

Analisis Sentimen Review Aplikasi MyPertamina menggunakan *Word Embedding Fasttext* dan Algoritma *K-Nearest Neighbor*

Nanda Sepriadi, Elvia Budianita, M.Fikry, Pizaini

Fakultas Sains dan Teknologi, Program Studi Teknik Informatika,
Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia
Email: nanda48898@gmail.com

ABSTRAK

Pertamina adalah perusahaan yang dimiliki negara bergerak di bidang energi seperti minyak, gas serta energi baru dan energi terbarukan. Pertamina juga telah menciptakan sebuah aplikasi yang dapat diakses secara *mobile* yang tersedia di *Play Store* untuk memudahkan konsumen dalam bertransaksi pembayaran bahan bakar minyak (BBM) yang dapat diakses melalui aplikasi MyPertamina yang dapat di download melalui App Store dan Play Store. Playstore adalah platform Google yang menyediakan aplikasi *mobile* dan fasilitas pengguna untuk berbagi ulasan. Dalam penelitian ini, pendekatan yang digunakan untuk analisis sentimen adalah menggunakan kamus *Lexicon Based* dan metode *K-Nearest Neighbor*. Kamus leksikon yang diterapkan yaitu *vader* yang memberikan label secara otomatis, sementara metode *K-Nearest Neighbor* digunakan untuk melakukan pengelompokan atau klasifikasi. Maksud penelitian ini dilakukan adalah untuk memahami bagaimana pengelompokan tanggapan masyarakat terhadap aplikasi MyPertamina dan mengetahui performa terbaik metode *K-NN* dengan 3 perbandingan data latih dan data uji. Penelitian ini menggunakan data berjumlah 8000 data, data ini kemudian dikelompokkan menjadi tiga bagian data latih dan data uji, dengan bagian 90/10, 80/20 dan 70/30. Diperoleh hasil pelabelan menghasilkan 1405 data dengan label positif, 1698 data dengan label netral, dan 4897 data dengan label negatif. Akurasi tertinggi diperoleh dataset B dengan porsi data 80/20 dengan nilai $K = 9$ yang menghasilkan akurasi sebesar 73%, presision sebesar 70% dan recall sebesar 73% tanpa menggunakan *Undersampling*. Sedangkan data yang sama menggunakan *Undersampling* diperoleh akurasi sebesar 57%, presision sebesar 73% dan recall sebesar 57%. Penurunan akurasi ini disebabkan oleh pengurangan jumlah data training akibat menggunakan *undersampling*.

Kata Kunci: Analisi Sentiment, *lexicon based*, MyPertamina, *K-NN*, *FastText*

ABSTRACT

Pertamina is a state-owned company engaged in the energy sector such as oil, gas as well as new energy and renewable energy. Pertamina has also created a mobile-accessible application available on the Play Store to make it easier for consumers to make fuel oil (BBM) payment transactions which can be accessed through the MyPertamina application which can be downloaded via the App Store and Play Store. Playstore is a Google platform that provides mobile applications and user facilities for sharing reviews. In this study, the approach used for sentiment analysis is using a Lexicon Based dictionary and the K-Nearest Neighbor method. The lexicon dictionary that is applied is vader which gives labels automatically, while the K-Nearest Neighbor

method is used to group or classify. The purpose of this research was to understand how people's responses to the MyPertamina application were grouped and to find out the best performance of the K-NN method with 3 comparisons of training data and test data. This study uses data totaling 8000 data, this data is then grouped into three parts training data and test data, with sections 90/10, 80/20 and 70/30. The labeling results obtained resulted in 1405 data with positive labels, 1698 data with neutral labels, and 4897 data with negative labels. The highest accuracy was obtained for dataset B with a data portion of 80/20 with a value of $K = 9$ which resulted in an accuracy of 73%, a precision of 70% and a recall of 73% without using undersampling. While the same data using undersampling obtained an accuracy of 57%, a precision of 73% and a recall of 57%. This decrease in accuracy is caused by a reduction in the amount of training data due to using undersampling.

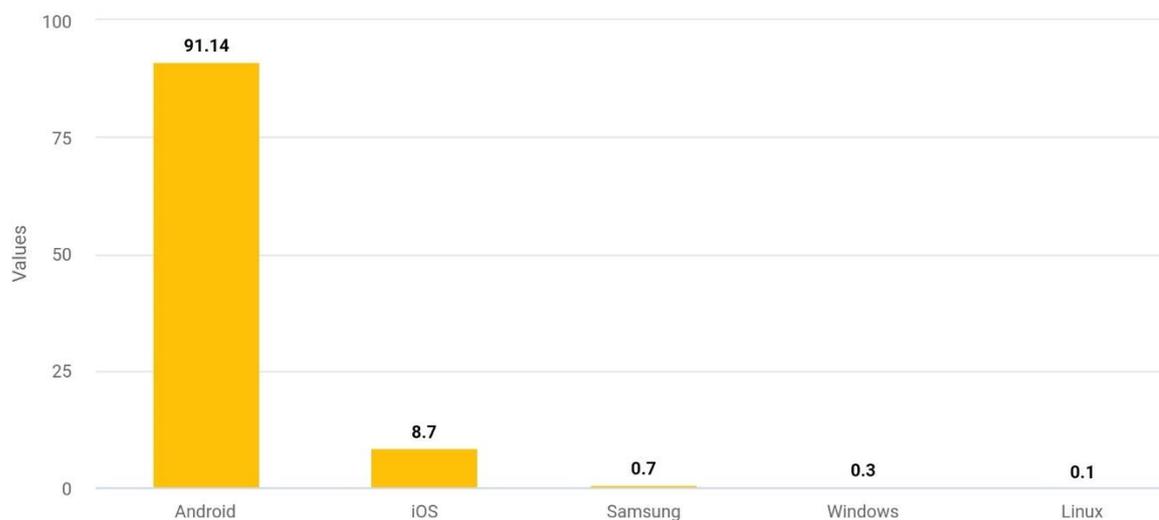
Keywords: *Sentiment Analysis , lexicon based, MyPertamina, K-NN, FastText*

