HOAQ: JURNAL TEKNOLOGI INFORMASI, Volume 17 Nomor 2, Desember 2026, p382-391
p-ISSN: 2337-5280, e-ISSN: 26207427
DOI: https://doi.org/10.52972/hoaq.vol17no2

PENGARUH TEKNIK PELABELAN BERDASARKAN *RATING* DAN MANUAL TERHADAP KINERJA MODEL *LOGISTIC REGRESSION* BERBASIS TF-IDF PADA ANALISIS SENTIMEN ULASAN APLIKASI WIKIPEDIA

Kevin Winardi[1], Jeremia Sandy Pratama[2], Delon Setiawan[3], Latius Hermawan[4]

[1-4]Program Studi Informatika, Universitas Katolik Musi Charitas
Jl. Bangau No. 60, Kecamatan Ilir Timur II, Palembang – Sumatera Selatan, Indonesia
Email: [1]kevinwinardi192005@gmail.com, [2]jeremiapratama0@gmail.com, [3]delonsetiawan11@gmail.com, [4]tiuz.hermawan@ukmc.ac.id

ABSTRAK

Wikipedia merupakan salah satu ensiklopedia daring yang menyediakan berbagai informasi dan pengetahuan yang dapat diakses melalui *browser* ataupun menggunakan aplikasi. Sebagai aplikasi yang tersedia di Google Play Store, Wikipedia menerima berbagai ulasan pengguna yang dapat dimanfaatkan untuk mengetahui persepsi dan tingkat kepuasan terhadap aplikasi. Penelitian ini bertujuan untuk menganalisis pengaruh teknik pelabelan sentimen ulasan pada aplikasi Wikipedia. Proses pelabelan sentimen dilakukan menggunakan dua teknik, yaitu pelabelan berdasarkan *rating* pengguna (1-2: negatif, 3: netral, 4-5: positif) dan pelabelan manual. Data yang telah melalui tahap *preprocessing* kemudian diekstraksi menggunakan TF-IDF dan diklasifikasikan menggunakan *Logistic Regression* dengan pembagian data latih dan data uji sebesar 80:20. Hasil penelitian menunjukkan kalau teknik pelabelan memiliki pengaruh terhadap kinerja model *Logistic Regression* berbasis *TF-IDF*. Hasil evaluasi awal menunjukkan nilai *macro average F1-score* sebesar 0,69 pada pelabelan manual lebih tinggi dibandingkan pelabelan berdasarkan *rating* sebesar 0,46. Data yang digunakan memiliki distribusi kelas yang tidak seimbang sehingga dilakukan penyeimbangan data menggunakan *SMOTE* dan optimasi parameter terbaik menggunakan *GridSearchCV*. Namun hasil analisis lanjutan tersebut belum memberikan peningkatan performa yang berarti pada kedua teknik pelabelan. Temuan penelitian ini menunjukkan bahwa pelabelan manual menghasilkan representasi sentimen yang lebih sesuai dengan isi ulasan dibandingkan pelabelan berdasarkan *rating*, sehingga memberikan performa klasifikasi yang lebih baik.

Kata kunci: analisis sentimen, Wikipedia, *TF-IDF*, *Logistic Regression*, klasifikasi

ABSTRACT

Wikipedia is an online encyclopedia providing a wealth of information and knowledge accessible via web browsers or mobile applications. As an app available on the Google Play Store, Wikipedia receives numerous user reviews that can be leveraged to gauge user perceptions and satisfaction levels. This study aims to analyze the impact of sentiment labeling techniques on Wikipedia app reviews. Sentiment labeling was conducted using two methods: labeling based on user ratings (1-2: negative, 3: neutral, 4-5: positive) and manual labeling. After undergoing preprocessing, the data were extracted using TF-IDF and classified using Logistic Regression, with an 80:20 split between training and testing data. The results indicate that the labeling method influences the performance of the TF-IDF-based Logistic Regression model. Initial evaluations showed a macro average F1-score of 0.69 for manual labeling, which was higher than the 0.46 score obtained with rating-based labeling. Due to the imbalanced class distribution of the dataset, data balancing was performed using SMOTE, and optimal parameters were determined using GridSearchCV. However, these further analyses did not yield significant performance improvements for either labeling method. The findings demonstrate that manual labeling produces sentiment representations that better align with review content compared to rating-based labeling, thereby resulting in superior classification performance.

Keywords: sentiment analysis, Wikipedia, TF-IDF, Logistic Regression, classification

1. PENDAHULUAN

Perkembangan teknologi terus mengalami peningkatan di berbagai aspek. Salah satu bentuk teknologi yang terus berkembang dan banyak digunakan di sekitar adalah internet. Penggunaan internet sudah menjadi bagian dari masyarakat Indonesia untuk berbagai kebutuhan, termasuk kebutuhan akan informasi [1]. Dengan adanya internet, informasi semakin mudah diperoleh dengan banyaknya ensiklopedia *online* yang tersedia.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Salah satu ensiklopedia *online* yang sangat populer saat ini adalah Wikipedia. Wikipedia telah menjadi sumber informasi dari berbagai pengguna internet dengan didukung akses keterbukaan, sehingga setiap orang dapat turut serta berkontribusi dalam penulisan Setiap sumber informasi yang ditawarkan di Wikipedia disadur dari berbagai referensi, sehingga sesuai dengan penelitian [2] yang menunjukkan kalau Wikipedia dapat digunakan sebagai pemberhentian pertama pencarian informasi.

Dalam hal kenyamanan pengguna, Wikipedia yang awalnya hanya diakses menggunakan peramban web (*browser*), sekarang dapat diakses dengan menggunakan aplikasi. Dalam hal keberlanjutan pengembangan, penting untuk melakukan evaluasi dari aplikasi yang dibuat. Analisis sentimen dapat digunakan untuk mengumpulkan informasi dari evaluasi masalah tertentu atau program aplikasi [3]. Informasi yang diperoleh dari Google Play Store sebagai tempat pendistribusian aplikasi dapat menjadi keputusan pengembangan dan peningkatan fungsionalitas aplikasi [4].

