
Analisis Sentimen Multi-Kelas Untuk Film Berbasis Teks Ulasan Menggunakan Model Regresi Logistik

Anasthasya Averina¹, Helen Hadi², Joko Siswanto^{3*}

^{1,2,3} Program Studi Teknik Informatika, Universitas Surabaya, Jawa Timur
Email: ¹s160418083@student.ubaya.ac.id, ²s160418084@student.ubaya.ac.id,
^{3*}joko_siswanto@staff.ubaya.ac.id

(Naskah masuk: 6 Mar 2022, direvisi: 18 Mei 2022, 17 Jun 2022, diterima: 20 Jun 2022)

Abstrak

Pengutaraan pendapat atau pengutaraan pemikiran secara sukarela terhadap suatu film pada situs pengulas film merupakan hal yang sering dilakukan oleh pengguna. Beberapa pengguna kadang-kadang memberikan ulasan yang ambigu terhadap sebuah film, yaitu dengan memberikan komentar yang buruk tetapi memberikan *rating* yang baik atau sebaliknya. Hal ini dapat berpengaruh pada citra film tersebut. Maka dari itu, diperlukan sistem yang dapat memprediksi *rating* agar sesuai dengan komentar yang diberikan atau sistem membenaran *rating*. Penelitian ini bertujuan untuk memprediksi *rating* suatu film berdasarkan ulasan yang diberikan oleh pengguna menggunakan model Regresi Logistik. *Dataset* yang digunakan pada penelitian ini adalah data ulasan 10 film yang berbeda dari *Mendeley Data*. Tahap pra-pemrosesan dilakukan dengan penghapusan kata umum, tanda baca, pengurangan dimensi, dan pengekstrakan ciri dari teks ulasan menggunakan *library scikit-learn*. Dengan 80% data sebagai *training* dan sisanya digunakan untuk *testing*, hasil perhitungan akurasi prediksi 10 kelas *rating* yang didapatkan dari *feature extraction CountVectorizer* adalah 36% dan *TfidfVectorizer* sebesar 32%. Sedangkan hasil dari perhitungan akurasi prediksi 2 *class sentiment*, didapatkan hasil tertinggi sebesar 83% oleh *feature extraction CountVectorizer* dan *feature extraction TfidfVectorizer* sebesar 76%.

Kata Kunci: Ulasan, Prediksi, *Sentiment Analysis*, Regresi Logistik

Multi-Class Sentiment Analysis for Movie Based on Review Text Using Logistic Regression Model

Abstract

It is normal to express opinions or expressing thoughts on a film review site. However, a few reviews seem ambiguous because it can have bad comments with good ratings or vice versa which can affect the image of the film. Therefore, we need a system that can predict the rating to match the sentiment which the comments give or a comment rating justification system. This study was conducted to predict the appropriate rating or sentiment for a review using the Logistic Regression method. The data used is text reviews from 10 different films, retrieved from Mendeley Data. The preprocessing stage is done by removing common words, punctuation marks, dimensionality reduction, and text-feature extraction using the scikit-learn library. With 80% of the data as training and the rest used for testing, the results of the prediction accuracy calculation of 10 class ratings classification obtained from the Count Vectorizer is 36% and TfidfVectorizer feature extraction is 32%. While the results of the prediction accuracy calculation of 2 sentiment classes were at 83% using CountVectorizer feature extraction and 76% using TfidfVectorizer feature extraction.

Keywords: Review, Prediction, Sentiment, Logistic Regression.

I. PENDAHULUAN

Pendapat merupakan sebuah pemikiran akan suatu hal [1]. Mengutarakan pendapat merupakan hal yang wajar pada negara demokrasi, apalagi mengutarakan pendapat untuk film. Dengan adanya pengutaraan pendapat atau *review* mengenai film, jumlah penonton suatu film dapat meningkat dan memberikan keuntungan maupun citra yang baik bagi produsen film.

Pada situs pengulas film, umumnya pengunjung maupun pengguna situs dipersilahkan untuk memberi komentar dan *rating* terhadap suatu film. Ulasan tersebut nantinya akan dipajang di halaman situs film tersebut dan dapat dilihat secara publik sebagai bahan pertimbangan bagus atau tidaknya film tersebut. Biasanya, bentuk komentar yang ditulis tentu saja dalam bentuk teks dan *rating* dalam bentuk bintang dengan skala 1-5 atau dalam kasus ini digunakan skala 1-10.

