat dilihat pada tabel 4.3 dimana kalimat sudah sama rata tanpa adanya huruf capital.

Tabel 4. 3Hasil dari Case Folding

Teks Kotor	Hasil
Paketnya sudah sampai tadi sore, trnyata lebih cepat dri estimasinya ,, di bahan nya Halus adem nyaman dipakai ... bakalan order lgi nih next time ðŸ˜‰ðŸ˜‰	paketnya sudah sampai tadi sore, trnyata lebih cepat dri estimasinya ,, di bahan nya halus adem nyaman dipakai ... bakalan order lgi nih next time ðŸ˜‰ðŸ˜‰

4.2.2 Remove Punctuation

Pada proses *remove punctuation* dilakukan untuk penghilangan atau penghapusan tanda baca, angka, dan karakter khusus, pada proses ini sehingga hasil dari proses *remove punctuation* dapat dilihat pada tabel 4.4

Tabel 4. 4Hasil dari Remove Punctuation

Sebelum	Sesudah
paketnya sudah sampai tadi sore, trnyata lebih cepat dri estimasinya ,, di bahan nya halus adem nyaman dipakai ... bakalan order lgi nih next time ðŸ˜‰ðŸ˜‰	paketnya sudah sampai tadi sore trnyata lebih cepat dri estimasinya di bahan nya halus adem nyaman dipakai bakalan order lgi nih next time

4.2.3 Stopword removal

Pada proses *stopword removal* dilakukan penghilangan kata penghubung atau konjungsi, pada tahap ini juga mengabaikan kata dengan jumlah huruf < 3 sehingga hasilnya dapat dilihat pada tabel 4.5

Tabel 4. 5Hasil dari Stopword Removal

Sebelum	Sesudah
paketnya sudah sampai tadi sore trnyata lebih cepat dri estimasinya di bahan nya halus adem nyaman dipakai bakalan order lgi nih next time	paketnya sudah sampai tadi sore trnyata lebih cepat estimasinya bahan halus adem nyaman dipakai bakalan order next time

4.2.4 Word Normalizer

Normalisasi kata dilakukan untuk mengubah kalimat yang mengandung slang words atau kata tidak baku dan kata singkat untuk diubah menjadi kata baku. Program akan secara otomatis mencocokan kata yang ada dengan kamus normalisasi jika mengandung slangword sama dengan kamus normaliasi pada *row* pertama maka kata tersebut akan diganti menjadi bentuk baku sesuai dengan yang ada di kamus normaliasi pada *row* kedua, hasil dari proses normaliasi kata dapat dilihat pada tabel 4.6

Tabel 4. 6Hasil Word Normalizer

Sebelum	Sesudah
paketnya sudah sampai tadi sore trnyata lebih cepat estimasinya bahan halus adem nyaman dipakai bakalan order next time	paketnya sudah sampai tadi sore ternyata lebih cepat estimasinya bahan halus adem nyaman dipakai bakalan order next time

4.2.5 Stemming

Proses *stemming* dilakukan untuk mengubah kata berimbuhan menjadi kata dasar dengan menghilangkan imbuhan yang ada pada kata tersebut. Proses menggunakan data dari hasil normalisasi kata. Proses *stemming* diawali dengan mengimport data dari *library sastrawi*, kemudian data yang ada tersebut didefinisikan dengan *Stremmer Factory* dan membuat kamus *stemmer* dengan menggunakan fungsi *create_stemmer*. Contoh hasil dari proses ini dapat dilihat pada tabel 4.7 dimana kata-kata yang memiliki imbuhan telah diubah menjadi kata dasarnya.

Tabel 4. 7Hasil Stemming

Sebelum	Sesudah
paketnya sudah sampai tadi sore ternyata lebih cepat estimasinya bahan halus adem nyaman dipakai bakalan order next time	paket sudah sampai tadi sore ternyata lebih cepat estimasi bahan halus adem nyaman pakai bakal order next time

4.2.6 Tokenization

Pada tahap *tokenization*, pada tahap ini kalimat akan dipecah menjadi perkata, dapat dilihat pada tabel 4.8

Tabel 4. 8Hasil Tokenisasi

Sebelum	Sesudah
paket sudah sampai tadi sore ternyata lebih cepat estimasi bahan halus adem nyaman pakai bakal order next time	"paket", "sudah", "sampai","tadi","sore", "ternyata", "lebih"," cepat" ,"estimasi", "bahan", "halus", "adem" ,"nyaman", "pakai", "bakal", "order","next", "time"

4.2.7 Worcloud

Berikut adalah tampilan ulasan setelah di *preprocessing* akan ditampilkan dalam bentuk *worcloud*, *Wordcloud* adalah visualisasi di mana kata-kata yang paling sering muncul dalam ukuran besar dan kata-kata yang lebih jarang muncul dalam ukuran yang lebih kecil. seperti pada gambar 4.1

Gambar 4.1 *Worcloud* hasil ulasan yang sudah di *preprocessing*

Berdasarkan frekuensi kemunculan kata terbanyak pada gambar 4.1 yang muncul dalam wordcloud untuk 5 kata dengan frekuensi sering muncul ada bagus, sesuai, bahan, barang, kirim, dan adem. Untuk lebih jelasnya untuk frekuensi kemunculan kata dari worcloud akan disajikan dalam grafik pada gambar 4.2

Gambar 4.2 Frekuensi Jumlah Kemunculan Kata

Dari gambar 4.2 dapat diketahui bahwa ulasan produk yang telah di scrapping dan diolah datanya untuk topic yang sering dibicarakan meliputi kata bagus memiliki frekuensi kemunculan 4081, sesuai 3967, bahan 2413, barang 1869 dan kirim 1788.

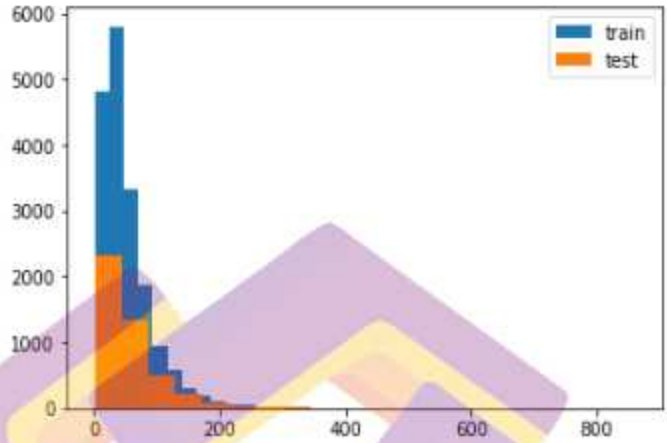
4.3 Data Splitting

Data yang telah selesai di *pre-processing* akan dibagi menjadi dua bagian yaitu 80% data *train*, 20% data *test*. Dalam proses pembagian *dataset* ini dilakukan juga pengacakan pada data agar data tersebut menjadi lebih bervariasi. Parameter yang digunakan dalam proses pembagian data ini adalah *text_size* : 0.20 dan *random state* : 42.

Keterangan dari parameter yang digunakan :

- text_size : fungsi yang digunakan untuk pembagian data *train* dan *test*.
- Random state : Banyaknya pengacakan kata yang akan dilakukan dalam proses pembagian data ini.

Untuk pembagian data dapat dilihat pada gambar 4.3



Gambar 4.3 Tampilan Pembagian Data *Train* dan Data *Test*

Data sejumlah 22.624 dibagi menjadi 180.99 sebagai data *training* yang nantinya kan digunakan untuk melatih model XGBoost pada klasifikasi ulasan teks, sedangkan 4.525 sebagai data *test* yang kemudian akan digunakan untuk memprediksi ulasan teks.

4.4 Fitur Ekstrasi

Setelah melalui proses pembagian data *train* dan data *test*, tahapan selanjutnya yaitu data training akan digunakan konversi atau di ekstrak menjadi angka sebelum diinputkan ke proses klasifikasi menggunakan XGBoost. Fitur ekstraksi merupakan faktor penting yang dapat mempengaruhi tingkat akurasi pada tahap klasifikasi, pada tahapan ini akan dijelaskan penghitungan untuk fitur ekstraksi yang dilakuan, pada tahapan ini ada 4 metode yang akan dilakukan yaitu *Bag of Words*, *TF- IDF*, *Word2vec*, dan *Doc2vec*.