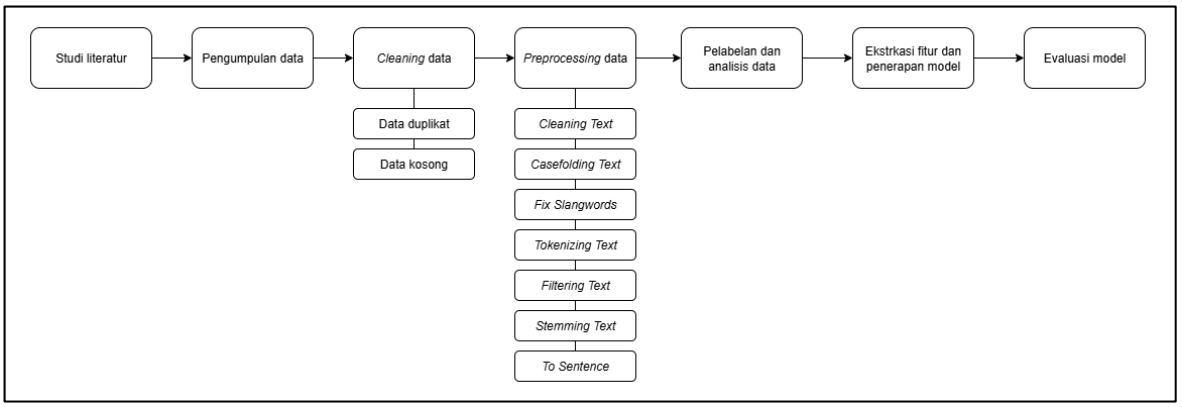
Dalam proses analisis sentimen ulasan, salah satu teknik pelabelan yang sering digunakan adalah menggunakan *rating*. Seperti penelitian [4] yang mengategorikan *rating* 1-2 sebagai negatif, sedangkan *rating* 3-5 sebagai positif. Berbeda dengan penelitian [5] di mana *rating* 1-3 dianggap sebagai sentimen negatif dan *rating* 4-5 dianggap sentimen positif. Selain menggunakan *rating*, terdapat juga proses pelabelan secara manual dengan membaca setiap ulasan dan mengategorikan makna ulasan sesuai dengan cirinya positif, netral, ataupun negatif [6].

Berbagai penelitian sebelumnya terkait analisis sentimen banyak menggunakan algoritma *Term Frequency – Inverse Document Frequency* (*TF-IDF*) untuk mendapatkan bobot setiap kata yang kemudian dikombinasi dengan algoritma klasifikasi lainnya. *Logistic Regression* adalah salah satu model yang sering digunakan karena kemampuannya dalam menangani data yang lebih kompleks seperti analisis sentimen [7]. Dalam hal analisis sentimen beberapa aplikasi *mobile banking,* penggunaan *Logistic Regression* dinilai mampu dalam menganalisis sentimen ulasan pengguna di Google Play Store [3]. Algoritma yang sama juga digunakan oleh peneliti [7] dengan objek penelitian ulasan produk *skincare* di Tokopedia dengan hasil performa akurasi cukup tinggi hingga 92%.

Berdasarkan latar belakang yang telah dipaparkan, penelitian ini difokuskan pada perbandingan kinerja antara pelabelan berdasarkan *rating* dan manual pada analisis sentimen ulasan aplikasi Wikipedia menggunakan algoritma *Logistic Regression* berbasis TF-IDF. Aplikasi Wikipedia dipilih karena memiliki jumlah ulasan pengguna yang cukup banyak dan beragam, sehingga dapat digunakan untuk melakukan analisis sentimen. Selain itu, penelitian ini juga dilakukan karena belum ditemukan kajian yang secara khusus menganalisis sentimen pengguna terhadap aplikasi Wikipedia, sehingga menjadi *research gap* yang perlu dikaji lebih lanjut. Kemudian untuk mengatasi perbedaan pendekatan pelabelan yang ditemukan pada penelitian terdahulu, penelitian ini menerapkan tiga kategori sentimen, yaitu negatif (*rating* 1-2), netral (*rating* 3), dan positif (*rating* 4-5). Melalui pendekatan tersebut, penelitian ini diharapkan dapat memberikan kontribusi ilmiah dalam pengembangan studi analisis sentimen serta menghasilkan informasi yang dapat digunakan sebagai bahan pertimbangan dalam evaluasi dan pengembangan aplikasi berdasarkan ulasan pengguna.

2. METODE PENELITIAN

Penelitian ini dilakukan dengan melewati beberapa tahapan dimulai dari studi literatur, pengumpulan data, *cleaning* data, *preprocessing* data, pelabelan dan analisis data, ekstraksi fitur dan pembuatan model, dan evaluasi model. Setiap tahapan dilakukan dengan mengikuti alur skema seperti yang disajikan pada Gambar 1 berikut ini.



Gambar 1. Alur Penelitian

Studi Literatur

Tahap pertama dilakukan dengan mencari sumber-sumber referensi penelitian yang berkaitan dengan analisis sentimen, *TF-IDF,* dan *Logistic Regression*.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

1. Analisis Sentimen
   Analisis sentimen merupakan bagian dari teknik *data mining* untuk mengolah data teks guna mengidentifikasi pola opini yang berkembang di kalangan pengguna [5]. Dengan analisis sentimen, maka *data miner* dapat mengetahui emosi yang dirasakan oleh pengguna yang dituangkan melalui teks.
2. TF-IDF
   TF-IDF (*Term Frequency-Inverse Document Frequency*) sebuah metode statistik yang digunakan untuk mengukur seberapa penting sebuah kata dalam suatu dokumen di dalam sebuah koleksi atau korpus [7]. Setiap kata akan diterjemahkan ke dalam representasi numerik [8].
3. *Logistic Regression*
   Logistic Regression adalah metode regresi yang digunakan untuk klasifikasi, baik biner maupun multi-kelas [7]. Model ini menghasilkan nilai probabilistik antara 0 dan 1 sehingga bisa dipakai untuk klasifikasi.

