

Analisis Sentimen Ulasan Aplikasi Transportasi *Online* Menggunakan *Multinomial Naïve Bayes* dan *Query Expansion Ranking*

Yuma Anugrah Virya Gunawan^{a1}, Ngurah Agus Sanjaya ER^{a2}, Ida Bagus Made Mahendra^{a3}, I Made Widiartha^{a4}, I Gusti Ngurah Anom Cahyadi Putra^{a5}, I Gusti Agung Gede Arya Kadyanan^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Udayana
Bali, Indonesia

¹yumagunawan22@gmail.com

²agus_sanjaya@unud.ac.id

³ibm.mahendra@unud.ac.id

⁴madewidiartha@unud.ac.id

⁵anom.cp@unud.ac.id

⁶gungde@unud.ac.id

Abstract

The rapid development of the transportation industry in recent years has led to a new innovation in the field of transportation, namely the application of online transportation services. To facilitate the translation of user satisfaction, in addition to users being able to provide reviews, the Google Play Store uses a rating system consisting of a rating of 1 to 5. However, users often do not provide a rating that is in accordance with the review so that this is not enough to determine the sentiment of the review. This research is focused on evaluating the performance of the features selection using Query Expansion Ranking on the Multinomial Naïve Bayes method in the problem of sentiment analysis on the two of most popular online transportation service applications in Indonesia, namely Gojek and Grab. From the results of the performance evaluation using k-fold cross validation, it was found that the best feature selection ratio was 20% with the best performance in terms of precision.

Keywords: Sentiment Analysis, Feature Selection, Multinomial Naïve Bayes, TF – IDF, Query Expansion Ranking

1. Pendahuluan

Perkembangan industri transportasi yang pesat dalam beberapa tahun terakhir membawa sebuah inovasi baru di bidang transportasi, yaitu adanya layanan transportasi *online* dengan menggunakan teknologi berbasis aplikasi. Aplikasi layanan transportasi *online* ini memberikan kemudahan bagi para penggunanya untuk memesan layanan seperti mengantar orang, makanan, atau barang hanya dengan menggunakan *smartphone* [1].

Dalam hal distribusi aplikasi, khususnya di sistem operasi *android*, salah satu wadah bagi pengguna untuk memperoleh aplikasi dari pengembang adalah melalui *Google Play Store*. Di samping itu, pengguna juga dapat menuliskan dan membagikan ulasan terkait pengalaman terhadap pemakaian suatu aplikasi. Untuk memudahkan penerjemahan kepuasan pengguna, banyak *platform* seperti *Google Play Store* memakai sistem rating yang terdiri dari rating 1 hingga 5, namun seringkali pengguna tidak memberikan rating yang sesuai dengan ulasannya sehingga hal tersebut belum cukup untuk menentukan sentimen dari ulasan tersebut. Oleh karena itu, untuk memudahkan pengelolaan dan analisis data ulasan, perlu dilakukan analisis sentimen. Pendekatan analisis sentimen dapat menyediakan analisis yang mendalam mengenai sentimen, pendapat, dan ekspresi seseorang dalam bentuk data tekstual [2].

Dalam klasifikasi sentimen, ada satu hal yang paling penting yaitu proses pelabelan atau *labeling*. Proses pelabelan adalah proses pemberian label pada dataset ulasan. Proses pelabelan yang paling umum adalah pelabelan manual dengan bantuan ahli bahasa. Namun, hal ini sulit dilakukan dengan jumlah dataset yang banyak karena akan memakan waktu yang lama dan memerlukan tim ahli yang

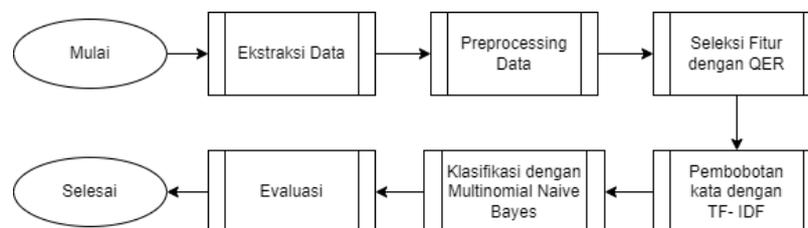
banyak. Penelitian ini menggunakan skema pelabelan *average labeling* karena dapat diterapkan pada ukuran dataset yang besar dengan waktu yang singkat [3].