4.4.1. *Bag of word*

Bag of Words merupakan salah satu metode paling sederhana dalam mengubah data teks menjadi vektor yang dapat dipahami oleh komputer. Algoritma ini mendeklarasikan hubungan antara mendokumentasikan dan mengevaluasi kata-kata berdasarkan istilah frekuensi dalam dokumen. Pada penelitian ini data *training* yang berbentuk teks akan diekstrak menjadi *vector*, Berikut perintah yang akan dijalankan pada program seperti di bawah ini menggunakan bahasa python.

```
bow_vectorizer =CountVectorizer(max_df=0.90, min_df=2, max_features=1000, stop_words=stop
word_indo)
```

Contoh sederhana penghitungan *Bag of Words*:

1. Ulasan 1 : Kaos disini bagus nyaman
2. Ulasan 2 : Kaos disini biasa saja
3. Ulasan 3 : Kaos disini tidak bagus

Kemudian setelah itu bisa dilihat terbentuknya sebuah korpus / kamus kata seperti berikut.

"kaos" "disini" "bagus" "nyaman" "biasa" "saja" "tidak"

Perlu diperhatikan sebelumnya, bahwa dalam membentuk korpus, *Bag- of-Words* hanya menghitung kata secara unik. Artinya, setiap kata yang berulang hanya akan ditulis sekali. Langkah berikutnya akan dilakukan proses hitung frekuensi kemunculan kata di korpus tersebut kepada ketiga ulasan sebelumnya. Setelah itu diberikan nilai 1 jika kata tersebut muncul pada sebuah ulasan dan 0 jika tidak muncul. Agar lebih mudah dalam memahmi dapat dilihat pada tabel 4.9

Tabel 4. 9Tabel Bag of Words

	Kaos	disini	bagus	nyaman	biasa	saja	Tidak
Ulasan 1	1	1	1	1	0	0	0
Ulasan 2	1	1	0	0	1	1	0
Ulasan 3	1	1	1	0	0	0	1

Maka diketahui untuk bentuk vector dari ketiga ulasan tersebut hasilnya adalah

- Vektor review 1 [1,1,1,1,0,0,0]
- Vektor review 2 [1,1,0,0,1,1,0]
- Vektor review 3 [1,1,1,0,0,0,1]

4.4.2.TF-IDF

Tahap TF-IDF Ini dilakukan dengan mengalikan dua metrik: berapa kali sebuah kata muncul dalam sebuah dokumen, dan frekuensi dokumen terbalik dari kata tersebut di seluruh kumpulan dokumen. Sejatinya, TF-IDF merupakan gabungan dari 2 proses yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Ini adalah metode lain yang didasarkan pada metode frekuensi tetapi berbeda dengan pendekatan bag-of-words dalam arti bahwa metode ini memperhitungkan tidak hanya kemunculan kata dalam satu dokumen (atau ulasan) tetapi di seluruh *Corpus*. Berikut perintah yang akan dijalankan pada program seperti di bawah ini menggunakan bahasa python.

```
tfidf_vectorizer = TfidfVectorizer(max_df=0.90, min_df=2, max_features=1000, stop_words=stop word_indo)
```

Contoh sederhana penghitungan TF-IDF:

1. Ulasan 1 : Kaos disini bagus nyaman
2. Ulasan 2 : Kaos disini biasa saja
3. Ulasan 3 : Kaos disini tidak bagus

TF:

Ulasan 1 : Kaos disini bagus nyaman

Jumlah kalimat : 4

Sehingga perhitungan untuk nilai TF nya menjadi:

TF("Kaos") = ¼ = 0.25

TF("disini") = ¼ = 0.25

TF("bagus") = ¼ = 0.25

TF("nyaman") = ¼ = 0.25

Untuk informasi lebih jelasnya dapatdilihat pada tabel 4.10 tentang penghitungan TF sebagai berikut

Tabel 4. 10Tabel penghitungan TF

	R1	R2	R3	TF1	TF2	F3
Kaos	1	1	1	0.25	0.25	0.25
Disini	1	1	1	0.25	0.25	0.25
Bagus	1	0	1	0.25	0	0.25
Nyaman	1	0	0	0.25	0	0
Biasa	0	1	0	0	0.25	0
Saja	0	1	0	0	0.25	0
Tidak	0	0	1	0	0	0.25

R1, R2, R3 merupakan notasi untuk setiap Ulasan 1, Ulasan2, dan Ulasan 3. Sedangkan TF1, TF2, TF3 merupakan notasi untuk nilai *Term Frequency* setiap ulasan.

IDF (Inverse Document Frequency)

Setelah TF berhasil dihitung, Langkah selanjutnya adalah menghitung IDF, yang merupakan nilai untuk mengukur seberapa penting sebuah kata. IDF akan menilai kata yang sering muncul sebagai kata yang kurang penting berdasarkan kemunculan kata tersebut pada seluruh dokumen. Semakin kecil nilai IDF maka akan dianggap semakin tidak penting kata tersebut, begitu pula sebaliknya

Ulasan 1 : Kaos disini bagus nyaman

Jumlah dokumen : 3

Kaos : $log(\frac{3}{3})$ = 0

Disini : $log(\frac{3}{3})$ = 0

Bagus : $log(\frac{3}{2})$ = 0.18

Nyaman : $log(\frac{3}{1})$ = 0.48

Ulasan 2 : Kaos disini biasa saja

Kaos : $log(\frac{3}{3})$ = 0

Disini : $log(\frac{3}{3})$ = 0

Biasa : $log(\frac{3}{1})$ = 0.48

Saja : $log(\frac{3}{1})$ = 0.48

Ulasan 3 : Kaos disini tidak bagus

Kaos : $log(\frac{3}{3})$ = 0

Disini : $log(\frac{3}{3})$ = 0

Tidak : $log(\frac{3}{1})$ = 0.48

Bagus : $log(\frac{3}{1})$ = 0.48

Setelah penghitungan IDF selesai maka akan masuk ke proses selanjutnya yaitu melengkapi tabel sebelumnya dengan nilai TF-IDF pada seluruh kata seperti pada tabel 4.11 berikut.

Tabel 4. 11TF-IDF

	R1	R2	R3	TF1	TF2	F3	IDF	TFIDF1	TFIDF2	TFIDF3
Kaos	1	1	1	0.25	0.25	0.25	0	0	0	0
Disini	1	1	1	0.25	0.25	0.25	0	0	0	0
Bagus	1	0	1	0.25	0	0.25	0.18	0.045	0	0.045
Nyaman	1	0	0	0.25	0	0	0.48	0,12	0	0
Biasa	0	1	0	0	0.25	0	0.48	0	0,12	0
Saja	0	1	0	0	0.25	0	0.48	0	0,12	0
Tidak	0	0	1	0	0	0.25	0.48	0	0	0,12

Dari tabel 4.11 tersebut, maka hasil akhirnya didapatkan vektor dari setiap ulasan yang dinotasikan oleh TFIDF1, TFIDF2, dan TFIDF3 seperti berikut

- Vektor Review 1 = [0,0,0.045,0.12,0,0,0]
- Vektor review 2 = [0,0,0,0,0.12,0.12,0]
- Vector review 3 = [0,0,0.045,0,0,0.012]

4.4.3.Word2vec

Alat ini bekerja dengan cara mengambil korpus teks sebagai input, lalu menghasilkan representasi vektor dari setiap kata yang ada pada korpus teks tersebut sebagai output. Berikut perintah yang akan dijalankan pada program seperti di bawah ini menggunakan bahasa python.

```
model_w2v = gensim.models.Word2vec(
        tokenized_tweet,
        size=200,
        window=5,
        min_count=2
        sg = 1,
        hs = 0,
        negative = 10,
        workers= 32,
        seed = 34 )
```

Dalam penelitian ini *pre-traned* menggunakan model skipgram dengan menggunakan *gensim models*, cara kerja *Word2vec* ini model *shallow neural network* yang merubah representasi kata yang merupakan kombinasi dari karakter *alphanumeric* menjadi vector. Representasi *vector* tersebut memiliki properti *relationship* terhadap kata-kata yang berkaitan melalui proses *training*. Hasil dari Word2vec untuk kata yang memiliki hubungan *semantic* pada kata "kecil" bisa dilihat pada tabel 4.12.