Pengumpulan Data

Data dikumpulkan menggunakan metode *scraping* dengan bantuan *library* Python yaitu *google-play-scraper*. Data yang diambil akan disimpan dalam format file CSV (*Comma Separated Value*) dengan hanya mengambil data *review* dan *rating*.

***Cleaning* Data**

Proses ini sebagai langkah awal untuk memastikan model dapat diolah dengan baik pada setiap kata di tahapan selanjutnya. Oleh karena itu, data yang mengandung nilai kosong ataupun duplikat perlu ditangani sebelum masuk ke tahap *preprocessing*.

***Preprocessing* Data**

Preprocessing merupakan tahapan membersihkan dan menyiapkan data untuk diproses [9]. Hal ini dikarenakan data yang diperoleh berupa ulasan memiliki banyak *noise* dan tidak terstruktur [5]. Adapun *preprocessing* yang dilakukan meliputi beberapa tahap, antara lain:

1. *Cleaning Text*
   Pada langkah ini dilakukan beberapa metode pembersihan kata yang tidak memiliki makna khusus dari ulasan yang diperoleh. Langkah yang dilewati antara lain:
   a. Menghapus *mentions* dengan ciri khas simbol "@"
   b. Menghapus *hashtag* dengan ciri khas simbol "#"
   c. Menghapus pranala (*link*) dengan ciri khas diawali "http" ataupun "www"
   d. Menghapus angka
   e. Menghapus karakter bukan alfanumerik
   f. Mengganti baris baru dengan spasi
   g. Menghapus tanda baca dengan bantuan *library NLTK*
   h. Menghapus spasi awal dan akhir kalimat
2. *Casefolding Text*
   Untuk menyelaraskan kata, maka dibuat setiap kata menjadi huruf kecil agar pemaknaan seperti kata "Bagus" dengan "bagus" sama.
3. *Fix Slangwords*
   Slangwords atau disebut sebagai bahasa gaul perlu ditransformasikan ke bentuk bahasa yang baku. Dalam hal ini proses perbaikan kata menggunakan kamus *slangwords* yang tersedia secara publik.
4. *Tokenizing Text*
   Kalimat yang awalnya terdiri dari banyak kata akan dipecah masing-masing menjadi bentuk data token [10]. Pada tahapan ini digunakan fungsi *word_tokenize* dari *library NLTK*.
5. *Filtering Text*
   Tahapan ini melakukan *filtering* dengan menghapus kata-kata ganti yang sedikit memiliki makna [11]. Penerapannya dengan menggunakan fungsi *stopwords* dari *library NLTK*.
6. *Stemming Text*
   Stemming adalah proses normalisasi kata menjadi bentuk kata dasarnya [10]. Misalnya kata "membantu" menjadi "bantu". Proses ini diterapkan dengan penggunaan *StemmerFactory* dari *library Sastrawi*.
7. *To Sentence*
   Proses terakhir yaitu dengan menggabungkan kata-kata yang telah dipisah sebelumnya menjadi kalimat utuh kembali.

Pelabelan dan Analisis Data

Pelabelan dilakukan dengan menggunakan dua teknik, yaitu berdasarkan dari *rating* dan manual. Adapun untuk teknik *rating* dari rentang 1-2 diberi label sebagai negatif, 3 sebagai netral, dan 4-5 sebagai positif. *Rating* 3 dianggap sebagai netral mengingat berada di tengah antara rentang 1-5. Sedangkan, pada teknik manual proses *labeling* dengan melihat makna ulasan dengan aturan data berlabel positif jika memiliki ciri terdapat ungkapan persetujuan, pujian, kebahagiaan serta respon yang baik; berlabel negatif jika memiliki ciri terdapat ungkapan

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

penolakan, amarah, kekecewaan, hujatan, respon yang tidak baik; dan data netral memiliki ciri kalimat dapat mengandung kalimat tanya, atau mengandung dua makna positif dan negatif [6].

Kemudian dari proses pelabelan dilanjutkan ke analisis data untuk mengetahui *insight* dari data yang diperoleh Dalam mengetahui kata-kata yang sering digunakan di ulasan Wikipedia, maka digunakan visualisasi *word cloud*. *Word cloud* merupakan visualisasi yang menampilkan setiap kata dengan ukuran yang semakin membesar sebagai indikator menandakan kata yang sering muncul dalam ulasan [4]. Kemudian analisis kedua yang digunakan adalah analisis berdasarkan *bigram* untuk melihat hubungan dari dua pasangan kata [12].

Ekstraksi Fitur dan Penerapan Model

Data yang telah bersih sebelumnya akan diekstrak menggunakan TF-IDF untuk mendapatkan bobot nilai angka setiap kata. Bobot-bobot nilai tersebut akan dijadikan sebagai data untuk proses pelatihan model *Logistic Regression*.

Evaluasi Model

Model yang telah dilatih perlu diuji performanya dengan menggunakan metrik evaluasi. Beberapa metrik yang umum digunakan pada kasus klasifikasi, yaitu *accuracy*, *precision*, *recall*, dan *F1-score* [13]. Metrik *accuracy* kurang sesuai jika digunakan pada data dengan label yang tidak seimbang karena akan cenderung memiliki nilai tinggi padahal bias terhadap label dominan [14]. Sehingga metrik seperti *precision*, *recall*, dan *F1-score* lebih diutamakan pada kasus dataset dengan label tidak seimbang. Untuk mendapatkan nilai metrik tersebut, maka perlu menggunakan tabel *confusion matrix* [6]. Tabel *confusion matrix* menampilkan hubungan jumlah antara prediksi dengan data sebenarnya. Adapun untuk *confusion matrix* dasar 2 label ditulis seperti pada Tabel 1.