Pemberian ulasan dilakukan secara sukarela dan bebas oleh pengunjung situs. Tetapi, kebebasan tersebut kadang dapat memberikan data yang kurang sesuai. Misalnya, pengunjung memberi komentar baik, tetapi memberikan *rating* jelek atau sebaliknya. Hal tersebut membuat ulasan menjadi ambigu dan kurang relevan antara komentar dan *rating* yang diberikan. Maka dari itu, diperlukan sistem yang dapat memprediksi *rating* agar sesuai dengan komentar yang diberikan.

Pada penelitian ini dilakukan percobaan untuk memprediksi *rating* yang sesuai untuk suatu ulasan menggunakan Regresi Logistik [2]. Regresi Logistik merupakan model yang digunakan untuk memprediksi hasil biner ($y \in \{0,1\}$) berdasarkan nilai dari input. Metode ini dapat digunakan dalam memecahkan masalah klasifikasi *multi-class*, seperti pada penelitian ini yaitu mengklasifikasikan *rating* suatu komentar [3].

Penelitian serupa telah dilakukan oleh Asghar menggunakan *Yelp Dataset* (<https://www.yelp.com/dataset>). [4]. Pada tahap pra-pemrosesan, penelitian tersebut menggunakan *library* standar *python* untuk menghapus kapitalisasi, kata umum, dan tanda baca. Penelitian tersebut menggunakan 16 model yang berbeda dengan kombinasi empat metode ekstraksi, yaitu *unigrams*, *bigrams*, *trigrams*, dan *latent semantic indexing*. Selanjutnya menggunakan Regresi Logistik, *Naive Bayes*, *Perceptrons*, dan *Linear Support Vector Machine* untuk proses klasifikasi. Pada penelitian ini diperoleh hasil RMSE 1.05 dan akurasi 56% untuk *Linear Support Vector Machine* serta RMSE 0.92 dan akurasi 54% untuk Regresi Logistik. Dari hasil ini dapat disimpulkan bahwa prestasi metode klasifikasi yang diusulkan masih kurang baik [4].

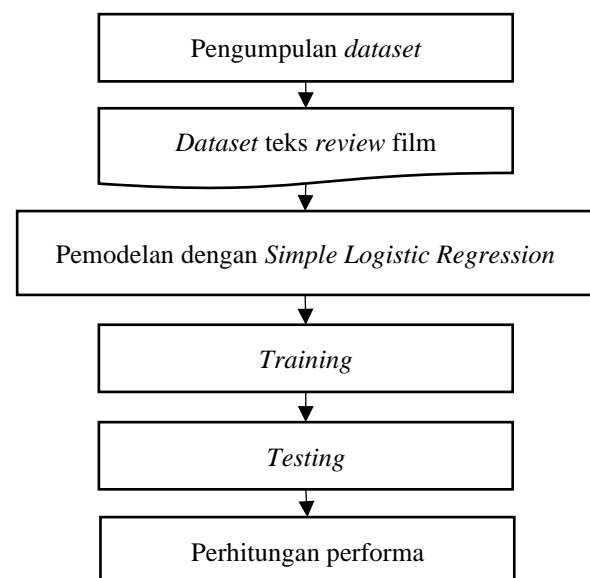
Umer et al. [5] juga melakukan penelitian serupa untuk memprediksi *rating Google Apps* menggunakan *text features* dan *ensemble learning*. Dataset pada penelitian tersebut diambil dari hasil *scraping* pada *Google Play Store* menggunakan situs *scraper BeautifulSoup*. Data yang diambil tidak lebih lama dari tahun 2014 agar tidak melampaui lima tahun dan berjumlah minimal 4000 ulasan. Pra-pemrosesan dilakukan dengan melakukan *stemming*, pengubahan ke huruf kecil, menghilangkan tanda baca, menghapus kata umum, dan memisahkan kata yang tidak memberikan berat tambahan.