Tabel 4. 12Representasi semantic dari kata "kecil"

Kata	Vector
Ngepas	0.639337956905365
Ngetat	0.631723165512085
Kecl	0.6266061067581177
Muat	0.6192826628684998
Kurus	0.6184771656990051

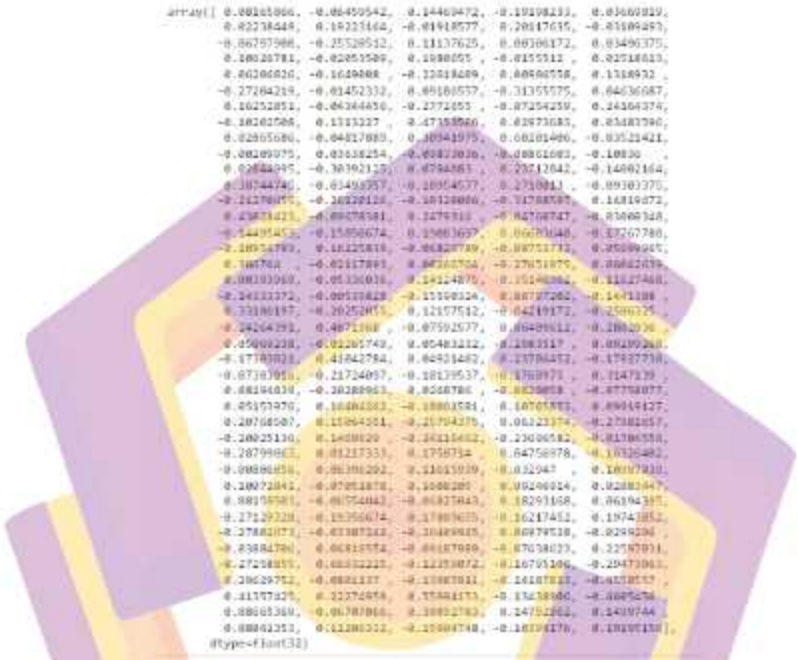
Contoh berikutnya adalah untuk kata kaos memiliki hubungan *semantic* bisa dilihat pada tabel 4.13.

Tabel 4. 13Representasi semantic dari kata "kaos"

Kata	Vector
baju	0.6257702112197876
kuliah	0.5758171081542969
bahus	0.5676507949829102
nyaaaaa	0.5562335848808289
bagados	0.5551339387893677

Dari dua contoh pada tabel 4.12 dan 4.13, dapat dlihat bahwa model *Word2vec* yang dibuat dapat melakukan pekerjaan yang baik untuk menemukan kata yang paling mirip untuk kata tertentu. Hal ini bisa terjadi karena model wod2vec telah mempelajari vektor untuk setiap kata unik dalam data dan menggunakan kesamaan kosinus untuk mengetahui vektor (kata) yang paling mirip. Karena sebelumnya pada model_w2v didefinisikan size = 200 untuk

fitur/variabel independen. Berikut adalah representasi vektor dari kata **kaos** pada korpus yang sudah dibuat data dilihat pada gambar 4.4



Gambar 4.4 Vector dari Kata Kaos

4.4.4.Doc2vec

Model *Doc2vec* adalah algoritma tanpa pengawasan untuk menghasilkan vektor untuk kalimat/paragraf/dokumen. Pendekatan ini merupakan perpanjangan dari *Word2vec*. Perbedaan utama antara keduanya adalah bahwa *Doc2vec* menyediakan konteks tambahan yang unik untuk setiap dokumen dalam korpus. Konteks tambahan ini tidak lain adalah vektor fitur lain untuk keseluruhan dokumen. Vektor dokumen ini dilatih bersama dengan kata *vector*. Dalam proses

Doc2vec langkah pertama adalah memuat memuat perpustakaan yang diperlukan, berikut adalah *source* untuk *library* yang nantinya akan digunakan.

```
from tqdm import tqdm
tqdm.pandas(desc="progress-bar")
from gensim.models.Doc2vec import TaggedDocument
```

Untuk mengimplementasikan *Doc2vec*, yang harus dilakukan adalah memberi label atau menandai setiap ulasan yang diberi token dengan ID unik. Pada tahap ini dapat dilakukan menggunakan Gensim's LabeledSentence () function.

```
def add_label(twt):
  output = []
  for i, s in zip(twt.index, twt):
    output.append(TaggedDocument(s, ["tweet_" + str(i)]))
  return output
labeled_tweets = add_label(tokenized_tweet)
```

Setelah data diberi token dengan id unik, maka selanjuntya adalah membuat model *Doc2vec* sebagai berikut,

```
model_d2v = gensim.models.Doc2vec(dm=1,
dm_mean=1,
vector_size=200,
window=5,
negative=7,
min_count=5,
workers=32,
alpha=0.1, seed = 23,)
```

Representasi *vector* tersebut memiliki properti relationship terhadap kata-kata yang berkaitan melalui proses training. Hasil dari *Doc2vec* hampir sama

dengan kedekatan *Word2vec* untuk kata yang memiliki hubungan *semantic* pada kata "ongkir" maka untuk dalam pengaplikasiannya untuk mengecek kedekatan kata "ongkir" seperti berikut

```
model_d2v.wv.most_similar(positive="ongkir")
```

Maka hasil dari kode tersebut bisa dilihat pada tabel 4.14.

Tabel 4. 14Representasi semantic dari kata "ongkir"

Kata	Vector
Gratis	0.5490841865539551
Kepala	0.4832165539264679
Turun	0.4782751798629761
Mesti	0.4270642399787903
Flashsale	0.3491603136062622

Dari tabel 4.14, diketahui bahwa kata "ongkir: memiliki nilai kedekatan dengan kata gratis sebesar 0.549 diikuti dengan kata kepala 0.4832, turun 0.4782 , mesti 0.427 dan flashsale 0.3492.

4.5 Klasifikasi

Tahapan berikutnya adalah langkah persiapan skenario untuk tahapan pelatihan dengan XGBoost, pembuatan skenario dilakukan sesuai degan jenis penelitian yang bersifat eksperimental dimana penulis akan mencoba bereksperimen menggunakan algoritma XGBoost dengan 4 metode fitur ekstrasi yaitu Bag of words, TF-IDF, Word2vec dan Doc2vec.

4.5.1. Tahap Pelatihan XGBoost

Extreme Gradient Boosting (XGBoost) adalah implementasi lanjutan dari algoritma peningkatan *gradient*. Ini memiliki pemecah model *linier* dan algoritma pembelajaran pohon. Kemampuannya untuk melakukan komputasi paralel pada satu mesin membuatnya sangat cepat. Ini juga memiliki fitur tambahan untuk melakukan validasi silang dan menemukan variabel penting. Ada banyak parameter yang perlu dikontrol untuk mengoptimalkan model. Beberapa manfaat utama XGBoost adalah:

- Regularisasi - membantu mengurangi *overfitting*
- Pemrosesan Paralel - XGBoost mengimplementasikan pemrosesan paralel dan jauh lebih cepat dibandingkan dengan GBM.
- Menangani Nilai yang hilang - Ini memiliki rutinitas bawaan untuk menangani nilai yang hilang.
- *Built-in Cross-Validation* - memungkinkan pengguna untuk menjalankan *cross-validation* pada setiap iterasi dari proses boosting.

Pada penelitian ini metode XGBoost diolah menggunakan bahasa python sebagai berikut.

```
xgb_model = XGBClassifier(max_depth=6, n_estimators=1000)
```

Pada tahap pengklasifikasian menggunakan XGBoost dan dalam penerapannya untuk lebih mudahkan diberikan sampling, data tersebut diinisialisasi menjadi data [X,Y] seperti yang diuraikan pada tabel 4.15

Tabel 4. 15Data Membangun Pohon XGBoost

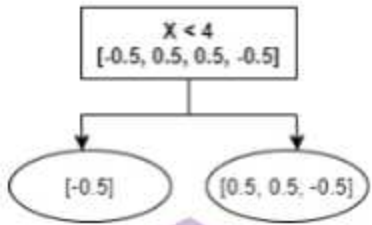
X	Y
2	0
6	1
10	1
12	0

Langkah pertama adalah melakukan sebuah prediksi awal. Pada penelitian ini nilai parameter *base_score* atau prediksi awal f0(x) adalah 0.5. kemudian langkah selanjutnya adalah melakukan proses perhitungan untuk mencari nilai error atau residuals. Hasil proses perhitungan ditunjukkan pada tabel 4.16

Tabel 4. 16Perhitungan Error ke-1

X	Y	$f_0(x)$	$Y = y\text{-}f_0(x)$
2	0	0,5	-0,5
6	1	0,5	0,5
10	1	0,5	0,5
12	0	0,5	-0,5

Pada model training atau latih, untuk mencegah split, pohon yang mau dibangun akan dibagi menjadi beberapa bagian, dapat dilihat pada gambar 4.5. Cara perhitungan ditampilkan pada gambar 4.5 didapat pada perhitungan tabel 4.16