Tabel 1. *Confusion Matrix* 2 Label

		Prediksi	
		Negatif	Positif
Aktual	Negatif	TN	FP
	Positif	FN	TP

Berdasarkan Tabel 1, diperoleh 4 data, antara lain:

1. TN (*True Negative*): model memprediksi negatif dan benar negatif.
2. FP (*False Positive*): model memprediksi positif, padahal sebenarnya negatif.
3. FN (*False Negative*): model memprediksi negatif, padahal sebenarnya positif.
4. TP (*True Positive*): model memprediksi positif dan benar positif.

Setelah mendapat data *confusion matrix*, maka rumus dari metrik evaluasi sebelumnya bisa dihitung dengan menggunakan rumus seperti di bawah ini.

1. *Accuracy*: mengukur seberapa baik model dapat mengklasifikasi data dengan benar.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

2. *Precision*: mengukur rasio prediksi benar positif dibandingkan seluruh data prediksi positif.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3. *Recall*: mengukur rasio prediksi benar positif dibandingkan keseluruhan data sebenarnya positif.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4. *F1-Score*: gabungan dari *precision* dan *recall* untuk mengukur keseimbangan nilai harmonik.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Dalam kasus prediksi 3 kelas, bentuk dari *confusion matrix* hanyalah peningkatan dari ordo matriks 2x2 menjadi 3x3 dengan rumus metrik evaluasi tetap menggunakan konsep sebelumnya. Adapun untuk *confusion matrix* 3 kelas disajikan pada Tabel 2.

Tabel 2. *Confusion Matrix* 3 Kelas

		Prediksi		
		Negatif	Netral	Positif
Aktual	Negatif	TN	FR_1	FP_1
	Netral	FN_1	TR	FP_2
	Positif	FN_2	FR_2	TP

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

3. HASIL DAN PEMBAHASAN

Pengumpulan Data

Data diambil menggunakan *library google-play-scraper* dengan menggunakan parameter *app*_id yaitu "org.wikipedia" sesuai dengan identifikasi unik dari aplikasi Wikipedia di Google Play Store. Selain itu, data yang diambil hanya berfokus pada Bahasa Indonesia (*lang = "id"*) dan asal negara Indonesia (*country = "id"*). Kemudian data yang diambil urut berdasarkan terbaru (*sort = Sort.NEWEST*). Dari hasil *scraping*, diperoleh data awal sebanyak 3.187 ulasan.

Tabel 3. Lima Contoh Ulasan yang Diperoleh dari *Scraping*

Ulasan	*Rating*
aplikasi bermanfaaat	5
info tidak di update	1
aplikasi nya berarti terimakasih 🤝 👍	5
Minta bantuannya .saya lupa passwordnya. Jadi saya gak bisa masuk Wikipedia	3
bagus sih bisa tau apa aja	5

***Cleaning* Data**

Data awal yang diperoleh masih bersifat data kotor di mana ketika dilakukan pengecekan awal terdapat 4 data kosong dan 634 data duplikat. Data kosong diputuskan untuk dihapus karena jumlahnya cenderung sedikit. Data duplikat yang sering ditemukan berupa ulasan singkat seperti "bagus", "ok", "mantap", dll. Data duplikat tersebut meskipun jumlahnya banyak tetap diputuskan untuk dihapus karena cenderung akan menimbulkan bias pada model [5].

Preprocessing Data

Data ulasan yang didapat masih memiliki *noise* dan tidak terstruktur. *Preprocessing* dilakukan secara bertahap seperti penjelasan pada bagian Metode Penelitian poin *Preprocessing* Data. Adapun untuk perbandingan hasilnya disajikan pada Tabel 4 berikut ini.

Tabel 4. Perbandingan Data Sebelum dan Sesudah *Preprocessing*

Sebelum *Preprocessing*	Setelah *Preprocessing*
aplikasi bermanfaaat	aplikasi manfaat
info tidak di update	informasi tidak update
aplikasi nya berarti terimakasih 🤝 👍	aplikasi arti terimakasih
Minta bantuannya. saya lupa passwordnya. Jadi saya gak bisa masuk Wikipedia	bantu lupa passwordnya tidak bisa masuk wikipedia
bagus sih bisa tau apa saja	bagus bisa tahu

Setelah proses *preprocessing* tersebut, muncul nilai kosong dan duplikat baru pada beberapa baris data. Hal ini wajar terjadi akibat dari *preprocessing*, seperti kasus kosong terjadi ketika ulasan dengan banyak kata *stopwords* dihapus sehingga menyisakan teks kosong. Pada kasus duplikat seperti ulasan "Bagus" dan "bagus" awalnya dianggap berbeda, namun ketika selesai *preprocessing* menjadi "bagus" pada keduanya sehingga dianggap duplikat. Keputusan diambil dengan menghapus data kosong dan duplikat agar model dapat belajar dengan data bersih dari kata yang ada. Sehingga setelah proses ini data bersih tersisa hanya 1.993 ulasan.

Pelabelan dan Analisis Data

a. Pelabelan

Proses pelabelan dibagi menjadi dua bagian, bagian pertama mengikuti aturan rentang *rating* 1-2 sebagai negatif, 3 sebagai netral, dan 4-5 sebagai positif. Bagian kedua menggunakan pelabelan secara manual dengan melihat makna ulasan. Hasil yang diperoleh menunjukkan perbedaan jumlah cukup signifikan pada kedua teknik.