Salah satu cara meningkatkan kualitas fitur dataset adalah dengan proses seleksi fitur. Metode seleksi fitur yang digunakan dalam penelitian ini adalah metode *Query Expansion Ranking* [4]. Sedangkan, untuk klasifikasi teks menggunakan salah satu metode yang banyak digunakan pada pendekatan klasifikasi dan analisis sentimen yakni varian *multinomial* dari metode *Naïve Bayes* yaitu *Multinomial Naïve Bayes* [5]. Implementasi *query expansion* sendiri sudah banyak diterapkan pada berbagai penelitian, salah satunya pada penelitian *set labelling* [6]. Penelitian oleh [7] menunjukkan bahwa metode seleksi fitur *Query Expansion Ranking* juga menghasilkan performa yang baik pada metode KNN.

Penelitian ini bertujuan untuk menemukan performa rasio seleksi fitur terbaik dari metode *Query Expansion Ranking* pada metode klasifikasi *Multinomial Naïve Bayes* dalam permasalahan analisis sentimen terhadap ulasan aplikasi layanan transportasi *online*. Performa diukur berdasarkan hasil *accuracy*, *precision*, *recall*, dan *f1 – score*.

2. Metode Penelitian

Pada bagian ini akan dijelaskan gambaran secara umum mengenai alur penelitian yang dilakukan oleh penulis. Pertama adalah proses ekstraksi data dari sumber yakni situs web *Google Play Store*, kemudian dari data yang diekstraksi dihasilkan *raw data*. Dataset hasil ekstraksi akan melalui proses *text preprocessing* agar menjadi *clean data*. Data yang sudah bersih akan melalui proses seleksi fitur menggunakan metode *Query Expansion Ranking* yang berkaitan dengan skenario pengujian yang akan dijalankan yakni untuk mencari rasio persentase seleksi fitur terbaik dengan penggunaan seluruh fitur sebagai *baseline*. Selanjutnya, dilakukan pembobotan kata dengan TF – IDF yang menghasilkan vektor fitur untuk digunakan dalam tahap klasifikasi. Tahap klasifikasi dilakukan dengan metode *Multinomial Naïve Bayes*. Hasil klasifikasi dapat dievaluasi berdasarkan nilai *accuracy*, *recall*, *precision*, dan *F – 1 score*. Diagram alir penelitian ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

2.1. Ekstraksi Data

Ekstraksi data dilakukan menggunakan teknik *web scraping* dengan bahasa pemrograman Python dan *library google-play-scrapers*. Data yang diambil merupakan ulasan dari *region* Indonesia yang berbahasa Indonesia dari dua aplikasi layanan transportasi *online* yakni *Gojek* dan *Grab*. Dalam penelitian ini, penulis mengambil 1000 ulasan dengan masing – masing 500 data untuk tiap kelas positif dan negatif dari tiap aplikasi sehingga total menggunakan 2000 ulasan. Adapun skema pelabelan yang digunakan yaitu *average labeling* dimana berdasarkan pada rating kedua aplikasi saat proses pengambilan data, maka ulasan dengan rating 1 – 4 dilabelkan sebagai negatif dan rating 5 sebagai positif. Hasil ekstraksi ulasan ditunjukkan pada Gambar 2.

```

{'reviewId': 'gp:A0qpT0E0Mv80XG85eNdMPiyow-CzPjKfW0wy2wh1RZRjDlX7s8pr242DEsK3P3Aeww5mQQZHec1weAti7Psdmg',
'userName': 'Syahdan N',
'userImage': 'https://play-lh.googleusercontent.com/a-/A0h14GjBvZh43xfIYY8DHmoFLt0CUFWQTwhYJXIC65ASDA',
'content': 'Tolong ditambahkan menu atau pilihan penghapusan akun', 'score': 4,
'thumbsUpCount': 0, 'reviewCreatedVersion': '4.16.1',
'at': datetime.datetime(2021, 4, 6, 9, 27, 46), 'replyContent': None, 'repliedAt': None}
  
```

Gambar 2. Hasil Ekstraksi Ulasan

2.2. Text Preprocessing

Dokumen hasil ekstraksi data yang sudah disimpan akan melalui proses *preprocessing* seperti yang ditunjukkan pada Tabel 1. Tahap *preprocessing* pada penelitian ini terdiri dari *case folding*, *tokenizing*, *word handling*, *filtering*, dan *stemming*.