Bobot masing-masing kata dalam suatu kalimat diperoleh dengan menggunakan metode penambahan berat TF-IDF (*Term frequency - Inverse Document Frequency*). Peneliti menggunakan kombinasi dari beberapa metode klasifikasi, yaitu *Random Forest*, *AdaBoost*, *Gradient Boosting*, *Extreme Gradient Boosting*, dan *The Extra-Trees Classifier* untuk memprediksi *rating* numerik. Kesimpulan dari hasil *testing* menggunakan analisis *TextBlob* adalah 24.72% pengguna *Google Play Store* memberikan *rating* yang kurang baik, selain itu dari hasil penelitian juga menunjukkan bahwa pengguna memberi *rating* yang tidak sesuai dengan ulasan yang diberikan. *Rating* yang dihasilkan dari prediksi melalui ulasan cenderung lebih tinggi dibandingkan *rating* yang diberikan langsung oleh pengguna. [5].

Dengan demikian, tujuan dari penelitian ini adalah menguji coba model Regresi Logistik dalam mengklasifikasikan *rating* film berdasarkan suatu komentar ke dalam salah satu nilai antara 1-10. Untuk pengembangan, penelitian ini dapat menjadi dasar pembuatan sistem pemeriksaan untuk mengubah nilai *rating* menjadi lebih sesuai dengan ulasan yang diberikan. Selain itu juga dapat digunakan sebagai upaya pencegahan terhadap pengulas agar tidak memberikan ulasan yang tidak sesuai dengan *rating* yang diberikan, sehingga ulasan yang bersifat ambigu terhadap suatu film dapat berkurang, dengan cara pengguna hanya memberikan ulasan dan *rating* akan ditentukan secara otomatis oleh sistem berdasarkan ulasan yang diberikan.

II. METODOLOGI PENELITIAN

Terdapat empat tahapan yang dilakukan dalam penelitian ini. Tahapan-tahapan tersebut meliputi pengumpulan *dataset*, pra-pemrosesan, pembentukan model dan *training-testing*, serta perhitungan performa. Pada Gambar 1, disajikan diagram blok alur proses penelitian secara lebih detail.



Gambar 1. Diagram Blok Metodologi Penelitian

1. Pengumpulan Dataset

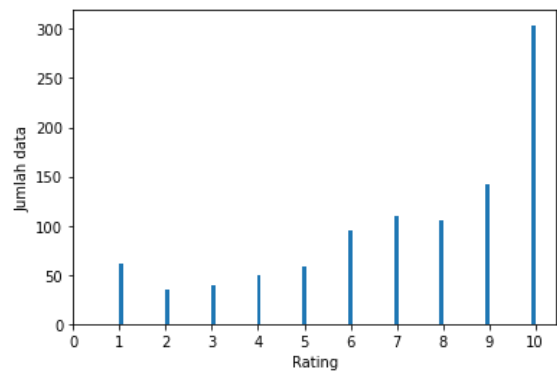
Dataset yang digunakan pada penelitian ini diperoleh dari Mendeley Data [6]. Dataset berisi kumpulan ulasan dari 10 film berbeda, dengan masing-masing film memiliki 100 ulasan. Sumber ulasan berasal dari website *imdb.com*. Data-data ulasan diambil secara acak, namun memperhatikan rating ulasan yang berkisar sesuai rata-rata *user-weighted vote* sehingga data ulasan yang diambil dapat dikatakan sebagai ulasan-ulasan yang mewakili keseluruhan. Pada Tabel 1, disajikan contoh isi dataset yang digunakan pada penelitian ini.

Tabel 1. Dataset Teks Ulasan dan Rating

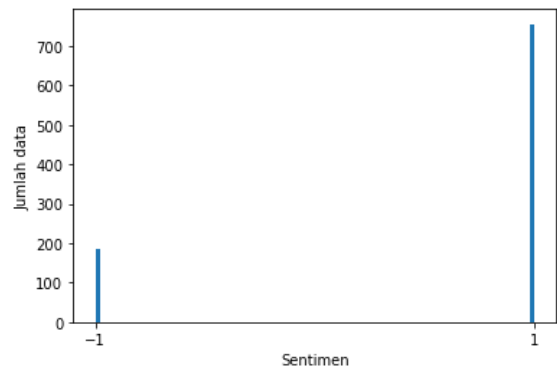
Text	Rating
<i>2012 (2009) is a science fiction film disaster that was released on 13 November 2009...</i>	10
<i>I have to say this movie is very tense. The disasters in it make you think, will the world really end this way?...</i>	9
<i>The only thing that let's this film down is the science. Ignore that and it's a very enjoyable film...</i>	8
<i>It's always amusing to read and/or hear people going on about the realness (or better lack of it)...</i>	7
<i>If I were to rate this movie strictly on science or believability it would be a 2, but ultimately I found...</i>	6
<i>First off, I found that the attempt at humor at this time of destruction was unnecessary and weak...</i>	5
<i>Largely a misfire from Emmerich that's so overstuffed that it crumbles under its own weight...</i>	4
<i>I like John Cusack. He usually makes some pretty good movies. This movie is a dog...</i>	3
<i>Where to begin. This move goes beyond asking you to suspend disbelief. It asks you to beat...</i>	2
<i>Surprised that Hollywood would bother to make a movie about the Mayan "curse" prediction...</i>	1