Tabel 5. Perbandingan Jumlah Label

Label	Jumlah pada Pelabelan Berdasarkan *Rating*	Jumlah pada Pelabelan Manual
Positif	1.772	1551
Netral	121	315
Negatif	100	127

Ditemukan beberapa kasus seperti data ulasan yang sebenarnya mengandung kata negatif tetapi pengguna memberi *rating* tinggi sehingga pada pelabelan berdasarkan *rating* masuk kategori positif. Hal yang sama juga terjadi di mana ulasan yang sifatnya cenderung kalimat tanya yang seharusnya netral namun diberi *rating* tinggi

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

sehingga masuk kategori positif. Hal ini menjadi faktor yang terlihat pada perbedaan kedua teknik, di mana jumlah label positif berdasarkan pelabelan berdasarkan *rating* jauh lebih tinggi dari pada pelabelan secara manual.

Tabel 6. Contoh Perbandingan Pelabelan Berdasarkan *Rating* dan Manual

Ulasan Setelah *Preprocessing*	Pelabelan Berdasarkan *Rating*	Pelabelan Manual
bagus	positif	positif
guna data tidak	positif	netral
offline tidak bisa	positif	negatif
tampil aneh	positif	negatif
mantap	positif	positif

b. Analisis *Word Cloud*

Pada sentimen negatif, kata "tidak" merupakan salah satu kata yang paling sering muncul pada kedua teknik pelabelan. Tingginya frekuensi kata tersebut menunjukkan bahwa sebagian besar ulasan negatif mengandung bentuk negasi yang berkaitan dengan keluhan atau ketidakpuasan pengguna terhadap aplikasi. Selain itu, kata "bisa" tampak lebih dominan pada hasil pelabelan manual dibandingkan pelabelan berdasarkan *rating*. Kondisi ini mengindikasikan bahwa pelabelan manual lebih banyak mengelompokkan ulasan yang kemungkinan mengandung gabungan kata seperti "tidak bisa" ke dalam kategori sentimen negatif, sehingga lebih mampu merepresentasikan keluhan pengguna berdasarkan isi ulasan dibandingkan hanya berdasarkan nilai *rating*.

Pada sentimen netral di kedua teknik cenderung bercampur kata dengan indikasi positif dan negatif. Hal ini wajar ditemukan, mengingat posisi netral berada di tengah-tengah yang di mana pengguna cenderung memilih netral sebagai nilai yang tidak terlalu condong ke sangat positif ataupun sangat negatif. Terdapat kata yang sering muncul seperti “bagus” yang di mana muncul juga pada sentimen positif sebelumnya. Kemudian terdapat kata “bisa” yang sebenarnya dapat memunculkan dua kemungkinan “bisa digunakan” ataupun “tidak bisa”.

Pada sentimen positif, baik menggunakan teknik pelabelan berdasarkan *rating* ataupun pelabelan manual menampilkan kata yang identik dengan dominasi kata “bagus”, “bisa”, “bantu”, “wawas”, “tahu”, “ajar”, dll. Hal ini menunjukkan bahwa aplikasi Wikipedia sangat bagus membantu dalam menambah wawasan / ilmu.

Tabel 7. Perbandingan *Word Cloud*

Label	Pelabelan Berdasarkan *Rating*	Pelabelan Manual
Negatif		
Netral		
Positif		

c. Analisis *Bigram*

Pada sentimen negatif, kedua teknik sama-sama menunjukkan bahwa *bigram* "tidak bisa" merupakan pasangan kata yang paling sering muncul, dengan frekuensi 26 pada pelabelan berdasarkan *rating* dan meningkat menjadi 32 pada pelabelan manual. Hal ini menunjukkan bahwa sebagian besar ulasan negatif berkaitan dengan kendala penggunaan aplikasi. *Bigram* lain seperti "aplikasi tidak", "bisa edit", "log in", "bisa login", dan "alamat ip" juga mengindikasikan adanya keluhan mengenai fitur maupun masalah teknis.

Pada sentimen netral, terlihat adanya perbedaan pola *bigram* antara kedua teknik pelabelan. Pelabelan berdasarkan *rating* didominasi oleh *bigram* "tambah wawas", "tambah bintang", "tidak bisa", "log out", dan "bintang iya". Sebaliknya, pelabelan manual lebih banyak menghasilkan *bigram* "kasih bintang", "kode undang", "tambah bintang", "tambah fitur", dan "bahasa inggris". Perbedaan ini menunjukkan bahwa pelabelan berdasarkan *rating* masih mengelompokkan beberapa ulasan bernada positif, seperti "tambah wawas", ke dalam kategori netral. Sementara itu, pelabelan manual cenderung menempatkan ulasan yang bersifat informatif atau berupa saran,

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

misalnya "tambah fitur" dan "kode undang" sebagai sentimen netral. Dengan demikian, pelabelan manual tampak lebih mampu memisahkan ulasan yang benar-benar netral dari ulasan yang mengandung apresiasi.

Pada sentimen positif, kedua teknik menghasilkan *bigram* yang hampir identik, yaitu "terima kasih", "tambah wawas", "aplikasi bagus", "tambah tahu", dan "ilmu tahu". *Bigram* "terima kasih" merupakan pasangan kata yang paling dominan dengan frekuensi 82 pada pelabelan berdasarkan *rating* dan 79 pada pelabelan manual. Hal ini menunjukkan bahwa pengguna yang memberikan ulasan positif umumnya mengungkapkan rasa terima kasih atas manfaat aplikasi.