1. PENDAHULUAN

Perkembangan teknologi informasi di Indonesia pada era saat ini berlangsung dengan cepat. Hampir semua lapisan masyarakat Indonesia mengadopsi teknologi informasi dalam berbagai aspek kehidupan. Seiring dengan kemajuan teknologi informasi, terjadi terobosan baru dalam bentuk sistem pembayaran elektronik. Masyarakat Indonesia dapat dengan mudah menggunakan pembayaran elektronik (e-payment) untuk berbagai transaksi, keperluan sehari-hari, layanan publik, dan berbagai kebutuhan lainnya. Pada era sekarang di Indonesia, telah terjadi pergeseran menuju transaksi tanpa uang tunai. Pembayaran non tunai merupakan metode yang dapat digunakan untuk melaksanakan transaksi yang dilakukan melalui aplikasi yang tersedia salah satu contohnya adalah MyPertamina (Irawan, 2019). PT. Pertamina (Persero) merupakan perusahaan milik negara yang bergerak di bidang energi meliputi minyak, gas serta energi baru dan terbarukan. PT. Pertamina juga telah menciptakan sebuah aplikasi yang dapat diakses secara mobile yang tersedia di Play Store untuk memudahkan konsumen dalam bertransaksi melalui pembayaran BBM (bahan bakar minyak) melalui aplikasi MyPertamina yang dapat di download melalui App Store dan Play Store.

Sistem operasi mobile berperan sebagai perangkat lunak utama yang bertanggung jawab dalam mengelola dan mengendalikan perangkat keras secara langsung, serta mengatur dan mengontrol perangkat lunak lainnya agar dapat berfungsi dengan baik. Agar memungkinkan pengembang perangkat lunak menciptakan aplikasi-aplikasi inovatif, sistem operasi perlu didesain dengan tingkat

fleksibilitas yang tinggi. Saat ini, dalam perkembangan sistem operasi smartphone, terdapat dua sistem operasi yang paling umum digunakan, yaitu Android OS dan iOS (Iphone OS).(Anggreni et al., 2022). berikut ini adalah persentase penggunaan *Mobile operating system* dapat dilihat pada Gambar 1.



Gambar 1: Presentase Pengguna *Mobile Operating System*

Pada gambar 1 dapat dilihat bahwa *mobile operating system* pada android menghasilkan presentase tertinggi dibandingkan dengan pengguna operating system yang lain yaitu dengan data pengguna android 91.14%, pengguna IOS 8.7% dan lainnya di bawah 0.7%. Android, sebagai sistem operasi yang digunakan oleh hampir semua vendor perangkat smartphone, telah menjadi salah satu platform utama dalam industri tersebut. Perkembangan Android sangat signifikan, terbukti pada kuartal pertama tahun 2022 Play store memiliki total 3.298.329 aplikasi yang telah tersedia untuk perangkat Android . Tak hanya itu, terdapat sekitar 111,3 miliar unduhan aplikasi pada tahun 2021 dari Google Play, toko utama aplikasi bagi perangkat Android(Wahyudi & Kuswandi, 2022). Sekarang, ketersediaan aplikasi di Google Play semakin melimpah. Aplikasi-aplikasi ini memiliki berbagai keunggulan yang menjadikan perangkat dengan sistem operasi Android sangat bermanfaat dan berguna.

Salah satu cara untuk menggunakan aplikasi MyPertamina, masyarakat harus mengunduh aplikasi tersebut di Playstore. Playstore, sebagai platform konten digital yang dimiliki oleh Google, menyediakan berbagai produk aplikasi mobile dan memberikan fasilitas kepada pengguna untuk memberikan ulasan mengenai

pengalaman mereka setelah menggunakan aplikasi tersebut. Ulasan tersebut dapat berupa pujian, keluhan, atau saran yang berguna bagi pengembangan aplikasi tersebut (Saputra et al., 2019). Ulasan dari pengguna telah menjadi alat yang paling efektif dan efisien dalam mendapatkan informasi mengenai produk atau jasa tertentu. Sebagian besar pengguna internet cenderung melihat ulasan produk sebelum mereka memutuskan untuk menggunakannya, karena ulasan dari pengguna lain memberikan informasi terbaru yang sangat berharga mengenai produk tersebut (Edyt Daryfayi Putra Daulay & Asror, 2020). Namun, memantau atau mengorganisir opini masyarakat tidaklah mudah. Jumlah opini yang tersebar di media sosial terlalu besar untuk diproses secara manual. Oleh karena itu, peneliti tertarik untuk menguji sejauh mana analisis ulasan pengguna aplikasi MyPertamina yang dapat dilakukan.

Penelitian ini berfokus pada Analisa sentiment pengguna aplikasi MyPertamina yang ada didalam situs Google play, untuk pengambilan datanya di ambil dari ulasan yang dibuat oleh pengguna aplikasi MyPertamina. Analisa sentiment pengguna aplikasi MyPertamina bisa dipengaruhi oleh beberapa hal yang belum menjadi perhatian Pertamina. Hal ini mungkin terjadikarenan adanya beberapa factor yang harus diperbaiki dan belum di perbaiki oleh pihak Pertamina (Masturoh, 2021).

Selanjutnya untuk representasi kalimat pada review aplikasi MyPertamina, digunakan model word embeddings *FastText*. *FastText* merupakan sebuah metode word embedding yang merupakan perluasan dari word2vec. Metode ini melakukan pembelajaran representasi kata dengan memperhatikan informasi subword. Pada metode ini, setiap kata direpresentasikan sebagai kumpulan karakter N-gram. Dengan demikian, penggunaan teknik embedding dalam pemrosesan bahasa alami dapat meningkatkan kemampuan untuk memahami makna kata-kata dengan sufiks dan prefiks yang lebih pendek. Pada teknik representasi vektor, setiap karakter ngram memiliki kaitan dengan sebuah vektor tertentu, sementara kata-kata direpresentasikan sebagai hasil penjumlahan dari representasi vektor karakter ngram yang membentuk kata tersebut. (Kusairi et al., 2022).