Dataset ini juga memiliki hasil polaritas sentimen dalam bentuk kelas “positif” dan “negatif”, berdasarkan informasi bahwa data-data ulasan yang memiliki *rating* ≥ 5 akan dikategorikan sebagai ulasan positif (dengan sentimen bernilai 1) dan ulasan yang memiliki *rating* < 5 akan dikategorikan sebagai ulasan negatif (dengan sentimen bernilai -1). Dalam penelitian ini, ulasan yang memiliki *rating* bernilai 5 (59 data) akan dikeluarkan dalam proses klasifikasi

berdasarkan *sentiment* karena ulasan dengan *rating* 5 cenderung netral. Seluruh teks ulasan menggunakan bahasa Inggris. Kemudian, data angka *rating* bertipe bilangan bulat (*integer*) yang berkisar antara 1-10. Dengan demikian, kolom data yang akan diolah sebagai fitur adalah “teks”, sedangkan kolom data “*rating*” akan dijadikan sebagai basis informasi kelas-kelas data. Berikut disajikan visualisasi jumlah data per kategorisasi *rating* pada Gambar 2 dan visualisasi data per kategorisasi sentimen pada Gambar 3.



Gambar 2. Visualisasi Jumlah Data per Kategorisasi Rating dari 1000 Data di Dalam Dataset



Gambar 3. Visualisasi Jumlah Data per Kategorisasi Sentimen (Positif dan Negatif) dari 941 Data Setelah Data Dengan Rating 5 Dikeluarkan

2. Pra-pemrosesan

Pada tahap pra-pemrosesan, terdapat dua sub-tahapan yang dilakukan yaitu penghapusan *stopwords* dan tanda baca menggunakan *library NLTK* serta *text-feature extraction* menggunakan *library scikit-learn*. Pada sub-tahapan *text-feature extraction*, terdapat dua jenis metode yang diujicobakan, yaitu *CountVectorizer* dan *TfidfVectorizer*.

CountVectorizer adalah metode dasar pengukuran kata dengan menghitung jumlah kemunculan masing-masing kata pada dokumen, sehingga dapat disebut juga sebagai metode *raw count*. Implementasi metode ini akan menghasilkan *sparse matrix* [7]. Perhitungan kemunculan kata adalah metode yang sederhana, namun jika kata-kata sering muncul pada banyak dokumen, kemungkinan jumlah kemunculan kata yang besar mungkin tidak berarti banyak dalam vektor kepentingan.

TF-IDF adalah cara lain untuk mengevaluasi pentingnya kata dalam sebuah dokumen. Skor yang dihasilkan dari setiap fitur TF-IDF adalah perkalian dari *term frequency* dan *inverse document frequency*. Detail parameter yang digunakan dalam fungsi *text-feature extraction* terdapat pada Tabel 2.

Tabel 2. Parameter *Text-Feature Extraction Countvectorizer* dan *Tfidfvectorizer*

Parameter	Spesifikasi
<i>encoding</i>	utf-8
<i>lowercase</i>	True
<i>token_pattern</i>	\\b\\w+\\b
<i>ngram_range</i>	(1,1)
<i>analyzer</i>	word

3. Pembuatan Model dan *Training-Testing*

Model klasifikasi yang diimplementasikan dalam penelitian ini adalah Regresi Logistik dari *library scikit-learn*. Selain itu, teknik *dimensionality reduction* (DR) menggunakan *TruncatedSVD* dari *library scikit-learn* juga diaplikasikan untuk mempercepat waktu komputasi. *Dimensionality reduction* ini digunakan untuk mengurangi dimensi *sparse matrix* yang dihasilkan oleh *CountVectorizer* dengan mencari kombinasi linier dari fitur awal yang paling berpengaruh. Hasilnya, jumlah fitur yang awalnya berkisar pada angka ribuan dapat berkurang jauh hingga menjadi ratusan fitur.

Jumlah data yang digunakan untuk *training* adalah 80% dari jumlah data sedangkan sisanya digunakan untuk *testing*. Selain klasifikasi *rating*, dilakukan juga klasifikasi terhadap sentimen positif - negatif sebagai pembandingan hasil. Detail dimensi *sparse matrix* yang digunakan untuk *training* terdapat pada Tabel 3. Perbedaan jumlah data pada klasifikasi 2 kelas dan 10 kelas terjadi karena ulasan dengan *rating* 5 dikeluarkan dari dataset untuk klasifikasi 2 kelas seperti dijelaskan pada bagian II.

Tabel 3. Dimensi *Sparse Matrix Data Training* Setelah *Dimensionality Reduction*

<i>Feature Extraction</i>	Dimensi	
	2 Kelas	10 Kelas
<i>CountVectorizer</i>	(752, 750)	(800, 300)
<i>TfidfVectorizer</i>	(752, 40000)	(800, 40000)

Detail konfigurasi parameter model yang digunakan pada penelitian ini dapat dilihat pada Tabel 4.

Tabel 4. Konfigurasi Parameter Model

Parameter	Spesifikasi
<i>solver</i>	<i>lbfgs</i>
<i>max_iter</i>	10000
<i>class_weight</i>	<i>balanced</i>

Detail konfigurasi *environment* perangkat keras yang dipakai selama uji coba disajikan pada Tabel 5.

Tabel 5. Spesifikasi Perangkat Keras

Parameter	Spesifikasi
CPU	Intel® Core™ i5-1135G7 CPU @2.40 GHz
RAM	8 GB
Sistem Operasi	Windows 10
GPU	Intel® Iris® Xe Graphics

4. Perhitungan Performa

Pada tahapan ini, dilakukan proses perhitungan metrik performa, yang meliputi *accuracy*, *f1 score*, *precision*, dan *recall* terhadap hasil prediksi. Persamaan (1) menunjukkan perhitungan *accuracy*. *Accuracy* merupakan hasil pembagian dari penjumlahan *True Positive* (TP) dan *True Negative* (TN) dengan penjumlahan TP, TN, *False Positive* (FP), dan *False Negative* (FN).

$$accuracy = \frac{TP + TN}{P + N} \tag{1}$$

dengan *P* adalah jumlah data pada kelas positif dan *N* adalah jumlah data pada kelas negatif.

Persamaan (2) menunjukkan rumus perhitungan dari *Precision*. *Precision* dihitung dengan cara membagi TP dengan total dari TP dan FP.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

Persamaan (3) menunjukkan rumus perhitungan dari *recall*. *Recall* dihitung dengan cara membagi antara TP dengan total dari TP dan FN.

$$recall = \frac{TP}{P} \tag{3}$$

Persamaan (4) menunjukkan rumus perhitungan dari *F1 score*. *F1 Score* dihitung dengan cara membagi antara perkalian *Precision* dengan *Recall* serta penambahan *Precision* dan *Recall*.

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

III. HASIL DAN PEMBAHASAN

Pada Tabel 6 dan Tabel 7 disajikan hasil akurasi dengan berbagai variasi kondisi yang telah diuji cobakan. Dari Tabel 6 dapat dilihat bahwa model Regresi Logistik menghasilkan akurasi klasifikasi terbaik pada prediksi 10 *class rating* sebesar 36% menggunakan *CountVectorizer* sebagai input fitur dengan menerapkan *dimensionality reduction*. Jika tanpa menerapkan *dimensionality reduction* pada fitur *CountVectorizer*, Regresi Logistik hanya menghasilkan akurasi 32%. Hal ini berbeda jika menggunakan fitur *TfidfVectorizer*, penerapan *dimensionality reduction* tidak berpengaruh terhadap akurasi klasifikasi Regresi Logistik, yaitu sebesar 32%. Dari hasil ini dapat disimpulkan bahwa akurasi model Regresi Logistik dengan fitur *CountVectorizer* dan *TfidfVectorizer* dalam memprediksi *rating* untuk 10 kelas hanya berbeda 4%.

Tabel 6. Akurasi Regresi Logistik untuk 10 *Class Rating*

Feature extraction	Akurasi	
	Tanpa DR	Dengan DR
<i>CountVectorizer</i>	32%	36%
<i>TfidfVectorizer</i>	32%	32%

Tabel 7. Akurasi Regresi Logistik untuk 2 *Class Sentimen*

Feature extraction	Akurasi	
	Tanpa DR	Dengan DR
<i>CountVectorizer</i>	82%	83%
<i>TfidfVectorizer</i>	76%	76%

Selanjutnya model Regresi Logistik juga digunakan untuk memprediksi *sentiment* positif atau negatif dari suatu ulasan. Seperti dijelaskan pada bagian II, suatu ulasan dilabeli *sentiment* positif jika memiliki *rating* lebih dari 5 dan dilabeli *sentiment* negatif jika memiliki *rating* kurang dari 5. Pada prediksi sentimen dari suatu ulasan model Regresi Logistik menghasilkan akurasi klasifikasi terbaik sebesar 83% menggunakan *CountVectorizer* sebagai input fitur dengan menerapkan *dimensionality reduction*. Hasil akurasi ini lebih tinggi jika dibandingkan tanpa penerapan *dimensionality reduction*, yaitu sebesar 82%. Di lain pihak penggunaan fitur *TfidfVectorizer* hanya menghasilkan akurasi klasifikasi sebesar 76% baik dengan menerapkan *dimensionality reduction* maupun tidak. Namun model Regresi Logistik ini tidak dapat memprediksi *rating* 10 kelas dengan akurasi yang tinggi karena jumlah data per kelas yang tidak seimbang.

Untuk analisis lebih lanjut, nilai *precision*, *recall*, dan *f1 score* digunakan untuk menilai performa Regresi Logistik pada setiap kelas. Pada Tabel 8 hingga Tabel 11 disajikan nilai *precision*, *recall*, dan *F1 score* pada masing-masing kelas dengan berbagai variasi kondisi yang telah diujicobakan. Seperti terlihat pada Tabel 8 dan Tabel 9, Regresi Logistik pada prediksi 10 *class rating* mempunyai performa terbaik dalam memprediksi *Rating* 10 menggunakan fitur *CountVectorizer* dengan nilai *precision*, *recall*, dan *F1 score* masing-masing 0,59; 0,60; dan 0,60 serta 0,59; 0,62; dan 0,61 untuk *TfidfVectorizer*. Performa terendah terjadi pada kelas *Rating* 2 dan *Rating* 3 jika menggunakan fitur *CountVectorizer* dengan nilai *precision*, *recall*, dan *F1 score* nol semua. Sedangkan jika menggunakan fitur *TfidfVectorizer* performa terendah pada kelas *Rating* 2, *Rating* 3, dan *Rating* 4 dengan nilai *precision*, *recall*, dan *F1 score* nol semua.

Tabel 8. Nilai *Precision*, *Recall*, dan *F1 Score* Prediksi 10 *Class Rating* Dengan *Countvectorizer*

Class	Precision	Recall	F1-score
<i>Rating</i> 1	0,27	0,25	0,26
<i>Rating</i> 2	0	0	0
<i>Rating</i> 3	0	0	0
<i>Rating</i> 4	0,20	0,33	0,25
<i>Rating</i> 5	0,18	0,18	0,18
<i>Rating</i> 6	0,43	0,39	0,41
<i>Rating</i> 7	0,35	0,21	0,26
<i>Rating</i> 8	0,15	0,24	0,19
<i>Rating</i> 9	0,38	0,26	0,31
<i>Rating</i> 10	0,59	0,60	0,60

Pada prediksi 2 *class sentiment*, Regresi Logistik menghasilkan performa klasifikasi terbaik pada kelas positif baik menggunakan fitur *CountVectorizer* maupun *TfidfVectorizer*, seperti tampak pada Tabel 10 dan Tabel 11. Lebih lanjut Regresi Logistik masih mampu memprediksi dengan tepat sebagian data pada sentimen negatif jika menggunakan fitur *CountVectorizer* meskipun dengan nilai *precision*, *recall*, dan *f1 score* masing-masing adalah 0,47, 0,70, dan 0,56. Di lain pihak, jika menggunakan fitur *TfidfVectorizer* Regresi Logistik tidak dapat memprediksi sentimen negatif sama sekali. Hal ini terlihat dari nilai *precision*, *recall*, dan *F1 score* yang semuanya nol, seperti ditunjukkan pada Tabel 11.

Tabel 9. Nilai *Precision*, *Recall*, dan *F1 Score* Prediksi 10 *Class Rating* Dengan *Tfidfvectorizer*

<i>Class</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Rating 1	0,09	0,14	0,11
Rating 2	0	0	0
Rating 3	0	0	0
Rating 4	0	0	0
Rating 5	0,27	0,15	0,19
Rating 6	0,24	0,29	0,26
Rating 7	0,35	0,24	0,29
Rating 8	0,08	0,22	0,11
Rating 9	0,48	0,20	0,29
Rating 10	0,59	0,62	0,61

Tabel 10. Nilai *Precision*, *Recall*, dan *F1 Score* Prediksi 2 *Class Sentimen* Dengan *Countvectorizer*

<i>Class</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Negatif	0,47	0,70	0,56
Positif	0,94	0,85	0,89

Tabel 11. Nilai *Precision*, *Recall*, dan *F1 Score* Prediksi 2 *Class Sentimen* Dengan *Tfidfvectorizer*

<i>Class</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Negatif	0	0	0
Positif	1	0,76	0,86

IV. KESIMPULAN

Berdasarkan hasil uji coba yang dilakukan pada penelitian ini, terdapat beberapa kesimpulan yang dapat ditarik yaitu untuk kasus *multi class sentiment analysis* dengan 10 kelas, metode *feature extraction* pada data teks memberikan hasil yang berbeda pada akurasi hasil prediksi. Dengan menerapkan *dimensionality reduction* pada fitur *CountVectorizer* Regresi

Logistik mampu menghasilkan akurasi prediksi 36% sedangkan *TfidfVectorizer* hanya 32%. Pengaplikasian teknik *dimensionality reduction* dapat meningkatkan akurasi hasil prediksi Regresi Logistik jika menggunakan fitur *CountVectorizer* pada klasifikasi 2 *class sentiment* dari 82% menjadi 83%.

Pada *multiclass sentiment analysis* dengan 10 kelas menggunakan Regresi Logistik diperoleh akurasi terbaik 36%. Hal ini mungkin disebabkan jumlah data di setiap kelas tidak seimbang. Dalam konteks data *training* yang digunakan dalam penelitian ini, lebih cocok apabila diklasifikasikan dengan label positif-negatif. Hal ini tampak dari akurasi klasifikasi 2 kelas lebih tinggi yaitu 83% dibandingkan akurasi klasifikasi 10 kelas yang hanya mencapai 36%

Untuk meningkatkan akurasi dalam memprediksi *rating* maupun *sentiment* sebuah film perlu diinvestigasi penggunaan model klasifikasi yang lebih kompleks, seperti *Artificial Neural Network* maupun *Support Vector Machine*, pada penelitian berikutnya. Selain itu penggunaan fitur-fitur lain seperti *Bag of Words*, *Word Embedding*, maupun fitur berbasis NLP (*Natural Language Processing*) juga perlu dipertimbangkan untuk meningkatkan akurasi.

REFERENSI

- [1] E. Setiawan, "Kamus Besar Bahasa Indonesia (KBBI)," *Arti kata pendapat - Kamus Besar Bahasa Indonesia (KBBI) Online*. [Online]. Available: <https://kbbi.web.id/pendapat>. [Accessed: 03-Jun-2021].
- [2] P. McCullagh and J. A. Nelder, "Generalized linear models," 1989.
- [3] R. N. Sucky, "Multiclass classification algorithm from scratch with a project in Python: Step by step guide," *Medium*, 03-Nov-2020. [Online]. Available: <https://towardsdatascience.com/multiclass-classification-algorithm-from-scratch-with-a-project-in-python-step-by-step-guide-485a83c79992>. [Accessed: 06-Sep-2020].
- [4] N. Asghar, "Yelp Dataset Challenge: Review Rating Prediction," *arXiv preprint*, 2016.
- [5] M. Umer, I. Ashraf, A. Mehmood, S. Ullah, and G. S. Choi, "Predicting numeric ratings for Google Apps using text features and Ensemble Learning," *ETRI Journal*, vol. 43, no. 1, pp. 95–108, 2020.
- [6] A. Benlahbib, "1000 Movie Reviews (Review + Attached rating + Sentiment polarity) for Reputation Generation," *Mendeley Data*, 2019.
- [7] R. Qiao, "Yelp review rating prediction: Sentiment analysis and the neighborhood-based recommender," dissertation, 2019.