Case folding adalah tahapan teknik pembersihan yang dilakukan untuk menyisakan kata dengan karakter huruf kecil. Oleh karena itu, dilakukan penghilangan angka, tanda baca, emotikon dan lain sebagainya serta pengubahan semua karakter ke huruf kecil.

Tokenizing merupakan tahapan pemecahan *string* kalimat menjadi *token* yang berupa kata.

Word handling merupakan tahapan yang dilakukan untuk menangani kata singkatan atau salah ejaan (*typo*) dengan cara mengganti kata pada ulasan menjadi kata yang sesuai dengan *dictionary* yang disediakan. Algoritma penanganan kata menggunakan algoritma dari Peter Norvig¹.

Filtering adalah tahapan untuk menghilangkan kata yang sering muncul namun tidak memiliki relevansi dengan teks atau dapat dikatakan kurang bermakna. Untuk kamus kata dan *stopwords* yang digunakan merupakan kombinasi dari *list stopwords* yang disediakan oleh *library Sastraw* dan media internet².

Stemming adalah tahapan untuk mengubah kata menjadi bentuk dasar dengan menghilangkan semua imbuhan.

Tabel 1. Proses *Text Preprocessing*

No	Proses	Input	Output
1	<i>Case Folding</i>	Sava sanqt kcewa pada aolikasi ini, pelayanan sring terlambat belakangan ini.. Harapannya bisa lebih baik lagi ke depannya,.. 🙏🙏🙏	sava sanqt kcewa pada aolikasi ini pelayanan sring terlambat belakangan ini harapannya bisa lebih baik lagi ke depannya
2	<i>Tokenizing</i>	sava sanqt kcewa pada aolikasi ini pelayanan sring terlambat belakangan ini harapannya bisa lebih baik lagi ke depannya	'saya', 'sangt', 'kcewa', 'pada', 'aolikasi', 'ini', 'pelayanan', 'sring', 'terlambat', 'belakangan', 'ini', 'harapannya', 'bisa', 'lebih', 'baik', 'lagi', 'ke', 'depannya'
3	<i>Word Handling</i>	'saya', 'sangt', 'kcewa', 'pada', 'aolikasi', 'ini', 'pelayanan', 'sring', 'terlambat', 'belakangan', 'ini', 'harapannya', 'bisa', 'lebih', 'baik', 'lagi', 'ke', 'depannya'	'saya', 'sangat', 'kecewa', 'pada', 'aplikasi', 'ini', 'pelayanan', 'sering', 'terlambat', 'belakangan', 'ini', 'harapannya', 'bisa', 'lebih', 'baik', 'lagi', 'ke', 'depannya'
4	<i>Filtering</i>	'saya', 'sangat', 'kecewa', 'pada', 'aplikasi', 'ini', 'pelayanan', 'sering', 'terlambat', 'belakangan', 'ini', 'harapannya', 'bisa', 'lebih', 'baik', 'lagi', 'ke', 'depannya'	'kecewa', 'aplikasi', 'pelayanan', 'terlambat', 'harapannya', 'lebih', 'baik', 'depannya'
5	<i>Stemming</i>	kecewa, 'aplikasi', 'pelayanan', 'terlambat', 'harapannya', 'lebih', 'baik', 'depannya'	kecewa, 'aplikasi', 'layan', 'lambat', 'harap', 'lebih', 'baik', 'depan'

2.3. Seleksi Fitur dengan *Query Expansion Ranking*

Dalam penelitian ini, penulis menggunakan metode seleksi fitur *Query Expansion Ranking* (QER). Pada metode QER, skor tiap fitur akan dihitung, kemudian fitur dengan skor terendah akhirnya dipilih dan digunakan dalam proses klasifikasi. Jika skor fitur rendah, perbedaan antara probabilitas kelas positif dan negatif menjadi besar. Hal ini menunjukkan bahwa fitur bersifat lebih unik pada setiap kelas dan lebih bermakna bagi proses klasifikasi [4]. Adapun langkah perhitungannya sebagai berikut:

- a. Hitung nilai probabilitas fitur *f* pada dokumen kelas positif.

$$pf = \frac{df^+ + 0.5}{n^+ + 1.0} \quad (1)$$

- b. Hitung nilai probabilitas fitur *f* pada dokumen kelas negatif.

$$qf = \frac{df^- + 0.5}{n^- + 0.5} \quad (2)$$

¹ <http://norvig.com/spell-correct.html>

² <http://hikaruyuuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-indonesia/>

c. Hitung skor QER untuk fitur f

$$score_f = \frac{|pf+qf|}{|pf-qf|} \quad (3)$$

Keterangan :

- df^+ : jumlah dokumen kelas positif yang memuat fitur f
- n^+ : jumlah dokumen kelas positif
- df^- : jumlah dokumen kelas negatif yang memuat fitur f
- n^- : jumlah dokumen kelas negatif
- pf : nilai probabilitas fitur f pada dokumen kelas positif
- qf : nilai probabilitas fitur f pada dokumen kelas negatif
- $score_f$: nilai skor QER fitur f

2.4. Pembobotan Kata dengan *Term Frequency – Inverse Document Frequency* (TF – IDF)

TF - IDF adalah metode pembobotan kata yang digunakan untuk menentukan pentingnya sebuah kata dalam sebuah dokumen. Ada dua komponen dalam perhitungan nilai TF – IDF, yaitu TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*). TF menentukan pentingnya suatu kata relatif terhadap kemunculannya dalam suatu dokumen, sedangkan IDF menentukan suatu kata penting dalam suatu dokumen jika tidak sering muncul di dokumen lain [8]. Adapun rumus perhitungan sesuai penelitian oleh [9]:

$$a_{ij} = tf_{ij} * \log \left(\frac{N}{n_j} \right) \quad (4)$$

Dengan keterangan :

- a_{ij} : nilai bobot *term j* pada dokumen i
- tf_{ij} : jumlah kemunculan kata atau nilai *term frequency term j* pada dokumen i
- N : jumlah dokumen pada dataset
- n_j : jumlah dokumen yang memuat *term j*

2.5. Klasifikasi dengan *Multinomial Naïve Bayes*

Multinomial Naïve Bayes merupakan implementasi algoritma *Naïve Bayes* yang umumnya digunakan dalam pemrosesan teks dengan mengikuti prinsip distribusi multinomial [10]. Penggunaan model distribusi multinomial menunjukkan bahwa vektor fitur dalam suatu dokumen dibentuk dari frekuensi kemunculan setiap kata pada dokumen tersebut. Adapun algoritma perhitungannya [11]:

a. Untuk tiap fitur pada vektor data latih, hitung nilai kemungkinan kemunculan fitur i pada kelas y .

$$\theta_{yi} = \frac{N_{yi+a}}{N_y+an} \quad (5)$$

Keterangan:

- θ_{yi} : nilai kemungkinan kemunculan *term i* pada kelas y
- N_{yi} : jumlah kemunculan *term i* pada kelas y
- N_y : jumlah kemunculan seluruh fitur pada kelas y
- a : nilai *Laplace smoothing parameter*
- n : total fitur dalam koleksi dokumen

b. Hitung nilai *prior probability* pada setiap kelas y

$$P_y = \frac{D_y}{D} \quad (6)$$

Keterangan:

- P_y : nilai *prior probability* pada kelas y
- D_y : jumlah dokumen pada kelas y
- D : jumlah seluruh dokumen

c. Tentukan hasil prediksi kelas pada data uji

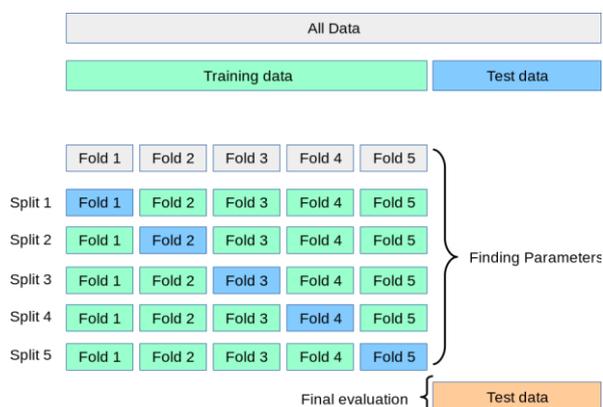
$$\hat{c} = \operatorname{argmax} P_y \prod (\theta_{yi})^{M_{yi}} \quad (7)$$