Adapun penelitian terkait analisis sentimen pada ulasan Google play oleh Siti pada tahun 2019 dimana di lakukan analisis sentiment pada E-Wallet seperti DANA dan OVO menggunakan algoritma *K-Nearest Neighbor*. Penelitian ini menunjukkan bahwa model *K-Nearest Neighbor* dengan nilai $k=1$ pada platform DANA/OVO

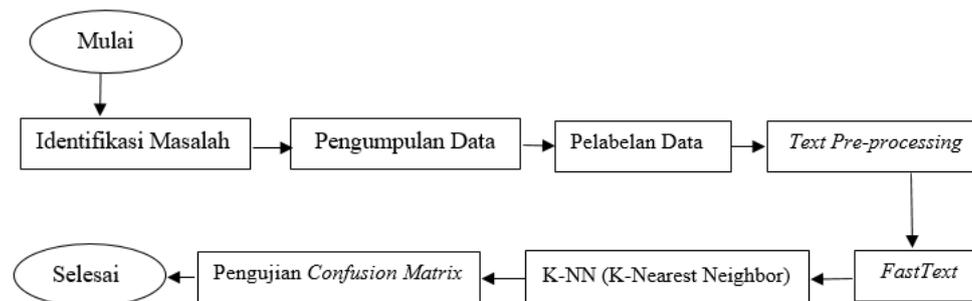
menghasilkan akurasi tertinggi. Hasil penelitian menunjukkan bahwa penggunaan metode ini pada OVO dengan dua kelas menghasilkan tingkat akurasi sebesar 84,86%, sedangkan pada DANA dengan dua kelas menghasilkan tingkat akurasi sebesar 86,64%. Penelitian selanjutnya pada penelitian sebelumnya dilakukan oleh firdaus dengan membandingkan algoritma naïve bayes dan *K-Nearest Neighbor* memiliki tingkat akurasi yang berbeda, Algoritma Naïve Bayes menghasilkan tingkat akurasi sebesar 75,30%, sedangkan algoritma *K-Nearest Neighbor* menghasilkan tingkat akurasi sebesar 86,09%. (Masturoh, 2021). Terakhir adalah penelitian dari Aluisius Dwiki Adhi Putra dan Safitri Juanita yang berjudul Analisis Sentimen Pada Ulasan Pengguna Aplikasi Bibit Dan Bareksa Dengan Algoritma KNN melalui tahapan modelling dengan menggunakan algoritma *K-Nearest Neighbors* dan perbandingan 60:40 untuk data training dan data testing, pengguna aplikasi Bibit dan Bareksa memperoleh hasil yang menunjukkan tingkat akurasi, precision, dan recall yang berbeda-beda. Pada aplikasi Bibit, tingkat akurasi mencapai 85,14%, sedangkan precision dan recall masing-masing sebesar 91,91% dan 76,44%. Sementara itu, pada aplikasi Bareksa, tingkat akurasi mencapai 81,70%, dengan precision dan recall masing-masing sebesar 87,15% dan 75,73%. (Adhi Putra, 2021).

Pada penelitian sebelumnya, dilakukan perbandingan kinerja word embedding seperti Word2Vec, GloVe, dan *FastText*. Hasil penelitian menunjukkan bahwa Word2Vec merupakan jenis word embedding yang berupa matriks bobot yang diperoleh dari hasil pelatihan unsupervised learning neural network. Word embedding seperti GloVe dan *FastText* memiliki keunggulan yang berbeda dari Word2Vec. GloVe, misalnya, dapat melibatkan informasi statistik global dalam dokumen untuk memperoleh representasi kata yang lebih baik. Sementara itu, *FastText* memetakan subword suku kata sehingga dapat menghasilkan representasi kata yang lebih baik dalam bahasa-bahasa yang memiliki banyak varian kata. Dalam penelitian terbaru, *FastText* menunjukkan performa terbaik dalam evaluasi kinerja yang diukur menggunakan F-measure, sehingga dapat dikatakan lebih unggul dibandingkan dua metode word embedding lainnya. (Nurdin et al., 2020).

Penelitian ini akan menganalisis sentimen review aplikasi MyPertamina di Google Play Store menggunakan metode word embedding *FastText* dan algoritma K-NN. Sentimen akan dikelompokkan menjadi tiga kelas: positif, negatif, dan netral.

Namun, penelitian ini memiliki batasan dalam penggunaan dataset hanya terbatas pada review pengguna aplikasi MyPertamina dengan jumlah data sebanyak 8000. Pelabelan data akan dilakukan secara otomatis menggunakan kamus vader lexicon. Algoritma K-NN akan menentukan klasifikasi berdasarkan jarak terdekat antara instance query dan sampel latihan. Jarak ini dapat dihitung menggunakan Euclidean Distance. Dalam penelitian ini, *FastText* digunakan sebagai kamus vektor untuk meningkatkan akurasi performa pada dataset review MyPertamina. Metode ini akan diimplementasikan menggunakan bahasa pemrograman *Python*, dan confusion matrix akan digunakan untuk menguji akurasi hasil klasifikasi sentimen dari algoritma K-NN

2. METODE PENELITIAN



Gambar 2: Gambaran Metodologi Penelitian

Identifikasi Masalah

Tujuan dari penelitian ini adalah untuk mengidentifikasi sentimen dari review aplikasi MyPertamina di Google Play dengan menggunakan metode *K-Nearest Neighbor* (K-NN) dan pembobotan kata menggunakan fitur *FastText*. Dengan demikian, penelitian ini diharapkan dapat mencapai tingkat akurasi yang tinggi dalam analisis sentimen review aplikasi MyPertamina.