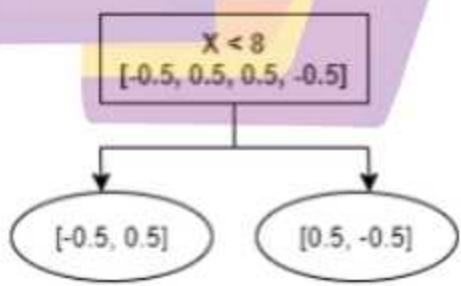


Gambar 4.5. contoh membangun pohon XGBoost ke 1

Proses selanjutnya atau yang kedua pada gambar 4.6 dan didapatkan penghitungan dari tabel 4.17

Tabel 4. 17Perhitungan Error ke-2

X	Y	$f_0(x)$	$\hat{Y} = y - f_0(x)$
2	0	0,5	-0,5
6	1	0,5	0,5
10	1	0,5	0,5
12	0	0,5	-0,5

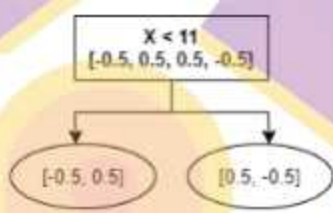


Gambar 4.6. contoh membangun pohon XGBoost ke 2

Selanjutnya pada bagian yang ketiga pada gambar 4.6 didapatkan penghitungan dari tabel 4.18

Tabel 4. 18Perhitungan Error ke-3

X	Y	$f_0(x)$	$\dot{Y} = y\text{-}f_0(x)$
2	0	0,5	-0,5
6	1	0,5	0,5
10	1	0,5	0,5
12	0	0,5	-0,5



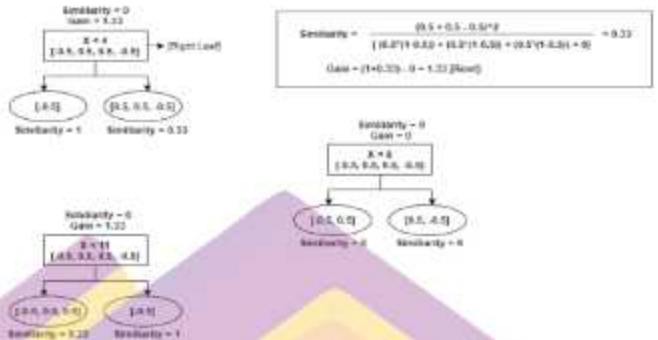
Gambar 4.7. contoh membangun pohon XGBoost ke 3

Langkah selanjutnya adalah tahapan menentukan batasan nilai *root* dengan menghitung nilai rata-rata antara 2 titik split dengan menerapkan persamaan,

$$Similiarity\ Score = \frac{(\sum \dot{Y}i)2}{\sum(\text{Previous } fi(x).(1-\text{Previous } fi(x))+\bar{e}} \quad (4.1)$$

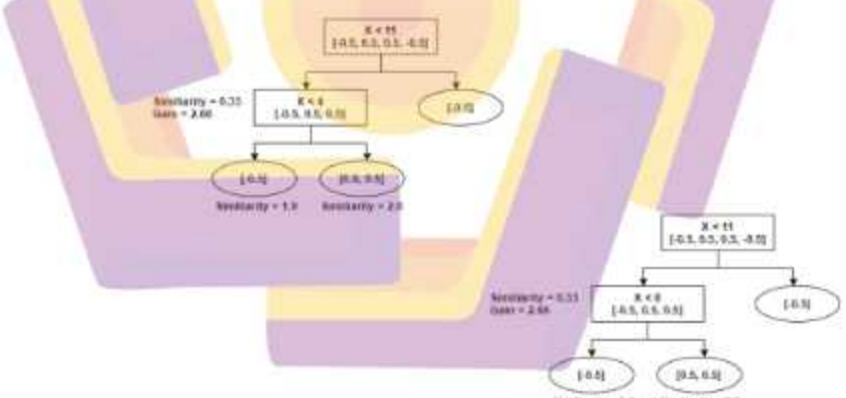
$$Gain = \left(Left_{similarity} \pm Right_{similarity}\right) - Root_{similarity} \quad (4.2)$$

untuk proses penghitunganya akan ditunjukan pada gambar 4.8.



Gambar 4.8 Perhitungan nilai similiarity dan gain

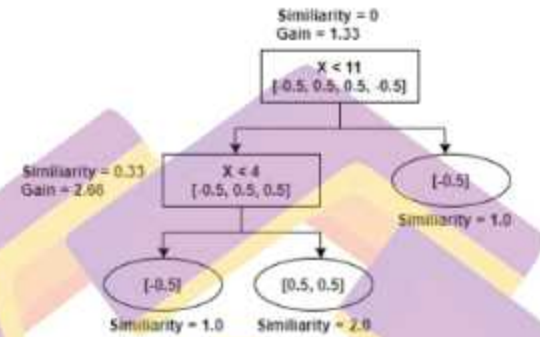
Lakukan pemisahan untuk pohon yang memiliki nilai gain maksimal dan akan dilakukan split yang ditunjukkan di Gambar 4.9



Gambar 4.9. Perhitungan nilai *similiarity* dan gain pada saat *split*

Dilanjutkan dengan melakukan pemangkasan untuk memperkecil ukuran pohon. Berdasarkan gambar 4.9 dapat disimpulkan internal root x<4 menjadi

konstruksi pohon karena memiliki nilai gain yang maksimal. Proses pemangkasan ditunjukkan pada gambar 4.10

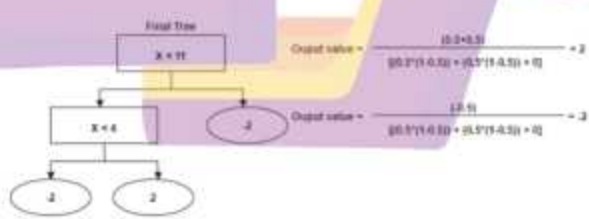


Gambar 4.10. Proses Pemangkasan

Lakukan perhitungan untuk mencari nilai output untuk mendapatkan model pohon dengan menggunakan persamaan (4.3)

$$Output\ Value = \frac{(\sum Yi)}{\sum[\text{Previous}\ \ pi(x).(1-\text{Previous}\ \ pi(x)]+\epsilon} \quad (4.3)$$

Proses perhitungan output ditampilkan di Gambar 4.10.

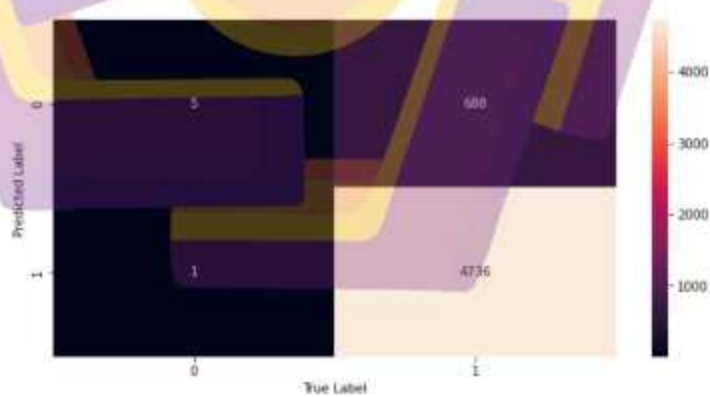


Gambar 4.11 output model XGBoost

Lakukan kembali perulangan pada langkah ketiga untuk mendapatkan hasil terbaik pada pohon XGBoost.