Keterangan:

- \hat{c} : kelas hasil prediksi
- M_{yi} : jumlah kemunculan *term i* pada kelas y dalam dokumen uji

2.6. Skenario Pengujian dan Evaluasi

Pengujian dalam penelitian ini menggunakan *K – Fold Cross – Validation* yang ditunjukkan pada Gambar 3 [12]. Dataset ulasan akan dibagi menjadi data latih dan data uji terlebih dahulu dengan porsi 80 : 20. Kemudian data latih akan dilakukan iterasi sebanyak *k* bagian atau *folds* yang membagi tiap bagiannya menjadi data latih dan data validasi dengan porsi satu bagian menjadi data validasi dan bagian lainnya menjadi data latih secara bergantian untuk setiap iterasinya. Pengujian ini dilakukan untuk mendapatkan parameter rasio seleksi fitur terbaik yakni dengan melihat rata – rata nilai metrik (*accuracy, precision, recall, dan f1 – score*) dari keseluruhan iterasi terhadap rasio seleksi fitur yang digunakan. Parameter rasio seleksi fitur terbaik akan dibandingkan performanya pada data uji dengan tanpa penggunaan seleksi fitur atau penggunaan fitur penuh sebagai metode *baseline*.



Gambar 3. *K – Fold Cross Validation*

Evaluasi bertujuan untuk mengukur performa pada sistem yang dibangun. Performa sistem klasifikasi dapat dievaluasi dengan membandingkan kelas yang diprediksi dengan kelas sebenarnya. Informasi ini dapat dituangkan dalam tabel yang disebut *confusion matrix* seperti yang ditunjukkan pada Tabel 2. Dari perhitungan pada *confusion matrix*, diperoleh nilai *accuracy, precision, recall, dan f1 – score*.

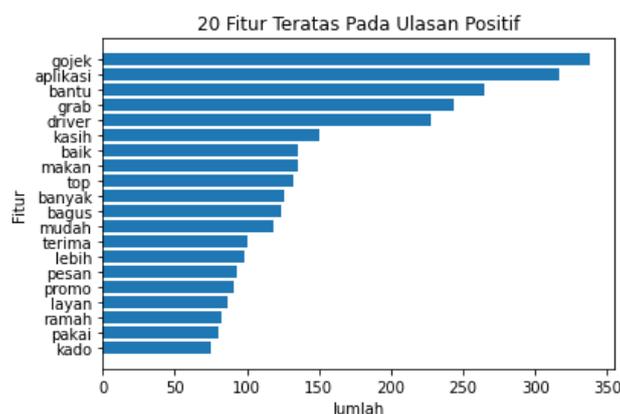
Tabel 2. *Confusion Matrix*

	Actual Positive	Actual Negative
Predicted Positive	<i>True Positive</i>	<i>False Positive</i>
Predicted Negative	<i>False Negative</i>	<i>True Negative</i>

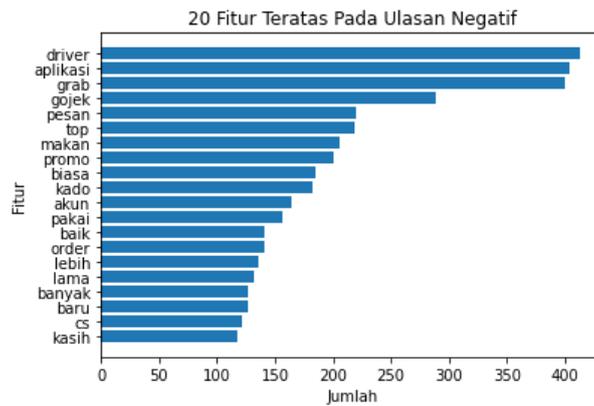
3. Hasil dan Pembahasan

3.1. Hasil Eksplorasi Data

Pada tahap ini, dilakukan eksplorasi data untuk mengetahui informasi sebaran fitur pada dataset yang digunakan. Salah satu langkah yang dapat dilakukan untuk mengetahui sebaran fitur adalah dengan melihat jumlah fitur terbanyak. Gambar 4 dan Gambar 5 menunjukkan 20 fitur teratas pada kelas positif dan negatif secara berurutan.