Pengumpulan Data

Dalam penelitian ini, proses pengumpulan data dilakukan dengan mengambil data review langsung dari sumber asli. Data review pengguna aplikasi MyPertamina diambil dari situs Google Play berbahasa Indonesia yang diakses melalui browser menggunakan teknik *web scraping*. Proses *web scraping* dilakukan secara otomatis dengan menggunakan program *Python*, dengan URL alamat:

play.google.com/store/apps/. Proses mendapatkan data review dari pengguna aplikasi MyPertamina, perlu membuat kodingan *Python* pada Google Colab terlebih dahulu, kemudian menginstal ekstensi Google Play Store di Google Colab, agar nantinya bisa terkoneksi ke Google Play Store dan bisa mengakses serta mengambil data yang di butuhkan sesuai keinginan, kemudian langkah selanjutnya memasukkan kata kunci com.daftarn.mypertamina. Data yang digunakan dalam penelitian ini sebanyak 8000 data dari Google Play Store. kemudian data disimpan dengan format Data8fix.xlsx. Setelah itu, data tersebut akan diterjemahkan ke dalam bahasa Inggris dan kemudian dilakukan pelabelan sesuai dengan kamus leksikon vader berbahasa Inggris. Selanjutnya, hasil terjemahan akan dipindahkan kembali ke dalam bahasa Indonesia. Selanjutnya, data akan dibagi menjadi dua kelompok, yaitu data latih (training data) dan data uji (testing data).

Pelabelan Data

Pelabelan data dilakukan untuk melabelkan data menjadi 3 kelas yaitu kelas positif, netral dan negatif. Pelabelan secara manual ini dianggap membutuhkan waktu yang lama jika data yang digunakan itu terlalu banyak, maka dibutuhkan metode berbasis lexicon yaitu vader. C.J. Hutto dan Eric Gilbert dari Georgia Institute of Technology berkolaborasi pada tahun 2014 dalam menciptakan Lexicon VADER (Valanced Aware Dictionary Sentiment Reasoner), yang digunakan untuk memberi label secara otomatis pada data. VADER (Valanced Aware Dictionary Sentiment Reasoner) adalah pendekatan leksikal yang digunakan sebagai model untuk analisis suasana hati. Selain itu, intensitas emosi yang dihasilkan oleh VADER dapat digunakan untuk mengevaluasi berbagai data. Sudut pandang VADER berakar pada pendekatan yang manusiawi, penilaian manusia, serta kebijaksanaan manusia. Kamus leksikon umumnya digunakan untuk mengevaluasi frasa dan kalimat sebagai sentimen tanpa perlu mengacu pada sumber lain. Simbol angka seperti "negatif, netral, dan positif" sering digunakan untuk mengklasifikasikan sentimen. Salah satu aspek unik dari pendekatan leksikal ini adalah tidak memerlukan data berlabel untuk melatih modelnya. (Asri et al., 2022). VADER merupakan salah satu jenis kamus leksikon yang mengandung sekitar 7500 token kata berbahasa Inggris, termasuk emotikon dan istilah terkait akronim dan inisial. Teks hasil pengambilan data dari web scraping akan

dinilai berdasarkan kamus leksikon ini untuk menentukan apakah sentimennya positif, netral, atau negatif, serta menambahkan skor total.

Text Pre-processing

Text pre-processing adalah tahapan melakukan persiapan data yang telah didapat agar dapat dilakukan pembersihan data atau teks dari yang tidak terstruktur atau yang masih mengandung *noisy* seperti : koma, garis miring, url, emoticon, angka dan lain sebagainya menjadi terstruktur, agar nantinya data yang di gunakan dalam penelitian ini lebih mudah untuk di proses dan di olah, sehingga perhitungan akan optimal. Dalam penelitian ini, di mana dalam tahapan *text pre-processing* data dibersihkan melalui beberapa tahapan seperti *cleaning*, *case folding*, *stopword removal*, dan *stemming* (Alrajak et al., 2020).

1. *Cleaning*

Tahap ini bertujuan untuk membersihkan data dengan menghapus data/komentar yang duplikat dan atribut yang tidak diperlukan dari komentar melalui proses Case Folding.

2. *Case Folding*

Tahapan case folding melibatkan transformasi seluruh huruf dalam dokumen menjadi huruf kecil, hanya menerima huruf 'a' hingga 'z'. Karakter selain huruf tersebut akan dihapus dan dianggap sebagai pemisah.

3. *Stopwords Removal*

Dalam tahapan ini, dilakukan penghapusan kata-kata yang termasuk dalam kategori stopword. Tujuannya adalah untuk menghilangkan kata-kata yang dianggap tidak penting atau tidak memiliki makna, seperti kata hubung "yang," "ini," "dengan," dan kata hubung lainnya.

4. *Stemming*

Proses ini memiliki tujuan untuk menemukan kata dasar dengan cara menghapus semua imbuhan yang melekat pada kata. Imbuhan yang dihilangkan meliputi awalan, sisipan, akhiran, serta kombinasi awalan dan akhiran.

FastText

Pada penelitian ini pembobotan kata dengan pembobotan *FastText*. *FastText* adalah pengembangan dari word2vec yang dikembangkan oleh *Facebook AI Research*

(FAIR). Model ini yang akan mengolah kata-kata itu menjadi vektor, *FastText* telah terbukti memberikan kinerja yang sangat baik dalam merepresentasikan kata dan mengklasifikasikan kalimat, terutama dalam kasus kata-kata yang jarang terjadi, dengan memanfaatkan informasi tingkat karakter. Pada tahapan *FastText*, data yang telah melewati tahap preprocessing selanjutnya akan dibuat model *FastText* tujuannya untuk mengubah setiap kata pada data training menjadi nilai vector agar dapat dilakukan klasifikasi pada Algoritma K-NN(Ihsan, 2021). Cara kerja *FastText* melibatkan penggunaan n-gram, di mana tidak hanya satu kata secara utuh yang diproses, tetapi memperhatikan bagian kata yang lebih kecil. Sebagai contoh, saat bekerja dengan kata "program", *FastText* akan menggunakan trigram (n=3) untuk membagi kata tersebut menjadi "pro", "rog", "ogr", "gra", "ram".