Tabel 8. Analisis Perbandingan Bigram

Label	*Top 5 Bigram* Pelabelan Berdasarkan *Rating*	Frekuensi	*Top 5 Bigram* Pelabelan Manual	Frekuensi
Negatif	tidak bisa	26	tidak bisa	32
	bisa edit	6	aplikasi tidak	7
	aplikasi tidak	6	bisa edit	6
	alamat ip	4	log in	5
	kasih bintang	4	bisa login	4
Netral	tambah wawas	5	kasih bintang	4
	tambah bintang	4	kode undang	4
	tidak bisa	3	tambah bintang	4
	log out	3	tambah fitur	4
	bintang iya	3	bahasa inggris	3
Positif	terima kasih	82	terima kasih	79
	tambah wawas	75	tambah wawas	78
	aplikasi bagus	54	aplikasi bagus	53
	tambah tahu	45	tambah tahu	45
	ilmu tahu	38	ilmu tahu	37

Ekstraksi Fitur dan Penerapan Model

Sebelum masuk ke tahap ekstraksi fitur, dataset dibagi menjadi dua bagian yaitu data latih dan data uji dengan rasio 80:20, di mana 80% data digunakan sebagai data latih sedangkan 20% sisanya digunakan sebagai data uji. Proses pembagian menggunakan fungsi *train_test_split* dari *library scikit-learn* dengan parameter bantuan *stratify* untuk membagi rata kelas yang tidak seimbang.

Pada tahap ekstraksi fitur, digunakan algoritma TF-IDF untuk mengubah data teks menjadi representasi numerik berupa bobot kata. Bobot hasil ekstraksi fitur kemudian digunakan sebagai masukan pada algoritma *Logistic Regression* untuk proses pelatihan klasifikasi model. Adapun, parameter yang digunakan pada algoritma TF-IDF dan *Logistic Regression* disajikan pada Tabel 9 berikut ini.

Tabel 9. Parameter yang Digunakan pada Algoritma

Algoritma	Parameter	Nilai	Keterangan
TF-IDF	*max_features*	5.000	Jumlah maksimum fitur kata yang digunakan
	min_df	1	Kata yang minimal muncul pada 1 ulasan
	max_df	0,8	Mengabaikan kata yang muncul lebih dari 80% ulasan
	ngram_range	(1,2)	Menangkap makna kelompok kata sehingga kata negasi seperti “tidak bagus” bisa dibedakan dengan “bagus”
Logistic Regression	*C*	10	Besaran kekuatan regulasi di mana semakin besar nilainya berarti semakin kecil kekuatan regulasi
	max_iter	1.000	Jumlah iterasi maksimum
	random_state	42	Memastikan hasil reproduksibilitas yang sama setiap kali dijalankan
	class_weight	“balanced”	Menyesuaikan bobot kelas yang tidak seimbang

Evaluasi

Berdasarkan pelatihan model di proses sebelumnya, didapat hasil *confusion matrix* antara label prediksi model dengan label aktual. Perbandingan *confusion matrix* antara kedua teknik pelabelan disajikan dalam Tabel 10. Berdasarkan hasil *confusion matrix*, model dengan pelabelan manual menunjukkan jumlah prediksi benar yang lebih tinggi dibandingkan pelabelan berdasarkan *rating*, terutama pada kelas netral dan negatif. Hal ini terlihat dari meningkatnya jumlah data yang berhasil diklasifikasikan dengan benar pada kelas netral dari 3 menjadi 47, serta pada kelas negatif dari 8 menjadi 15. Sebaliknya, pelabelan berdasarkan *rating* masih banyak mengklasifikasikan data negatif dan netral sebagai kelas positif.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Tabel 10. Perbandingan *Confusion Matrix* Awal

		Prediksi Model					
		Pelabelan Berdasarkan *Rating*			Pelabelan Manual		
		Negatif	Netral	Positif	Negatif	Netral	Positif
Label Aktual	Negatif	8	1	15	15	1	9
	Netral	1	3	16	2	47	14
	Positif	16	16	323	6	49	256

Evaluasi selanjutnya dilakukan pada model *Logistic Regression* menggunakan metrik *precision*, *recall*, dan *F1-score* untuk setiap kelas sentimen. Hasil evaluasi pada Tabel 11 menunjukkan bahwa model yang menggunakan pelabelan manual memiliki performa yang lebih baik dibandingkan model yang menggunakan pelabelan berdasarkan *rating*. Hal ini terlihat dari nilai *macro average F1-score* sebesar 0,69 pada pelabelan manual, yang lebih tinggi dibandingkan pelabelan berdasarkan *rating* sebesar 0,46. Karena *macro average* menghitung rata-rata performa setiap kelas dengan bobot yang sama, hasil tersebut menunjukkan bahwa model dengan pelabelan manual mampu mengklasifikasikan seluruh kelas sentimen secara lebih seimbang, terutama pada kelas netral dan negatif yang memiliki nilai *F1-score* lebih tinggi dibandingkan pelabelan berdasarkan *rating*.

Tabel 11. Perbandingan Metrik Evaluasi Awal

Label/Metode Evaluasi	Pelabelan Berdasarkan *Rating*			Pelabelan Manual		
	Precision	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*
Negatif	0,32	0,33	0,33	0,65	0,60	0,62
Netral	0,15	0,15	0,15	0,48	0,75	0,59
Positif	0,91	0,91	0,91	0,92	0,82	0,87
Macro Average	0,46	0,46	0,46	0,68	0,72	0,69

Berdasarkan hasil evaluasi sebelumnya, model masih menunjukkan performa yang kurang optimal terutama dalam mengklasifikasikan kelas netral dan negatif. Hal ini terlihat dari nilai *precision* dan *recall* yang rendah pada kedua kelas minoritas, terlebih lagi pada teknik pelabelan berdasarkan *rating*. Oleh karena itu, dilakukan percobaan lanjutan dengan menerapkan teknik *oversampling* dan *hyperparameter tuning* untuk meningkatkan performa model. Teknik *oversampling* menggunakan *SMOTE* (*Synthetic Minority Over-sampling Technique)* bertujuan untuk mengurangi ketidakseimbangan distribusi kelas pada data latih dengan menghasilkan sampel sintetis pada kelas minoritas sehingga model dapat mempelajari karakteristik setiap kelas secara lebih baik [7]. Selain itu, *hyperparameter tuning* menggunakan *GridSearchCV* diterapkan untuk memperoleh kombinasi parameter yang optimal sehingga diharapkan dapat meningkatkan kemampuan generalisasi model [15]. Pada proses *GridSearchCV* digunakan *5-fold cross-validation* untuk mengevaluasi setiap kombinasi *hyperparameter* dan *scoring* terbaik menggunakan *macro F1-score*. *Hyperparameter* beserta nilai yang diuji disajikan pada Tabel 12.