Gambar 4. 20 Fitur Teratas Pada Ulasan Kelas Positif



Gambar 5. 20 Fitur Teratas Pada Ulasan Kelas Negatif

Dengan melihat 20 fitur teratas, dapat memberikan gambaran lebih jelas mengenai jumlah tiap fitur pada tiap kelas. Dari kedua hasil visualisasi terlihat bahwa terdapat banyak kesamaan fitur yang sering muncul pada kedua kelas yakni kata “*gojek*”, “*grab*”, “*driver*”, dan “*aplikasi*”. Dimana kata “*gojek*” dan “*grab*” merupakan nama aplikasi yang merupakan sumber dari dataset yang digunakan serta kata “*driver*” dan “*aplikasi*” yang masih berkaitan dengan aplikasi layanan transportasi *online*. Selain itu, juga terdapat fitur yang dominan pada kelas tertentu seperti kata “*bantu*” dan “*bagus*” yang dominan pada kelas positif serta kata “*promo*”, “*akun*” dan “*kecewa*” yang dominan pada kelas negatif. Dalam visualisasi juga ditunjukkan bahwa terdapat ketimpangan pada jumlah fitur, dimana kata “*driver*” pada kelas positif hanya berjumlah sekitar 200, sedangkan pada kelas negatif berjumlah sekitar 400. Hal itu berarti dalam dataset yang digunakan lebih banyak kata “*driver*” dengan konotasi negatif. Begitu pula untuk kata “*grab*”.

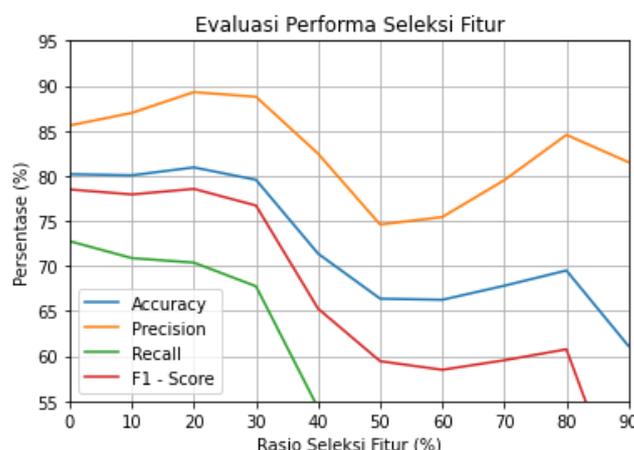
3.2. Hasil Evaluasi Performa Seleksi Fitur

Pengujian dilakukan untuk mencari parameter rasio seleksi fitur terbaik menggunakan *cross validation* dengan 10 *folds*. Dari 1600 data sebagai data latih, dibagi lagi menjadi data latih dan data validasi pada proses *cross validation*. Performa rasio seleksi fitur diukur dari rata – rata nilai metrik *accuracy*, *precision*, *recall* dan *f1 – score*. Adapun hasil evaluasi performa seleksi fitur dapat dilihat pada Tabel 3.

Tabel 3. Evaluasi Performa Seleksi Fitur

Rasio Seleksi Fitur	Rata – Rata			
	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 – score</i>
0	0.80188	0.85591	0.7275	0.78492
10	0.80063	0.86985	0.70875	0.7794
20	0.81	0.89514	0.70625	0.78559
30	0.79688	0.8765	0.6775	0.76812
40	0.71438	0.82535	0.53875	0.6507
50	0.66188	0.74384	0.4925	0.59155
60	0.65686	0.75234	0.4775	0.58318
70	0.67812	0.79471	0.47625	0.59446
80	0.69312	0.84773	0.47625	0.60801
90	0.61063	0.81064	0.28625	0.41826

Untuk memudahkan melihat perubahan tren kenaikan dan penurunan nilai metrik evaluasi, dibentuk visualisasi berupa grafik yang ditunjukkan pada Gambar 6. Dimana 0% berarti tanpa penggunaan seleksi fitur atau penggunaan seluruh fitur pada dokumen hingga 90% pengurangan fitur. Dari grafik tersebut diketahui bahwa *precision*, *accuracy*, dan *f1 – score* terdapat kenaikan tren paling tinggi mencapai 20% penggunaan seleksi fitur, kemudian tren menurun setelahnya. Adapun nilai *recall* terdapat penurunan dari metode *baseline*. Kenaikan dan penurunan dari metode *baseline* terlihat tidak terlalu signifikan pada segi *accuracy*, *f1 – score*, dan *recall*. Namun, dari segi *precision* perubahan hampir mencapai 5% pada penggunaan rasio seleksi fitur 20%.



Gambar 6. Tren Evaluasi Performa Seleksi Fitur

Penurunan tren setelah rasio penggunaan seleksi fitur sebanyak 20% menunjukkan adanya pengurangan fitur yang terlalu banyak jumlahnya sehingga menyebabkan fitur yang memiliki relevansi dengan data validasi juga ikut dihilangkan. Kenaikan yang cukup signifikan dari segi *precision* terjadi karena nilai *false positive* mengalami penurunan dimana berarti pada rasio seleksi fitur 20%, jumlah ulasan dengan kelas negatif yang diberi label positif oleh model sudah berkurang dibandingkan pada metode *baseline*. Namun, di sisi lain penurunan *recall* disebabkan oleh peningkatan nilai *false negative* yang berarti model lebih banyak memberikan label negatif pada ulasan positif dibandingkan metode *baseline*. Nilai *f1-score* hampir tidak terlihat perubahan karena kenaikan dan penurunan nilai *precision* dan *recall* yang sebanding. Sedangkan kenaikan akurasi yang tidak terlalu signifikan disebabkan oleh sedikit peningkatan pada jumlah prediksi yang benar dari kedua kelas. Seleksi fitur dengan rasio 20% dinilai paling optimal dibandingkan rasio lainnya, maka akan dipilih untuk dibandingkan performanya dengan metode *baseline* pada data uji yang belum dikenali pada proses *cross validation* sejumlah 400 data. Evaluasi performa model pada data uji dapat dilihat pada Tabel 4.

Tabel 4. Evaluasi Performa pada Data Uji

	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 - score</i>
Tanpa Seleksi Fitur	0.7575	0.81988	0.66	0.73130
Rasio Seleksi Fitur Terbaik (20%)	0.75	0.85211	0.605	0.70760

Berdasarkan Tabel 4 dari hasil evaluasi performa pada data uji didapatkan nilai metode *baseline* atau tanpa penggunaan seleksi fitur untuk *accuracy*, *precision*, *recall*, dan *f1-score* secara berturut – turut adalah 0.7575, 0.81988, 0.66 dan 0.73130. Sedangkan untuk penggunaan rasio seleksi fitur terbaik yakni sebesar 20% didapatkan 0.75, 0.85211, 0.605, 0.70760. Dapat diketahui bahwa dalam hal performa pada data uji, metode *baseline* lebih baik dari segi *accuracy*, *recall*, dan *f1-score*. Sedangkan penggunaan seleksi fitur sedikit lebih baik dari sisi *precision*.

4. Kesimpulan

Dari hasil pengujian dan evaluasi pada penelitian ini diketahui bahwa penggunaan rasio seleksi fitur paling optimal yaitu pada rasio pengurangan 20% fitur. Hasil evaluasi performa pada data uji menunjukkan bahwa penggunaan rasio seleksi fitur sebanyak 20% memperoleh nilai *accuracy*, *precision*, *recall*, dan *f1-score* secara berturut – turut sebesar 0.75, 0.85211, 0.605, dan 0.70760. Sedangkan pada metode *baseline* memperoleh nilai 0.7575, 0.81988, 0.66 dan 0.73130. Dalam hal performa pada data uji, penggunaan seleksi fitur lebih unggul dari segi *precision*, sedangkan metode *baseline* sedikit lebih baik dari segi *accuracy*, *recall*, dan *f1-score*.

Adapun dari pengujian pada variasi rasio seleksi fitur yang berbeda, diketahui bahwa performa dari segi *precision*, *accuracy*, dan *f1-score* terdapat kenaikan tren hingga paling tinggi mencapai 20%, kemudian tren menurun setelahnya. Adapun nilai *recall* terdapat penurunan tren. Peningkatan nilai *precision* dan penurunan nilai *recall* tersebut dipengaruhi oleh penurunan nilai *false positive* dan

kenaikan nilai *false negative*, sehingga terdapat kecenderungan model lebih banyak kepada pemberian label negatif.

Daftar Pustaka

- [1] I. Farida, A. Tarmizi, and Y. November, "Analisis Pengaruh Bauran Pemasaran 7P Terhadap Kepuasan Pelanggan Pengguna Gojek Online," *J. Ris. Manaj. dan Bisnis Fak. Ekon. UNIAT*, vol. 1, no. 1, pp. 31–40, 2016, doi: 10.36226/jrmb.v1i1.8.
- [2] L. Zhang and B. Liu, "Sentiment Analysis and Opinion Mining", *Encyclopedia of Machine Learning and Data Mining*, pp. 1152-1161, 2017. Available: 10.1007/978-1-4899-7687-1_907 [Accessed 12 July 2022].
- [3] V. O. Tama, Y. Sibaroni, and Adiwijaya, "Labeling Analysis in the Classification of Product Review Sentiments by using Multinomial Naive Bayes Algorithm," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012036.
- [4] T. Parlar, S. A. Özel, and F. Song, "QER: a new feature selection method for sentiment analysis," *Human-centric Comput. Inf. Sci.*, vol. 8, no. 1, pp. 1–19, 2018, doi: 10.1186/s13673-018-0135-8.
- [5] A. Sabrani, I. G. W. Wedashwara W., and F. Bimantoro, "Multinomial Naïve Bayes untuk Klasifikasi Artikel Online tentang Gempa di Indonesia," *J. Teknol. Informasi, Komputer, dan Apl. (JTika)*, vol. 2, no. 1, pp. 89–100, 2020, doi: 10.29303/jtika.v2i1.87.
- [6] E. R. Ngurah Agus Sanjaya, T. Abdessalem, J. Read, and S. Bressan, "Set labelling using multi-label classification," *ACM Int. Conf. Proceeding Ser.*, pp. 216–220, 2018, doi: 10.1145/3282373.3282391.
- [7] N. D. Mentari, M. A. Fauzi, and L. Muflikhah, "Analisis Sentimen Kurikulum 2013 Pada Sosial Media Twitter Menggunakan Metode K-Nearest Neighbor dan Feature Selection Query Expansion Ranking," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 8, pp. 2739–2743, 2018.
- [8] F. Alzami, E. D. Udayanti, D. P. Prabowo, and R. A. Megantara, "Document Preprocessing with TF-IDF to Improve the Polarity Classification Performance of Unstructured Sentiment Analysis," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, pp. 235–242, 2020, doi: 10.22219/kinetik.v5i3.1066.
- [9] M. Liu and J. Yang, "An improvement of TFIDF weighting in text categorization," *Int. Conf. Comput. Technol. Sci.*, vol. 47, no. Iccts, pp. 44–47, 2012, doi: 10.7763/IPCSIT.2012.V47.9.
- [10] A. A. Farisi, Y. Sibaroni, and S. Al Faraby, "Sentiment analysis on hotel reviews using Multinomial Naïve Bayes classifier," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012024.
- [11] N. Rezaeian and G. Novikova, "Persian text classification using naive bayes algorithms and support vector machine algorithm," *Indones. J. Electr. Eng. Informatics*, vol. 8, no. 1, pp. 178–188, 2020, doi: 10.11591/ijeii.v8i1.1696.
- [12] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python Pedregosa, Varoquaux, Gramfort *et al.*," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011, [Online]. Available: <http://scikit-learn.org>.