K-Nearest Neighbor (K-NN)

Masuk pada proses klasifikasi, proses ini dilakukan untuk menguji akurasi metode *K-Nearest Neighbor (KNN)*. Algoritma *K-Nearest Neighbors (k-NN)* merupakan metode klasifikasi yang didasarkan pada contoh-contoh dasar yang tidak membangun representasi deklaratif eksplisit dari kategori, melainkan mengandalkan label kategori yang terdapat pada dokumen pelatihan yang mirip dengan dokumen uji. Dalam proses pengujian dokumen, sistem akan menemukan k tetangga terdekat antara dokumen pelatihan. Rata-rata kesamaan antara dokumen tetangga terdekat dan dokumen uji digunakan sebagai bobot kelas untuk dokumen tetangga tersebut. (Sahara & Permana, 2019). Dalam menentukan sentimen dari ulasan pada aplikasi MyPertamina, metode *K-Nearest Neighbor (KNN)* digunakan untuk melakukan klasifikasi berdasarkan data pembelajaran yang memiliki jarak paling dekat dengan objek yang sedang dianalisis. Tujuan dari metode ini adalah untuk mengklasifikasikan objek baru dengan mengacu pada atribut-atributnya dan sampel pelatihan yang telah ada. Proses klasifikasi menggunakan Bahasa pemrograman *Python*. Dalam metode k-NN, peneliti menghasilkan sebuah model dengan memilih nilai k dari rentang 1 hingga 10 untuk mencari tingkat akurasi yang tinggi saat menguji dataset. Akurasi yang di dapatkan nantinya akan diuji Kembali menggunakan confusion matrix untuk memastikan kembali apakah hasil akurasi yang diperoleh sama dengan hasil perhitungan dari confusion matrix.

Langkah-langkah dalam klasifikasi menggunakan metode *K-Nearest Neighbors* (K-NN) adalah sebagai berikut:

1. Tentukan nilai K, misalnya k=23, yang berarti kita akan mengambil 23 dokumen dengan jarak terdekat dari dokumen pelatihan yang sedang diuji.
2. Setelah itu, lakukan perhitungan jarak antara data baru dengan setiap label data menggunakan jarak Euclidean terhadap seluruh data pelatihan model. Selanjutnya, untuk menghitung tingkat kesamaan dokumen tersebut, digunakan metode Euclidean distance dengan rumus sebagai berikut:

$$D(X,Y) = \sqrt{\sum_{k=1}^n (X_k - Y_k)^2}$$

3. Langkah selanjutnya adalah mengurutkan hasil dari euclidean distance sesuai dengan jarak K yang telah ditentukan sebelumnya. Dalam hal ini, nilai jarak yang digunakan adalah nilai jarak yang paling kecil.
4. Setelah itu, model yang digunakan adalah model yang merupakan mayoritas dari atribut kelas pada tetangga terdekat yang telah dipilih untuk melakukan prediksi kelas terhadap data yang paling baru.

Pengujian *Confusion Matrix*

Dalam melakukan klasifikasi, confusion matrix saat ini merupakan alat yang banyak digunakan dalam mengevaluasi performa. Hasil pengujian akan dievaluasi menggunakan confusion matrix untuk memeriksa dan menganalisis hasil yang diperoleh dari sistem tersebut. Dalam Confusion Matrix, akurasi memberikan gambaran tentang sejauh mana model mampu mengklasifikasikan data dengan benar. Presisi menggambarkan tingkat dari akurasi yang dipanggil dengan data yang dihasilkan oleh model, sementara recall menggambarkan keberhasilan model ini dalam melakukan pencarian informasi yang relevan.

3. HASIL DAN PEMBAHASAN

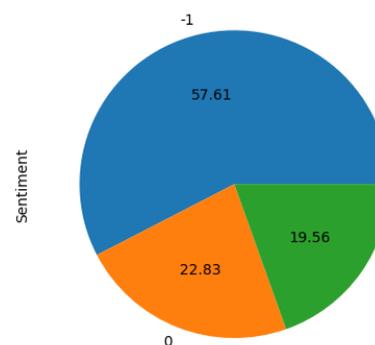
Tahap ini melibatkan hasil dan pembahasan pada setiap proses dalam penelitian ini, khususnya terkait analisis sentimen terhadap ulasan aplikasi MyPertamina di Google Play Store menggunakan *FastText* dan K-NN.

Translate dan Pelabelan

Sebelum data dilabel menggunakan pendekatan *Lexicon-based*, dilakukan terjemahan data dari bahasa Indonesia ke bahasa Inggris, karena dalam proses pelabelan data menggunakan kamus sentimen VADER yang berbahasa Inggris. Setelah dilakukan proses pelabelan, dataset yang berjumlah 8000 data ulasan aplikasi MyPertamina menghasilkan 1405 data dengan label positif, 1698 data dengan label netral, dan 4897 data dengan label negatif. Hasil terjemahan, pelabelan, dan visualisasi data dapat dilihat dalam gambaran di bawah ini. Tabel 2 yaitu hasil *translate* dan pelabelan, gambar 3. yaitu visualisasi jumlah kelas sentiment setelah dilakukan pelabelan.

Tabel 2. Hasil *translate*/terjemah dan pelabelan

Conten	Content	Vader Sentiment	Sentiment
menyusahkan masyarakat	trouble the community	-0,4019	Negative
Sangat membantu	Very helpful	0.4754	Positive
Kerenn	Cool	0.3612	positive



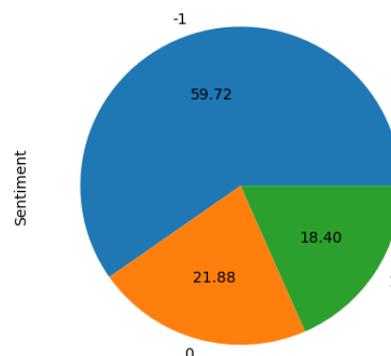
Gambar 3: Visualisasi data hasil dari hasil pelabelan *review* aplikasi MyPertamina

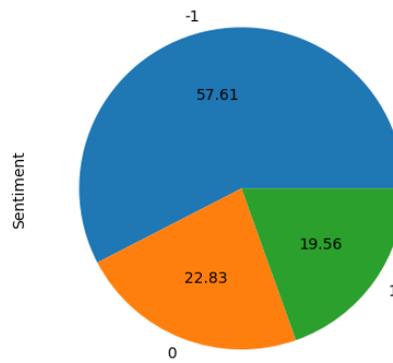
Setelah data diterjemahkan dan dilabeli, langkah selanjutnya adalah membagi data yaitu menjadi tiga bagian dengan perbandingan antara jumlah data latih (Training) dan data uji (Test), yaitu 90%:10%, 80%:20%, dan 70%:30%. Pembagian jumlah data bisa dilihat pada Tabel 3.

Tabel 3. Pembagian Jumlah Data

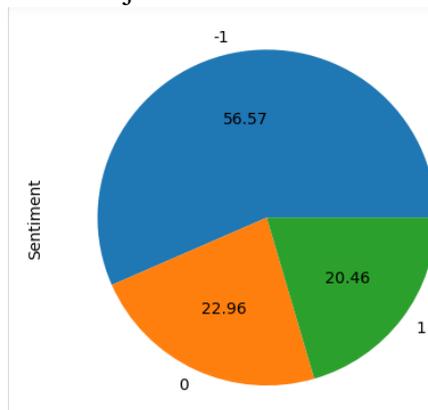
Dataset	Jumlah Data			
	Latih (Training)	Uji (Test)	Latih (Training)	Uji (Test)
Dataset A	70%	30%	5600	2400
Dataset B	80%	20%	6400	1600
Dataset C	90%	10%	7200	800

Tahap selanjutnya melibatkan pembagian data menjadi tiga kelompok, di mana setiap kelompok memiliki jumlah kelas sentimen yang berbeda. Dataset A terdiri dari 1325 data dengan sentimen positif, 1575 data dengan sentimen netral, dan 4300 data dengan sentimen negatif. Dataset B terdiri dari 1252 data dengan sentimen positif, 1461 data dengan sentimen netral, dan 3687 data dengan sentimen negatif. Sementara itu, dataset C terdiri dari 1146 data dengan sentimen positif, 1286 data dengan sentimen netral, dan 3136 data dengan sentimen negatif. Gambar 4, 5, dan 6 menunjukkan visualisasi jumlah kelas sentimen dalam dataset A, B, dan C secara berturut-turut.

**Gambar 4:** Visualisasi jumlah kelas sentiment pada Dataset A



Gambar 5: Visualisasi jumlah kelas sentiment pada Dataset B



Gambar 6: Visualisasi jumlah kelas sentiment pada Dataset C

Text Pre-processing

Tahap *Text pre - processing* memegang peranan penting dalam analisis klasifikasi. Hasil *Text pre-processing* pada penelitian ini dapat ditemukan dalam Tabel 4.

Tabel 4. *Text pre-processing*

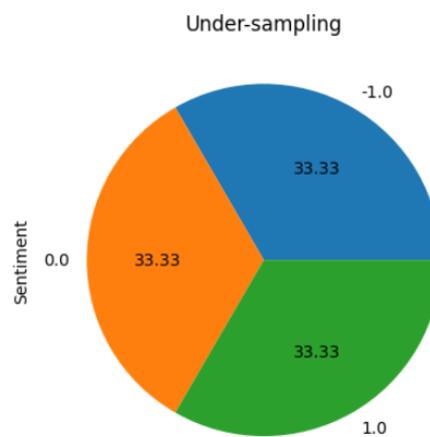
<i>Pre-processing</i>	Contoh Coding	Hasil
Kalimat dasar		Trimakasi untuk aplikasinya ini.
<i>Celaning</i>	<code>def cleansing(Content):</code>	Trimakasi untuk aplikasinya ini.
<i>Case Folding</i>	<code>def preprocessing (Content):</code>	trimakasi untuk aplikasinya ini.
<i>Stopwords Removal</i> <i>Stemming</i>	<code>def stopwordsremoval (Content):</code> <code>factory = StemmerFactory()</code>	trimakasi untuk aplikasinya trimakasi untuk aplikasi

Undersampling

Undersampling merupakan proses menyeimbangkan data dengan mengurangi ukuran kelas yang berlimpah. Proses ini akan menggunakan bagian dataset B dengan pembagian data 80%/20%. Selanjutnya presentase terkait data *Undersampling* bisa dilihat pada tabel 5 dan gambar 7 visualisasi data *Undersampling*.

Tabel 5. *Undersampling data*

Dataset	Positif	Netral	Negatif	Jumlah Data
Dataset B	1325	1325	1325	3756



Gambar 7. Visualisasi data *Undersampling*

FastText

Dalam penelitian ini, *FastText* digunakan untuk mengkonversi kalimat atau kata menjadi vektor. *FastText* memanfaatkan data pelatihan dan pengujian untuk membangun kamus vektor. Kamus tersebut akan digunakan untuk melakukan vektorisasi pada data latih dan data uji dari ulasan, sehingga dapat menjadi inputan untuk metode K-N. Cara kerja vektorisasi adalah dengan menghitung vektor untuk setiap kata dalam kamus vektor, kemudian dilakukan penjumlahan vektor sesuai dengan kalimat pada ulasan. Berikut ini adalah contoh vektorisasi:

Kamus *Word embedding FastText*:

Kamus Vektor = ([[-0.27668244, -0.31748974, 1.3061035, ..., -0.49450234,
0.9806198, 1.2579889],
[-0.12932043, -0.58630246, 1.0095788, ..., 0.09615417,
-0.04567867, -0.4587119],
[-0.31706437, 1.7710205, 1.7432487, ..., -0.19766723,
0.1831129, 0.9847286],
...,

Aplikasi = [-0.07067239, -0.1646062 , -0.12944901, ..., -0.10813623,
0.32489496, 0.09758133],

Mempersulit = [-0.07788797, -0.1602737 , -0.11642849, ..., -0.10081337,
0.33393034, 0.09264055],

Diri = [-0.08689163, -0.1944982 , -0.14723593, ..., -0.120224 ,
0.37440884, 0.10334609], dtype=float32)

Sendiri =[-0.07165466, -0.16273238, -0.12529205, ..., -0.10404629,
0.31876343, 0.08580446],

Aplikasi mempersulit diri sendiri =[-0.03293088, -0.07296342, -0.05460508, ..., -
0.04607699, 0.14666049, 0.04117316],

Klasifikasi K-NN

Setelah data diberi bobot, langkah selanjutnya adalah melakukan proses klasifikasi. Dalam proses ini, klasifikasi dilakukan menggunakan library *K-Nearest Neighbor* di *Python*. Dataset yang terdiri dari 8000 data yang dari aplikasi Mypertamina, yang sebelumnya telah diberi bobot, dan sekarang akan digunakan dalam proses klasifikasi. Setelah data diberi bobot, data akan dibagi menjadi data training dan testing dengan perbandingan 90%:10%, 80%:20%, dan 70%:30%. Dalam penelitian ini, nilai K yang akan digunakan berkisar dari k=1 hingga k=10.

Tabel 6. Hasil Pengujian dari K-NN

Nilai K	Nilai Akurasi		
	A	B	C
	70% / 30%	80% / 20%	90% / 10%
K = 1	0.5520	0.5825	0.6150
K = 2	0.6720	0.7000	0.7050
K = 3	0.6545	0.6906	0.7062
K = 4	0.6554	0.6981	0.6975
K = 5	0.6762	0,7118	0.7225
K = 6	0.6700	0.7081	0.7125
K = 7	0.6712	0.7200	0.7250
K = 8	0.6754	0.7243	0.7212
K = 9	0.6879	0.7300	0.7225
K = 10	0.6854	0,7275	0.7300

Tabel di atas merupakan hasil pengujian menggunakan algoritma K-NN , dengan menggunakan nilai dari K = 1 hingga K =10 dengan perbandingan data 70%/30%, 80%/20%, 90%/10% . Hasilnya menunjukkan pada penerapan metode *K-Nearest Neighbors* pada Tabel 6 dengan penentuan nilai k=9 dan perbandingan data 80/20 serta nilai k = 10 dengan perbandingan data 90/10 yang nilai akurasi sama mencapai 73.00% menunjukkan hasil yang tertinggi diantara penentuan nilai k yang lain.

Pengujian *Confusion Matrix*

Confusion matrix digunakan untuk menghitung berbagai metrik evaluasi seperti akurasi, recall, presisi, dan error. Akurasi mengukur sejauh mana prediksi model mendekati nilai aktual. Berikut adalah perhitungan dan tabel pengukuran akurasi menggunakan confusion matrix:

Tabel 7. *Confusion Matrix*

	Positif	Negatif	Netral
Positif	Tpos	FPosNeg	FposNet
Negatif	FnegPos	Tneg	FnegNet
Netral	FNetPos	FnetNeg	TNet

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%$$

$$\text{Presisi} = \frac{\text{TP}}{\text{FP} + \text{TP}} \times 100\%$$

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \times 100\%$$

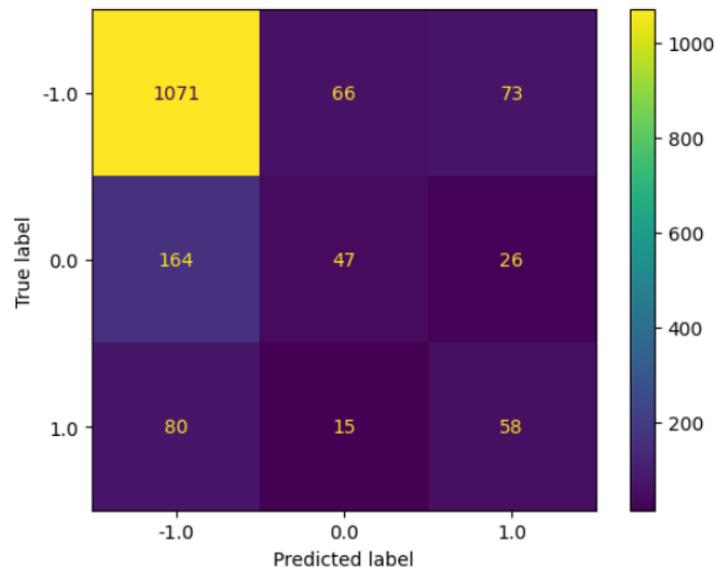
Dimana :

TP adalah singkatan dari "*True Positif*", yang mengacu pada data positif yang diprediksi dengan benar atau bernilai positif.

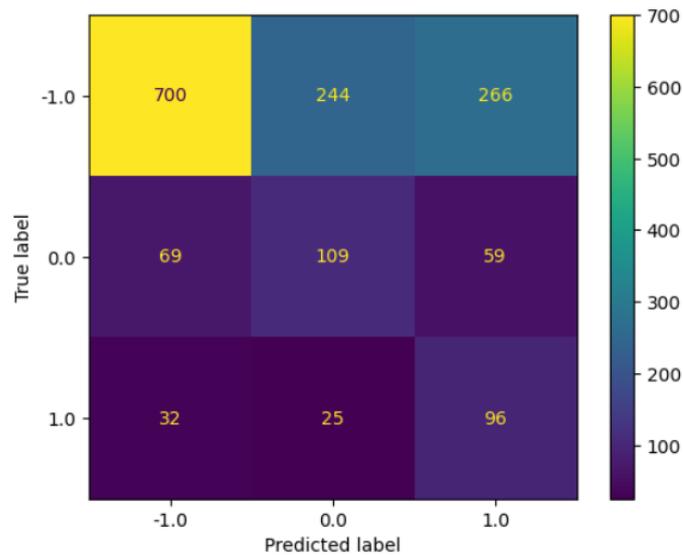
FN merupakan singkatan dari "*False Negatif*", yang menggambarkan data positif yang sebenarnya, tetapi diprediksi sebagai data negatif atau salah.

TN merupakan singkatan dari "*True Negatif*", yang merujuk pada data negatif yang diprediksi dengan benar atau bernilai negatif.

FP adalah singkatan dari "*False Positif*", yang merujuk pada data yang sebenarnya negatif, tetapi diprediksi sebagai data positif atau salah.



Gambar 8: *Confusion Matrix*



Gambar 9: *Confusion Matrix Setelah Undersampling*

Hasil akurasi, Precision, Recall sebelum dan sesudah under *Undersampling* yang didapatkan dari *confusion matrix* dengan perbandingan dataset B 80%/20% dan nilai K=9 dapat dilihat pada tabel berikut:

Tabel 8. Performa dan hasil dari pengujian sebelum dan sesudah *Undersampling*

Data B	Accuracy	Precision	Recall
Tanpa <i>Undersampling</i>	0.73	0.70	0.73
<i>Undersampling</i>	0.57	0.73	0.57

Dalam Tabel 8 terlihat hasil pengujian metode K-NN menggunakan fitur *FastText* dengan perbandingan dataset B dan hasil pengujian sebelum dan sesudah menggunakan *Undersampling*. Pada dataset B tanpa *Undersampling* diperoleh nilai akurasi sebesar 0.73, *precision* sebesar 0.70 dan *recall* sebesar 0.73. Sedangkan pada dataset B menggunakan *Undersampling* diperoleh hasil akurasi sebesar 0.57, *precision* 0.73 dan *recall* sebesar 0.57.

4. SIMPULAN

Berdasarkan penjelasan dan uraian sebelumnya, dilakukan analisis sentimen terhadap review aplikasi MyPertamina di *Google Play Store* menggunakan pendekatan *word embedding FastText* dan algoritma K-NN. Data review tersebut diklasifikasikan ke dalam tiga kelas sentimen, dan 3 perbandingan bagian data. Data juga dilakukan pengujian penyeimbangan dengan menggunakan *Undersampling* dan tidak menggunakan *Undersampling*. Dapat di tarik kesimpulan bahwa bagian data yang memiliki akurasi tertinggi diperoleh Dataset B dengan porsi data 80/20 dengan nilai $K = 9$ yang menghasilkan akurasi sebesar 73%, *precision* sebesar 70% dan *recall* sebesar 73% tanpa menggunakan *Undersampling*. Sedangkan data yang sama menggunakan *Undersampling* diperoleh akurasi sebesar 57%, *precision* sebesar 73% dan *recall* sebesar 57%. Penurunan akurasi ini disebabkan oleh pengurangan jumlah data training akibat menggunakan *Undersampling*.

5. DAFTAR PUSTAKA

- Adhi Putra, A. D. (2021). Analisis Sentimen pada Ulasan pengguna Aplikasi Bibit Dan Bareksa dengan Algoritma KNN. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 8(2), 636–646. <https://doi.org/10.35957/jatisi.v8i2.962>
- Alrajak, M. S., Ernawati, I., & Nurlaili, I. (2020). Analisis Sentimen Terhadap Pelayanan PT PLN di Jakarta pada Twitter dengan Algoritma K- Nearest Neighbor (K-NN). *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA)*, 110–122.
- Anggreni, P., Wayan, I., & Arsana, G. (2022). Preferensi Konsumen Terhadap Merek Smartphone Berdasarkan Sistem Operasi (Studi Perbandingan Smartphone menggunakan Iphone S/IOS dengan Android OS). *Juima*, 12(1), 111–129.

<https://www.gartner.com/newsroom/id>

- Asri, Y., Suliyanti, W. N., Kuswardani, D., & Fajri, M. (2022). Pelabelan Otomatis Lexicon Vader dan Klasifikasi Naive Bayes dalam menganalisis sentimen data ulasan PLN Mobile. *Petir*, 15(2), 264–275. <https://doi.org/10.33322/petir.v15i2.1733>
- Edyt Daryfayi Putra Daulay, & Asror, I. (2020). Sentimen Analisis pada Ulasan Google Play Store Menggunakan Metode Naïve Bayes. *Sentimen Analisis Pada Ulasan Google Play Store Menggunakan Metode Naïve Bayes*, 7(2355–9365), 11.
- Ihsan, F. (2021). *Implementasi Algoritma Decision Tree untuk Mendeteksi Multi-Label Hate Speech dan Abusive Language pada Twitter Bahasa Indonesia*. 1–98.
- Irawan, A. (2019). Kualitas Pelayanan PT. Pertamina (Persero) Dalam Penyaluran Bahan Bakar Minyak (BBM) di Kabupaten Merauke. *Madani Jurnal Politik Dan Sosial Kemasyarakatan*, 11(2), 152–168. <http://www.e-jurnal.unisda.ac.id/index.php/MADANI/article/download/1605/1003>
- Kusairi, M. M., Agustian, S., Informatika, T., Islam, U., Sultan, N., Kasim, S., & Baru, S. (2022). *SVM Method with FastText Representation Feature for Classification of Twitter Sentiments Regarding the Covid-19 Vaccination Program 1,2*. x(02), 140–150.
- Masturoh, S. (2021). Analisis Sentimen Terhadap E-Wallet Dana Pada Ulasan Google Play Menggunakan Algoritma *K-Nearest Neighbor*. *Jurnal Pilar Nusa Mandiri*, 17(1), 53–58. <https://doi.org/10.33480/pilar.v17i1.2182>
- 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>
- Sahara, S., & Permana, R. A. (2019). Metode K-Nn for Analys Sentiment Review Kids Apps. *JST (Jurnal Sains Dan Teknologi)*, 8(2), 127–137. <https://doi.org/10.23887/jstundiksha.v8i2.21240>
- Saputra, S. A., Rosiyadi, D., Gata, W., & Husain, S. M. (2019). Google Play E-Wallet Sentiment Analysis Using Naive Bayes Algorithm Based on Particle Swarm Optimization. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 3(3), 377–382.
- Wahyudi, B., & Kuswandi, I. (2022). Prediksi Peringkat Aplikasi di Google Play Menggunakan Metode Random Forest. *Jurnal Nasional Teknologi Komputer*, 2(1), 10.