Tabel 12. Parameter yang Diuji pada *GridSearchCV*

Algoritma	*HyperParameter*	Nilai yang Diuji
TF-IDF	*max_df*	0,6; 0,7; 0,8
	max_features	3.000; 5.000; 7.000
	ngram_range	(1,2); (1,3); (2,3)
Logistic Regression	*C*	0,1; 1; 10
	max_iter	1.000; 3.000; 5.000

Berdasarkan Tabel 13, *GridSearchCV* menghasilkan kombinasi *hyperparameter* yang berbeda untuk setiap teknik pelabelan. Perbedaan tersebut menunjukkan bahwa konfigurasi model yang optimal dipengaruhi oleh karakteristik data pada masing-masing teknik pelabelan. Pada tahap pembobotan menggunakan *TF-IDF*, nilai *max_df*, *max_features*, dan *ngram_range* yang terpilih berbeda, sehingga menunjukkan bahwa representasi fitur terbaik tidak selalu sama untuk setiap dataset. Demikian pula pada algoritma *Logistic Regression*, nilai *C* dan *max_iter* yang diperoleh merupakan kombinasi yang menghasilkan performa terbaik.

Tabel 13. Perbandingan Hasil *HyperParameter* yang Optimal

Algoritma	*HyperParameter*	Nilai Pelabelan Berdasarkan *Rating*	Nilai Pelabelan Manual
TF-IDF	*max_df*	0,6	0,6
	max_features	7.000	5.000
	ngram_range	(1,2)	(1,3)
Logistic Regression	*C*	0,1	1
	max_iter	1.000	1.000

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Setelah menerapkan *SMOTE* dan *GridSearchCV*, dilakukan evaluasi kembali dengan melihat perbandingan *confusion matrix* pada kedua teknik pelabelan. Berdasarkan Tabel 14, model dengan pelabelan manual masih menunjukkan hasil klasifikasi yang lebih baik dibandingkan pelabelan berdasarkan *rating*. Pada kelas negatif, jumlah prediksi yang benar meningkat dari 9 pada pelabelan berdasarkan *rating* menjadi 17 pada pelabelan manual. Demikian pula pada kelas netral, jumlah prediksi yang benar meningkat secara signifikan dari 3 menjadi 48. Sementara itu, pada kelas positif, pelabelan berdasarkan *rating* menghasilkan prediksi benar yang lebih banyak (316) dibandingkan pelabelan manual (243). Namun, penurunan tersebut diikuti oleh peningkatan kemampuan model dalam mengenali kelas negatif dan netral, sehingga performa klasifikasi menjadi lebih seimbang.

Tabel 14. Perbandingan *Confusion Matrix* Setelah *SMOTE* dan *GridSearchCV*

		Prediksi Model					
		Pelabelan Berdasarkan *Rating*			Pelabelan Manual		
		Negatif	Netral	Positif	Negatif	Netral	Positif
Label Aktual	Negatif	9	0	15	17	3	5
	Netral	5	3	12	3	48	12
	Positif	20	19	316	10	58	243

Dari sisi metrik *precision*, *recall*, dan *F1-score* yang dirangkum pada Tabel 15, menunjukkan bahwa penerapan kedua teknik tersebut belum memberikan peningkatan performa secara keseluruhan dibandingkan evaluasi awal tanpa *SMOTE* dan *GridSearchCV*. Hal ini terlihat dari nilai *macro F1-score* pada pelabelan berdasarkan *rating* yang berubah dari 0,46 menjadi 0,45, sedangkan pada pelabelan manual mengalami sedikit penurunan dari 0,69 menjadi 0,68. Meskipun demikian, pelabelan manual tetap memiliki kinerja yang lebih baik dibandingkan pelabelan berdasarkan *rating*.

Tabel 15. Perbandingan Metrik Evaluasi Setelah *SMOTE* dan *GridSearchCV*

Label/Metode Evaluasi	Pelabelan Berdasarkan *Rating*			Pelabelan Manual		
	Precision	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*
Negatif	0,26	0,38	0,31	0,57	0,68	0,62
Netral	0,14	0,15	0,14	0,44	0,76	0,56
Positif	0,92	0,89	0,91	0,93	0,78	0,85
Macro Average	0,44	0,47	0,45	0,65	0,74	0,68

4. SIMPULAN

Berdasarkan hasil penelitian mengenai perbandingan teknik pelabelan berdasarkan *rating* dan pelabelan manual pada ulasan aplikasi Wikipedia, dapat disimpulkan bahwa teknik pelabelan memengaruhi kinerja model TF-IDF dengan *Logistic Regression*. Hasil penelitian menunjukkan bahwa pelabelan manual menghasilkan performa yang lebih baik dibandingkan pelabelan berdasarkan *rating*. Hal ini ditandai dengan nilai *macro average F1-score* sebesar 0,69, sedangkan pelabelan berdasarkan *rating* memperoleh nilai 0,46. Hasil tersebut menunjukkan bahwa pelabelan manual mampu menghasilkan representasi sentimen yang lebih sesuai dengan isi ulasan sehingga meningkatkan kemampuan model dalam mengklasifikasikan setiap kelas sentimen. Selain itu, penerapan *SMOTE* dan *GridSearchCV* belum memberikan peningkatan performa yang berarti pada kedua teknik pelabelan. Oleh karena itu, penelitian selanjutnya diharapkan dapat berfokus pada peningkatan kualitas data, penerapan teknik penyeimbangan kelas yang lebih efektif, serta penggunaan model klasifikasi yang lebih canggih agar performa klasifikasi pada seluruh kategori sentimen dapat ditingkatkan.