4.6 Confusion Matrix

Proses perhitungan kinerja model ini sebagai ukuran seberapa baik sebuah model dalam mengklasifikasi sebuah teks. Pada penelitian ini perhitungan kinerja model akan menggunakan metode *confusion matrix*. Metode ini akan mencari nilai persentase dari akurasi, presisi, *recall* dan *f1score* dari sebuah model arsitektur. Agar bisa lebih mengerti dengan perhitungan ini, penulis akan memberikan sebuah contoh perhitungan *confusion matrix*. Pada contoh ini, penulis akan menggunakan jumlah kelas yang sesuai dengan kelas pada penelitian yaitu 2 kelas. Untuk lebih lengkapnya bisa kita lihat pada Gambar 4.12



Gambar 4.12 Confusion Matrix

Pada gambar 4.12 dapat diketahui untuk penentuan nilai pada confusion matrix terdapat *True Positive* (TP) merupakan data yang diprediksi dengan tepat

sebagai keluaran positive atau benar.*True Negative* (TN) merupakan data yang diprediksi tepat sebagai keluaran negative atau salah. *False Positive* (FP) merupakan data prediksi dengan kurang tepat apabila keluaran berupa positif atau benar. Yang terakhir adalah *False Negative* (FN) merupakan data yang diprediksi kurang tepat. Berikut perhitungan dari confusion matrix:

TP = 4736

FP = 1

FN = 688

TN = 5

1. Akurasi

   Pada proses pencarian nilai akurasi, rumus yang digunakan adalah

   TP + TN / (TP + TN + FP + FN)

   4736 + 5 / (4736 + 5 + 1 + 688)

   4741 / 5430 = 0,873112 atau 87.3 %

2. Presisi

   Pada proses pencarian nilai presisi, rumus yang digunakan adalah

   TP / (TP + FP)

   4736 / (4736 + 1)

   96 / 98 = 0.999789 atau 100 %

3. Recall

   Pada proses pencarian nilai recall, rumus yang digunakan adalah

   TP / (TP + FN)

   4736 / (4736 + 688)

4736 / 5424 = 0.873156 atau 87 %

4. F1 Pada proses pencarian nilai F1, rumus yang digunakan adalah

$$F1 - score = 2\frac{Precision * Recal}{recision + Recall}$$

2 (0.999789*0.873156)/(0.999789+0.873156)= 0.932192 atau 93%

4.7 Analisa dan Pembahasan

Selanjutnya adalah tahap evaluasi pengujian pembahasan dari ulasan yang sudah diklasifikasikan oleh model.

4.7.1 Analisa Hasil

Untuk hasil pengujian akan mengacu pada nilai *F1-Score*, penulis menggunakan parameter *F1-Score* dibandingkan *accuracy* karena *accuracy* bagus digunakan jika distribusi klasifikasi positif dan negatif seimbang (jumlah nyata antara positif dan negatif berimbang), sedangkan *F1-score* lebih baik digunakan ketika distribusi klasifikasi positif dan negatif tidak seimbang. Akan dilakukan skenario empat pengujian yaitu dengan komposisi perbandingan data latih dan uji 80:20 pada tabel 4.19

Tabel 4. 19Perbandingan Kinerja Model

	Accuracy	Prescision	Recall	F1-Score
Bag of Words	0.876978	0.8960483	0.971712	0.932348
TF-IDF	0.878085	0.897405	0.9712898	0.932887
Word2vec	0.880663	0.897685	0.974245	0.934399
Doc2vec	0.878268	0.887305	0.985644	0.9338933

Pada Tabel 4.19 bisa dilihat hasil percobaan dari semua skenario terhadap 4 model arsitektur yang ada. Pada hasil ini didapatkan untuk metode *Bag of Words* menghasilkan *accuracy* 0.876, nilai *prescision* 0.896, *recall* 0.972 dan *F1 Score* 0.932. Metode *Bag of word* ini sebenarnya cukup baik dalam mengekstrak bentuk kata menjadi *vector* karena konsepnya yang menghitung frekuensi kemunculan kata pada dokumen, namun karena metode ini hanya mengikuti jumlah *word*(kata) unik dari seluruh dokumen. Artinya, jika nanti terdapat kata yang beragam macam bentuknya dan terdapat berbagai kata unik baru maka ukuran korpus juga akan semakin membesar sehingga hal ini akan berpengaruh pada komputasi yang dibutuhkan pada saat dilakukan pelatihan model pembelajaran mesin khususnya pada klasifikasi teks.

Selanjutnya adalah skenario pengujian untuk metode TF-IDF menghasilkan *accuracy* 0.878, *prescision* 0.897, *recall* 0.971 dan *F1 Score* 0.932. Metode TF-IDF ini memiliki kelebihan sebagaimana proses fitur ekstrasi lainya, hanya saja perbedaan dengan *bag of word*, metode ini cukup baik dalam dalam memberikan frekuensi kemunculan kata pada sebuah dokumen, menurut (Flores & Jasa, 2020) TF-IDF menganggap semakin sedikit tingkat frekuensi yang muncul maka kata itu unik dan penting, pada penelitian yang dilakukan metode TF-IDF ini memiliki peforma lebih tinggi daripada metode *Bag of Word* baik dari segi *accuracy*, *prescision*, *recall* dan *F1-Score*, akan tetapi hal yang perlu diperhatikan dari metode TF-IDF adalah kelemahanya pada saat mendeteksi keterkaitan kata atau tidak bisa menangkap posisi teks dan semantiknya.

Sehingga pada skenario percobaan ketiga menggunakan *Word2vec* ini memiliki performa yang lebih baik dibandingkan dengan TF-IDF, pada penelitian ini juga membuktikan bahwa metode *Word2vec* memiliki kinerja yang baik dalam menentukan relasi *semantic* antar kata. Untuk lebih jelasnya bagaimana Word2vec menentukan semantic antar kata, misal terdapat kata yang memiliki hubungan *semantic* pada kata "kecil" bisa dilihat pada tabel 4.20.

Tabel 4. 20 Representasi semantic Word2vec

Kata	vektor
Ngepas	0.639337956905365
Ngetat	0.631723165512085
Kecl	0.6266061067581177
Muat	0.6192826628684998
Kurus	0.6184771656990051

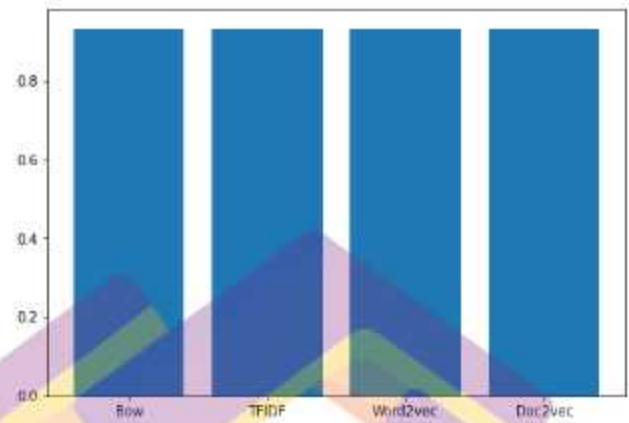
Dari tabel 4.20 dapat dlihat bahwa model Word2vec yang dibuat dapat melakukan pekerjaan yang baik untuk menemukan kata yang paling mirip untuk kata tertentu. Hal ini bisa terjadi karena model wod2vec telah mempelajari vektor untuk setiap kata unik dalam data dan menggunakan kesamaan kosinus untuk mengetahui vektor (kata) yang paling mirip. Tahap pertama dari proses *Word2vec* adalah membangun kosakata dari data teks pelatihan dan mempelajari representasi vektor dari urutan kata(Kurniawan & Maharani, 2020), pada proses pengujiannya metode *Word2vec* memiliki nilai *accuracy* 0.881, *prescision* 0.898, *recall* 0.974 dan *F1 Score* 0.934. Model *Skip-gram* dalam *Word2vec* membutuhkan proses komputasi yang begitu besar hal ini dikarenakan semua bobot dalam *embedding*

matrix harus di *update* untuk setiap *word* target, konteks kata atau pasangan kata, hal ini tentunya menjadi tidak praktis jika kosa katanya banyak. Sehingga pada model *Word2vec* diberikan metode evaluasi *negative sampling*. Pengambilan sampel negatif hanya memungkinkan sebagian kecil dari bobot berubah selama pelatihan (Af'idah et al., 2021). Sejumlah kecil kata negatif, atau kata-kata yang tidak diharapkan muncul dalam konteks kata target, dipilih dan hanya bobot negatifnya yang diperbarui. Meskipun metode *Word2vec* memiliki performa lebih tinggi dibanding metode *Bag-of-Words* dan TFIDF namun untuk hasil ini belum bisa dikatakan maksimal karena *Word2vec* tidak mampu merepresentasikan vektor dari kata yang tidak ada dalam korpus (Nurdin, Anggo Seno Aji, et al., 2020).

Pengujian yang keempat yang terakhir adalah menggunakan fitur *Doc2vec*, metode ini merupakan pengembangan dari *Word2vec*. *Vector* yang *Doc2vec* dapat digunakan untuk beberapa tujuan, seperti menemukan kesamaan antar kalimat/paragraph. Pada skenario pengujian untuk metode *Doc2vec* menghasilkan nilai *accuracy* 0.878, *prescision* 0.887, *recall* 0.985 dan *F1-Score* 0.933. Dalam peningkatannya metode *Doc2vec* ini masih bisa optimal jika didukung oleh jumlah kosakata yang beragam dengan cara mengubah *vector_size* dari 100 menjadi sesuatu yang lebih kecil atau lebih besar, dikarenakan *dataset* pada penelitian ini hanya *focus* ke respon ulasan masyarakat terhadap produk *ecommerce* belum terlalu kompleks ke masalah pelayanan, kurir, jasa kirim sehingga dalam pengaplikasiannya ke metode *Doc2vec* masih terbatas.

Dari semua hasil percobaan yang dilakukan pada setiap skenario dapat diketahui bahwa semua metode mendapatkan hasil tinggi dimana hasilnya lebih dari 0.90 untuk nilai F1 Scorenya, dimana tentunya banyak hal yang mempengaruhi hasil yang maksimal pada setiap skenario. Secara umum dari keempat model fitur ekstrasi, kombinasi metode XGBoost yang menggunakan Word2vec memiliki nilai F1-Score lebih baik dibanding ketiga metode lainnya.

Menurut (Nurdin, Anggo Seno Aji, et al., 2020) metode ini memang dikembangkan menggunakan neural network yang terdiri dari sebuah *hidden layer* dan *fully con_nected layer* yang telah terlatih dalam melakukan klasifikasi pada banyak teks dan teruji pada banyak penelitian sebelumnya termasuk pada klasifikasi teks. Jumlah data tentunya juga berpengaruh dimana semakin banyak *dataset* yang digunakan maka dapat menangkap kemiripan makna kata dengan baik. Memang dalam hal ini *Word2vec* secara efektif menangkap hubungan semantik antar kata namun jika ada kondisi yang membutuhkan hubungan antar kalimat dan dokumen. Misalnya, jika ada kondisi untuk mengetahui apakah dua ulasan produk adalah duplikat satu sama lain tentunya *Doc2vec* bisa lebih optimal dengan menyesuaikan komposisi kumpulan datanya. Selanjutnya untuk uji coba empat metode lebih jelasnya akan disediakan dalam bentuk grafik pada gambar 4.13.



Gambar 4.13 Grafik F1-Score Perbandingan Empat Metode

Pada kasus penelitian ini untuk uji coba menggunakan empat metode(Bag-of-Words, TF-IDF, Word2vec dan Doc2vec) tidak terlalu signifikan pada XGBoost, hal ini dikarenakan setiap metode menghasilkan nilai F1-Score yang tidak terlalu jauh untuk jarak perbedaan.

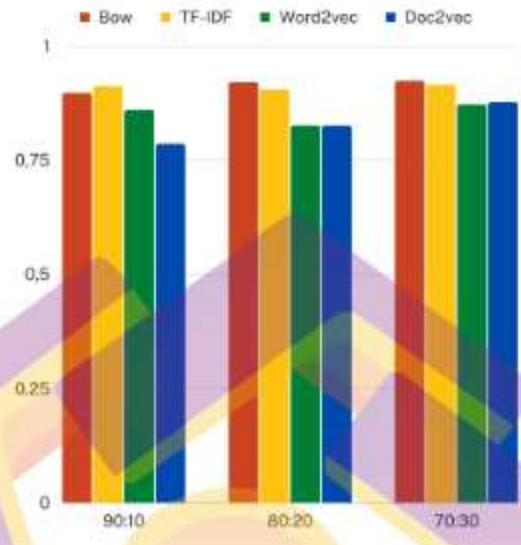
Temuan selanjutnya adalah perbedaan empat metode ketika memproses dari segi jumlah datanya. Word2vec dan Doc2vec pada penelitian yang sudah dilakukan memiliki performa yang cukup baik dibanding metode tradisional TF-IDF dan Bag of Words saat memproses kata menjadi bentuk vector pada dataset yang besar yang memiliki kelas tidak seimbang, akan tetapi ketika mengolah data yang tidak terlalu banyak dan cenderung memiliki kelas *balance* TF-IDF dan Bag of Words cenderung lebih baik. Hal ini dibuktikan dengan penelitian ini yang mencoba menguji keempat metode fitur ekstrasi berdasarkan penelitian yang telah dilakukan sebelumnya oleh(Efrizoni et al., 2022) data ulasan produk sejumlah 1000 data dengan komposisi pembagi 90:10, 80:20, dan 70:30 dengan tahapan

yang sama pada alur penelitian. Hasil dari uji coba ditunjukan pada tabel 4.21 sebagai berikut.

Tabel 4. 21Perbandingan Metode Komposisi 1000 Data

90:10	Accuracy	Prescision	Recall	F1-Score
Bag of Words	0.896	0.932	0.885	0.908
TF-IDF	0.911	0.965	0.878	0.921
Word2vec	0.859	0.883	0.872	0.878
Doc2vec	0.785	0.792	0.853	0.822
80:20	Accuracy	Prescision	Recall	F1-Score
Bag of Words	0.920	0.956	0.910	0.933
TF-IDF	0.904	0.910	0.910	0.920
Word2vec	0.825	0.861	0.849	0.855
Doc2vec	0.825	0.856	0.856	0.856
70:30	Accuracy	Prescision	Recall	F1-Score
Bag of Words	0.923	0.967	0.934	0.945
TF-IDF	0.914	0.953	0.928	0.940
Word2vec	0.871	0.931	0.889	0.901
Doc2vec	0.876	0.914	0.913	0.914

Pada tabel 4.21 bisa dilihat hasil percobaan dari semua skenario terhadap 4 model arsitektur yang diujicobakan dengan komposisi 1000 data didapatkan bahwa metode *Bag of Words* dan TF-IDF menghasilkan peforma lebih tinggi dari segi *accuracy* dibandingkan dengan *Word2vec* dan *Doc2vec*. Untuk lebih jelasnya akan disajikan dalam bentuk grafik tingkat akurasi yang dapat dilihat pada gambar 4.14



Gambar 4.14 Grafik peningkatan Akurasi Komposisi 1000 data

Dari gambar 4.14 dapat diketahui dari untuk komposisi 1000 data baik dari ujicoba pembagian 90:10, 80:20 dan 70:30 metode *Bag of Words* dan TF-IDF memiliki nilai akurasi lebih baik dibanding *Word2vec* maupun *Doc2vec*. Dengan demikian empat metode yang digunakan pada penelitian ini memiliki keunggulan dan kekuranganya masing masing. Jika dihadapkan dengan data yang relative sedikit maka fitur TF-IDF dan *Bag of Word* lebih baik karena keunggulan dari metode tradisional ini menganggap semakin sedikit tingkat frekuensi yang muncul maka kata itu unik dan penting(Flores & Jasa, 2020), untuk *Word2vec* dan *Doc2vec* lebih *powerfull* ketika memproses data yang lebih besar hal ini dikarenakan jumlah *dataset* yang sedikit *Word2vec* tidak dapat menangkap kemiripan makna kata dengan baik(Efrizoni et al., 2022).

Hasil lain yang dicapai dalam penelitian yang dilakukan adalah terletak pada proses *preprocessing* dan *cleaning* data dimana model yang telah diuji coba mampu meminimalisir kerusakan data seperti *missing value*, maupun format data yang tidak sesuai dengan *system* bisa diatasi. Pada tahap *preprocessing* khususnya fitur normalisasi mampu mengoreksi kata *slangword* menjadi bahasa baku sehingga dapat diolah oleh model secara optimal. Pada tahapan *cleaning* juga mampu menghapus data redundan sehingga mampu meminimalisir resiko terjadinya *inkosistensi* sehingga akan membantu model Word2vec dalam memberikan rekomendasi kata terdekat yang sesuai dengan kesamaan penggunaan, kedekatan konteks, maupun relasi antar kata-kata. Penelitian ini juga membuktikan bahwa metode klasifikasi XGBoost dengan fitur Bag of Word, TF-IDF, Word2vec maupun Doc2vec dapat digunakan dalam menganalisis ulasan produk berbahasa Indonesia di *ecommerce* yang memiliki kelas data tidak seimbang.

4.7.2 Perbandingan dengan penelitian Terdahulu

Kombinasi TFIDF+XGBoost pada penelitian ini tentunya menghasilkan nilai F1-Score yang tinggi dibandingkan penelitian sebelumnya yang dilakukan oleh (Akter et al., 2021) pada saat mengklasifikasikan *imbalance data* , dalam proses pelabelan data baik dari penelitian terdahulu dengan yang saat ini dilakukan menggunakan skema yang sama yaitu dibawah rating tiga akan diberi label buruk dan diatas tiga akan dibeli label baik. Untuk tabel perbandingannya dapat dilihat pada tabel 4.22 sebagai berikut

Tabel 4. 22Perbandingan TF-IDF+XGBoost

Dataset	Preprocessing	Model	Hasil
Bahasa Indonesia, 22.624 ulasan	Case Folding, Remove Punctuation, Stopword Removal, word normalize, Stemming, Tokenization	TF-IDF + XGBoost, Split Data 80:20	0.93
Bahasa Bengali, 7905 ulasan	Remove punctuation, tokenization, stopword removal dan stemming	TF-IDF + XGBoost, Split Data 80:20	0.91

Dari tabel 4.22diketahui bahwa penelitian yang dilakukan meghasilkan F1 Score lebih tinggi 0.93 dibanding penelitian sebelumnya yang dilakukan oleh (Akter et al., 2021) dengan metode yang sama hanya menghasilkan nilai 0.91. Penelitian terdahulu menggunakan dataset Bengali sedangkan penelitian yang dilakukan menggunakan dataset berbahasa Indonesia, meski ada perbedaan pada dataset akan tetapi pada situasi saat ini mencoba mengoptimalisasi pada tahapan preprocessing dimana pada tahap ini menggunakan fitur *normalizer* kata, fitur ini berfungsi mengubah kata menjadi kata baku sehingga bisa melewati proses stemming dengan optimal. Case folding pada tahapan ini berfungsi untuk menyemaratakan huruf antar kata, misal kata "KAos" dan "kaos" jika memasuki proses case folding maka akan dihitung sama oleh fitur TF-IDF, pada penelitian (Akter et al., 2021) tidak menggunakan case folding dan normalize kata sehingga kata yang seharusnya memiliki arti sama maka bisa jadi dianggap berbeda. Kompleksitas dan keberagaman data juga berpengaruh dimana pada penelitihan

terdahulu hanya menggunakan 7905 ulasan sedangkan di penelitian saat ini menggunakan 22.624 ulasan.

BAB V
KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilaksanakan terhadap 4 fitur ekstrasi yang berbeda untuk mengklasifikasikan data teks produk dapat disimpulkan:

1. Berdasarkan jumlah dataset yang sudah dikumpulkan menggunakan teknik scrapping data, respon masyarakat terhadap produk local Indonesia dapat diketahui menggunakan worcloud bahwa mereka lebih sering memberikan ulasan tentang topic yang berkaitan dengan kata bagus dengan frekuensi kemunculan 4081, sesuai 3967, bahan 2413, barang 1869 dan kirim 1788.
2. Pada hasil uji coba didapatkan metode *Bag of Words* menghasilkan *accuracy* 0.876, nilai *prescision* 0.896, *recall* 0.972 dan *F1 Score* 0.932. Selanjuntya untuk metode TFIDF menghasilkan *accuracy* 0.878, *prescision* 0.897, *recall* 0.971 dan *F1 Score* 0.932. Metode Word2vec menghasilkan *accuracy* 0.881, *prescision* 0.898, *recall* 0.974 dan *F1 Score* 0.934 dan untuk metode Doc2vec menghasilkan *accuracy* 0.878, *prescision* 0.887, *recall* 0.985 dan *F1-Score* 0.933.
3. *Feature extraction* yang menghasilkan peforma terbaik pada algoritma klasifikasi XGBoost adalah Word2vec ketika memproses 22.624 data akan tetapi jika dihadapkan dengan data yang relative sedikit yaitu

sejumlah 1000 data dengan komposisi *balance* maka fitur TF-IDF dan Bag of Word mempnyai peforma lebih tinggi.

5.2 Saran

Berikut ini adalah beberapa saran yang dapat dijadikan pedoman untuk melakukan pengembangan penelitian ini, antara lain:

1. Untuk memaksimalkan peforma Doc2vec jika ingin meneliti permasalahan yang berkaitan dengan ulasan produk pada penelitian yang akan datang diperluhkan dataset yang lebih beragam agar hal ini dikarenakan metode Doc2vec ini masih bisa optimal jika didukung oleh jumlah kosakata yang beragam dengan cara mengubah vector_size dari 100 menjadi sesuatu yang lebih kecil atau lebih besar.
2. Meskipun metode Word2vec menghasilkan nilai F1 Score lebih dari 0.90, namun metode ini memiliki kekurangan tidak mampu merepresentasikan vektor dari kata yang tidak ada dalam korpus (out of vocabulary) sehingga diperluhkan metode fitur ekstrasi yang dapat mengatasi permasalahan tersebut.

DAFTAR PUSTAKA

PUSTAKA BUKU

Bhattacharjee, J. (2018). *FastText Quick Start Guide*. Packt Publishing.

Kotu, V., & Deshpande, B. (2019). Chapter 8 - Model Evaluation. In V. Kotu & B. Deshpande (Eds.), *Data Science (Second Edition)* (Second Edition, pp. 263–279). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-12-814761-0.00008-3

Gorunescu, F. (2010). Data Mining Concepts, Models and Techiniques. Springer International Publishing. https://doi.org/10.1007/978-3-642-19721-5

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Af'idah, D. I., Dairoh, Handayani, S. F., & Pratiwi, R. W. (2021). Pengaruh Parameter Word2vec terhadap Performa Deep Learning pada Klasifikasi Sentimen. *Jurnal Informatika: Jurunal Pengembangan IT (JPIT)*, *6*(3), 156–161.

Afifah, K., Yulita, I. N., & Sarathan, I. (2021). Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier. *2021 International Conference on Artificial Intelligence and Big Data Analytics*, 22–27. https://doi.org/10.1109/ICAIBDA53487.2021.9689735

Agustiningsih, K. K., Utami, E., Muhammad, O., & Alsyaibani, A. (2022). *Sentiment Analysis of COVID-19 Vaccines in Indonesia on Twitter Using Pre-Trained and Self-Training Word Embeddings*. *1*, 39–46.

Akter, M. T., Begum, M., & Mustafa, R. (2021). Bengali Sentiment Analysis of E-commerce Product Reviews using K-Nearest Neighbors. *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, 40–44. https://doi.org/10.1109/ICICT4SD50815.2021.9396910

Amien, S., Perdana, P., Aji, T. B., & Ferdiana, R. (2021). *Aspect Category Classification dengan Pendekatan Machine Learning Menggunakan Dataset Bahasa Indonesia (Aspect Category Classification with Machine Learning Approach Using Indonesian Language Dataset)*. *10*(3), 229–235.

Amin, S., Uddin, M. I., AlSaeed, D., & Khan, A. (2021). Early Detection of Seasonal Outbreaks from Twitter Data Using Machine Learning Approaches. *Complexity*, *2021*, 1–12. https://doi.org/10.1155/2021/5520366

Ay, B., Talo, M., Hallac, I., & Aydin, G. (2018). Evaluating deep learning models for sentiment classification. *Concurrency and Computation: Practice and Experience*, *30*. https://doi.org/10.1002/cpe.4783

Basani, Y., Sibuea, H. V, Sianipar, S. I. P., & Samosir, J. P. (2019). Application of Sentiment Analysis on Product Review E-Commerce. *Journal of Physics: Conference Series*, *1175*, 12103. https://doi.org/10.1088/1742-6596/1175/1/012103

Bhattacharjee, J. (2018). *FastText Quick Start Guide*. Packt Publishing.

Bhoi, A., & Joshi, S. (2018). Various Approaches to Aspect-based Sentiment Analysis. *CoRR*, *abs/1805.0*.

Bi, J.-W., Liu, Y., & Fan, Z.-P. (2019). Representing sentiment analysis results of online reviews using interval type-2 fuzzy numbers and its application to product ranking. *Information Sciences*, *504*, 293–307. https://doi.org/https://doi.org/10.1016/j.ins.2019.07.025

Edy, S., Imam, R., Reza, F., & Eriszana. (2021). *SENTIMENT EMBEDDINGS DOC2VEC PADA KLASIFIKASI KELUHAN POLUSI UDARA*. *9*(1), 1–7.

Efrizoni, L., Defit, S., Tajuddin, M., & Anggrawan, A. (2022). *Komparasi Ekstraksi Fitur dalam Klasifikasi Teks Multilabel Menggunakan Algoritma Machine Learning Comparison of Feature Extraction in Multilabel Text Classification Using Machine Learning Algorithm*. *21*(3). https://doi.org/10.30812/matrik.v21i3.1851

Eligüzel, N., Çetinkaya, C., & Dereli, T. (2022). Comparative analysis with topic modeling and word embedding methods after the Aegean Sea earthquake on Twitter. *Evolving Systems*. https://doi.org/10.1007/s12530-022-09450-4

Farzindar, A. A., & Inkpen, D. (2020). *Natural Language Processing for Social Media (G. Hirst (ed.); Third Edit)* (Morgan & Claypoo (ed.)).

Flores, V. A., & Jasa, L. (2020). *Analisis Sentimen untuk Mengetahui Kelemahan dan Kelebihan Pesaing Bisnis Rumah Makan Berdasarkan Komentar Positif dan Negatif di Instagram*. *19*(1). https://doi.org/10.24843/MITE.2020.v19i01.P07

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive Logistic Regression: A Statistical View of Boosting. *The Annals of Statistics*, *28*, 337–407. https://doi.org/10.1214/aos/1016218223

Hakim, B. (2021). ANALISA SENTIMEN DATA TEXT PREPROCESSING PADA DATA MINING DENGAN MENGGUNAKAN MACHINE LEARNING DATA TEXT PRE-PROCESSING SENTIMENT ANALYSIS IN DATA MINING USING MACHINE LEARNING School of Computer Science and Technology , Harbin Institute of Technology. *Journal of Business and Audit Information Systems*, *4*(2), 16–22. https://doi.org/DOI: http://dx.doi.org/10.30813/jbase.v4i2.3000

Hidayatullah, A. F., Cahyaningtyas, S., & Hakim, A. (2021). Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset. *IOP Conference Series: Materials Science and Engineering*, *1077*, 12001. https://doi.org/10.1088/1757-899X/1077/1/012001

Ishihara, S. (2021). Score-based likelihood ratios for linguistic text evidence with a bag-of-words model. *Forensic Science International*, *327*, 110980. https://doi.org/https://doi.org/10.1016/j.forsciint.2021.110980

Jaya Hidayat, T. H., Ruldeviyani, Y., Aditama, A. R., Madya, G. R., Nugraha, A. W., & Adisaputra, M. W. (2022). Sentiment analysis of twitter data related to Rinca Island development using Doc2vec and SVM and logistic regression as classifier. *Procedia Computer Science*, *197*, 660–667. https://doi.org/https://doi.org/10.1016/j.procs.2021.12.187

JAYADI, S. F. N. H. R. (2022). *Sentiment Analysis Of Indonesian E-Commerce Product Reviews Using Support Vector Machine Based Term Frequency Inverse Document*. *99*(17), 4316–4325.

Jurafsky, D., & Martin, J. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Vol. 2).

Kevin, V., Que, S., Iriani, A., & Purnomo, H. D. (2020). Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization (Online Transportation Sentiment Analysis Using Support Vector Machine Based on Particle Swarm Optimization). *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *9*(2), 162–170.

Khattak, F. K., Jeblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., & Rudzicz, F. (2019). A survey of word embeddings for clinical text. *Journal of Biomedical Informatics: X*, *4*(December), 100057. https://doi.org/10.1016/j.yjbinx.2019.100057

Kotu, V., & Deshpande, B. (2019). Chapter 8 - Model Evaluation. In V. Kotu & B. Deshpande (Eds.), *Data Science (Second Edition)* (Second Edi, pp. 263–279). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-12-814761-0.00008-3

Kowsari, K., Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text Classification Algorithms: A Survey. *Inf.*, *10*, 150.

Kurniawan, F. W., & Maharani, W. (2020). *Analisis Sentimen Twitter Bahasa Indonesia dengan Word2vec*. *7*(2), 7821–7829.

Kusumaningrum, R., Nisa, I., Nawangsari, R., & Wibowo, A. (2021). Sentiment analysis of Indonesian hotel reviews: from classical machine learning to deep learning. *International Journal of Advances in Intelligent Informatics*, *7*, 292. https://doi.org/10.26555/ijain.v7i3.737

Mo, H., Sun, H., Liu, J., & Wei, S. (2019). Developing window behavior models for residential buildings using XGBoost algorithm. *Energy and Buildings*, *205*, 109564. https://doi.org/10.1016/j.enbuild.2019.109564

Muslim, I., Karo, K., Informatika, F., & Telkom, U. (2020). Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan. *Journal of Software Engineering, Information and Communication Technology*, *1*(1), 10–16.

Nawangsari, R. P., Kusumaningrum, R., & Wibowo, A. (2019). Word2vec for Indonesian Sentiment Analysis towards Hotel Reviews: An Evaluation Study. *Procedia Computer Science*.

Nurdin, A., Anggo Seno Aji, B., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal Tekno Kompak*, *14*(2), 74. https://doi.org/10.33365/jtk.v14i2.732

Nurdin, A., Seno aji, B., Bustamin, A., & Abidin, Z. (2020). PERBANDINGAN KINERJA WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT PADA KLASIFIKASI TEKS. *Jurnal Tekno Kompak*, *14*, 74. https://doi.org/10.33365/jtk.v14i2.732

Purnama, N. (2021). Implementasi Doc2vec untuk rekomendasi penginapan di Bali. *Jurnal Teknik Informatika Unika St. Thomas (JTIUST)*, *06*(02), 2657–1501.

Ramadhan Al-Mubaraq, R., Al Faraby, S., & Dwifebri Purbolaksono, M. (2021). *Analisis Sentimen pada Ulasan Film dengan Kombinasi Seleksi Fitur Chi-Square dan TF-IDF menggunakan Metode KNN*. *8*(5), 10116–10126.

Rohman, A. N., Luviana Musyarofah, R., Utami, E., & Raharjo, S. (2020). Natural Language Processing on Marketplace Product Review Sentiment Analysis. *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, 1–5. https://doi.org/10.1109/ICORIS50180.2020.9320827

Rozy, F., Rangkuti, S., Fauzi, M. A., Sari, Y. A., Dewi, E., & Sari, L. (2018). Analisis Sentimen Opini Film Menggunakan Metode Naïve Bayes dengan Ensemble Feature dan Seleksi Fitur Pearson Correlation Coefficient. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(12), 6354–6361.

Septian, J. A., Fahrudin, T. M., & Nugroho, A. (2019). Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF - IDF dan K - Nearest Neighbor. *JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTATION*, 43–49.

Shuai, Q., Huang, Y., Jin, L., & Pang, L. (2018). Sentiment Analysis on Chinese Hotel Reviews with Doc2vec and Classifiers. *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 1171–1174. https://doi.org/10.1109/IAEAC.2018.8577581

Sihombing, L., Hannie, H., & Dermawan, B. (2021). *Sentimen Analisis Customer Review Produk Shopee Indonesia Menggunakan Algortima Naïve Bayes*

Classifier. *5*, 233–242. https://doi.org/10.29408/edumatic.v5i2.4089

Sistem, R., Lestandy, M., Abdurrahim, A., & Syafa, L. (2021). *Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent*. *5*(10), 802–808.

Trisari, W., Putri, H., Hendrowati, R., & Belakang, L. (2020). *Penggalian Teks Dengan Model Bag of Words Terhadap*. *2*(1), 129–138.

Wang, Q., Zhang, W., Li, J., Mai, F., & Ma, Z. (2022). Effect of online review sentiment on product sales: The moderating role of review credibility perception. *Computers in Human Behavior*, *133*, 107272. https://doi.org/https://doi.org/10.1016/j.chb.2022.107272

Wang, X., Zhou, T., Wang, X., & Fang, Y. (2022). Harshness-aware sentiment mining framework for product review. *Expert Systems with Applications*, *187*, 115887. https://doi.org/https://doi.org/10.1016/j.eswa.2021.115887

Willy, Setiawan, E., & Nugraha, F. (2019). *Implementation of Decision Tree C4.5 for Big Five Personality Predictions with TF-RF and TF-CHI2 on Social Media Twitter*. 114–119. https://doi.org/10.1109/IC3INA48034.2019.8949601

Yennimar, Y., & Rizal, R. (2019). Comparison of Machine Learning Classification Algorithms in Sentiment Analysis Product Review of North Padang Lawas Regency. *SinkrOn*, *4*, 268. https://doi.org/10.33395/sinkron.v4i1.10416

Zhang, W., Li, Y., & Wang, S. (2019). Learning document representation via topic-enhanced LSTM model. *Knowledge-Based Systems*, *174*, 194–204. https://doi.org/https://doi.org/10.1016/j.knosys.2019.03.007

PUSTAKA ELEKTRONIK

Simon Kemp, 5 April 2022, Digital 2022 Indonesia :Internet use in Indonesia 2022, https://datareportal.com/reports/digital-2022%20indonesia?rq=indonesia%202022

LAMPIRAN