DAFTAR PUSTAKA

[1] H. C. K. Sari, "Analisis Terjemahan Profil Perusahaan Asing di Wikipedia Indonesia," *KREDO : Jurnal Ilmiah Bahasa dan Sastra*, vol. 4, no. 1, pp. 184–200, Oct. 2020, doi: 10.24176/kredo.v4i1.4794.

[2] I. Y. N. Hizqiyah, A. Widodo, and S. Sriyati, "Pembelajaran Abad 21 dengan menggunakan Wikipedia sebagai Sumber Informasi untuk Meningkatkan Kemampuan Pemecahan Masalah," *Biosfer : Jurnal Biologi dan Pendidikan Biologi*, vol. 7, no. 1, pp. 1–12, Jun. 2022, doi: 10.23969/biosfer.v7i1.5607.

[3] A. Aryanusa, "Evaluasi Kinerja Logistic Regression dan Multinomial Naïve Bayes Dalam Klasifikasi Sentimen Mobile Banking," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, Oct. 2025, doi: 10.23960/jitet.v13i3S1.7687.

[4] L. Rhomaningtias, A. Khairunisa, S. S. M. Wara, and K. M. Hindrayani, "Analisis Sentimen Ulasan Aplikasi Smile Indonesia Menggunakan Metode Naive Bayes dan Support Vector Machine (SVM)," *HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi*, vol. 16, no. 1, pp. 79–91, May 2025, doi: 10.52972/hoaq.vol16no1.p79-91.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

[5] J. M. Manampiring, J. Halawa, J. Zai, I. Kurniawati, R. Fahlapi, and W. Bismi, "Analisis Sentimen Ulasan Aplikasi X Pada Google Play Store Menggunakan Komparasi SVM, KNN, Dan Naive Bayes," *Djtechno: Jurnal Teknologi Informasi*, vol. 7, no. 1, pp. 1–13, Mar. 2026, doi: 10.46576/djtechno.v7i1.8095.

[6] Regina, T. H. Saragih, and D. Kartini, "Analisis Sentimen Brand Ambassador BTS Terhadap Tokopedia Menggunakan Klasifikasi Bayesian Network Dengan Ekstraksi Fitur TF-IDF," *Jurnal Informatika Polinema*, vol. 9, no. 4, pp. 383–390, Aug. 2023, doi: 10.33795/jip.v9i4.1333.

[7] I. Dikusuma and Y. Widjaja, "Perbandingan Naïve Bayes dan Logistic Regression untuk Analisis Sentimen Ulasan Produk Skincare di Tokopedia," *Jurnal Informatika Dan Tekonologi Komputer (JITEK)*, vol. 5, no. 3, pp. 242–251, Nov. 2025, doi: 10.55606/jitek.v5i3.8420.

[8] W. A. Ma'arif, S. Sarwido, and T. Tamrin, "Analisis Sentimen Terhadap Ulasan Game Mobile Legend di Playstore menggunakan Algoritma Logistic Regression," *JUKI : Jurnal Komputer dan Informatika*, vol. 8, no. 1, pp. 43–49, May 2026, doi: 10.53842/juki.v8i1.2319.

[9] F. R. Pratama and Dahlan, "Analisis Sentimen Opini Netizen Di Instagram Terhadap Berita Sepakbola Liga Indonesia Menggunakan Algoritma Logistic Regression," Jurnal Informatika dan Teknik Elektro Terapan, vol. 13, no. 3, pp. 1338–1345, Jul. 2025, doi: 10.23960/jitet.v13i3.7269.

[10] F. S. Widyastuti and E. Mailoa, "Analisis Sentimen Ulasan Konsumen Menggunakan Algoritma TF-IDF Untuk Mengetahui Tingkat Kepuasan Pelanggan (Studi Kasus : Gunthem Premium Coffee)," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 16, no. 2, pp. 8–16, Sep. 2025, doi: 10.51903/jtikp.v16i2.1010.

[11] H. S. Mulyono and U. Saprudin, "Efektivitas Logistic Regression dalam Analisis Sentimen Berbahasa Indonesia pada Komentar YouTube tentang Isu Ketenagakerjaan," Jurnal Indonesia : Manajemen Informatika dan Komunikasi, vol. 6, no. 3, pp. 1547–1555, Sep. 2025, doi: 10.63447/jimik.v6i3.1481.

[12] R. Nurhidayat and K. E. Dewi, "Penerapan Algoritma K-Nearest Neighbor Dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek," *Komputa : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 1, pp. 91–100, May 2023, doi: 10.34010/komputa.v12i1.9458.

[13] B. Ramadhani and R. R. Suryono, "Komparasi Algoritma Naïve Bayes dan Logistic Regression Untuk Analisis Sentimen Metaverse," *Jurnal Media Informatika Budidarma*, vol. 8, no. 2, p. 714, Apr. 2024, doi: 10.30865/mib.v8i2.7458.

[14] A. N. Putro and M. A. Muslim, "Penanganan Ketidakseimbangan Data Ekstrim pada Sistem Prediksi," *Techno.Com*, vol. 24, no. 4, pp. 1359–1367, Nov. 2025, doi: 10.62411/tc.v24i4.15005.

[15] A. Shiddicky and S. Agustian, "Analisis Sentimen Masyarakat Terhadap Kebijakan Vaksinasi Covid-19 pada Media Sosial Twitter menggunakan Metode Logistic Regression," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 2, pp. 99–106, Aug. 2022, doi: 10.37859/coscitech.v3i2.3836.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi