

Analisis Sentimen dan Pemodelan Topik Ulasan Aplikasi Noice Menggunakan XGBoost dan LDA

Alim Ikegami^{a1}, I Dewa Made Bayu Atmaja Darmawan^{a2}

^aProgram Studi Informatika, Universitas Udayana
Jimbaran, Badung, Bali, Indonesia, 80361

¹alimikegami1@gmail.com

²dewabayu@unud.ac.id (Corresponding author)

Abstract

In recent years, audio content has seen a rise in consumption. The COVID-19 pandemic also contributes to the rise in consumption. In the survey that was conducted in 2021, more than 40% of people in France, Germany, and Spain have been listening to more audio content since the first restriction on COVID-19 came into place. One of the rising startups in Indonesia that offers audio content with their original and exclusive content is Noice. To maintain their quality of service, it's important to look into the reviews that were written for their application. To analyze the reviews, sentiment analysis and topic modeling can be used to extract the sentiment polarity and the topics that are discussed on each sentiment polarity. In this study, XGBoost and Latent Dirichlet Allocation are used to analyze the reviews that were written in Google Play Store. The result of the sentiment analysis yielded accuracy, precision, recall, and F1-score of 86,8%, 83,9%, 77,9%, and 80,2%. While the topic modeling managed to extract 3 and 6 topics respectively for positive and negative reviews.

Keywords: Sentiment Analysis, Topic Modeling, Noice, XGBoost, LDA

1. Pendahuluan

Konten audio merupakan salah satu bentuk konten yang dapat dinikmati dengan mudah melalui *smartphone* yang kini berada di genggaman sebagian besar masyarakat di seluruh dunia. Konten audio disajikan dalam berbagai jenis, yaitu dapat berupa *podcast*, *audioseries*, ataupun *audiobook*. Dalam beberapa tahun terakhir ini, konsumsi masyarakat terhadap konten audio semakin meningkat. Terjadinya pandemi akibat virus COVID-19 juga mendorong konsumsi konten audio. Berdasarkan survei yang dilakukan oleh *Sortlist* pada tahun 2021, lebih dari 40% masyarakat di Prancis, Jerman, dan Spanyol menyatakan bahwa konsumsi konten audio mereka meningkat sejak pertama kalinya terjadinya restriksi akibat COVID-19 [1].

Salah satu *platform* konten audio yang menyuguhkan berbagai jenis konten audio dalam bahasa Indonesia adalah Noice. Noice menyediakan konten audio seperti *podcast*, radio, *audiobook*, maupun *audioseries*. Noice menghadirkan berbagai konten original atau eksklusif yang hanya dapat didengarkan melalui aplikasinya. Pada tahun 2022, Noice dinobatkan sebagai salah satu dari top 50 *startup* di Asia pada list versi Tech In Asia [2]. Untuk menjaga kualitas layanan yang ditawarkan, ulasan dari pengguna aplikasi dapat memberikan berbagai informasi yang berharga terkait dengan pengalaman para pengguna dalam menggunakan aplikasi tersebut.

Banyaknya ulasan yang ada di internet menyebabkan sulitnya proses ekstraksi informasi dari ulasan tersebut jika proses penggalian informasi dilakukan secara manual dengan membaca keseluruhan ulasan satu persatu. Sehingga, dibutuhkan pendekatan yang berbeda untuk memperoleh informasi dari ulasan yang tersedia. Analisis sentimen dan pemodelan topik adalah alat analisis yang dapat digunakan untuk mengidentifikasi, mengekstraksi, dan mengukur sentimen dan topik dari suatu data tekstual yang sifatnya subjektif secara cepat, efektif, murah, dan dengan memperhitungkan tema umum yang dibahas dalam suatu dokumen [3].

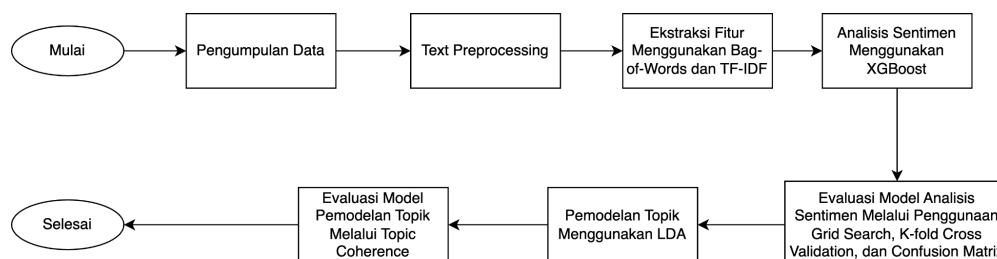
Penelitian terkait analisis sentimen dan pemodelan topik pernah dilakukan sebelumnya pada *tweet* mengenai edukasi daring selama adanya pandemi COVID-19 [4]. Pada penelitian tersebut, dilakukan perbandingan akurasi analisis sentimen melalui pendekatan *machine learning* dan

deep learning yang menghasilkan kesimpulan bahwa akurasi yang dihasilkan lebih baik pada penggunaan algoritma *machine learning* pada ukuran *dataset* yang tidak terlalu besar. Penelitian tersebut juga menggunakan metode LSA dalam melakukan pemodelan topik pada masing-masing kelas sentimen dan masing-masing menghasilkan 10 topik pembahasan. Selain itu, penelitian analisis sentimen dan pemodelan topik juga pernah dilakukan untuk mengidentifikasi kepuasan terhadap layanan pemerintah melalui metode *Support Vector Machine* dan LDA [5]. Penelitian tersebut menghasilkan akurasi analisis sentimen sebesar 75% dan 5 topik pada masing-masing kelas sentimen.

Penelitian ini bertujuan untuk menggabungkan analisis sentimen dan pemodelan topik dalam memperoleh informasi terkait dengan pengalaman pengguna dalam menggunakan aplikasi Noice. Melalui penggunaan analisis sentimen, ulasan dapat diklasifikasikan menjadi ulasan dengan sentimen positif dan negatif. Sedangkan, pemodelan topik digunakan untuk memperoleh topik yang dibahas pada masing-masing kelas sentimen. Metode analisis sentimen yang digunakan pada penelitian ini adalah XGBoost yang menghasilkan performa yang baik dalam permasalahan klasifikasi di penelitian sebelumnya [6]. XGBoost menerapkan mekanisme *decision tree* untuk membuat *weak learner* menjadi *learner* yang lebih baik. Selain itu, XGBoost adalah bentuk peningkatan dari algoritma *gradient boosting* yang menerapkan regularisasi untuk mengurangi *overfitting*. Sedangkan, metode pemodelan topik yang digunakan pada penelitian ini adalah LDA karena metode tersebut menghasilkan performa yang lebih baik jika dibandingkan dengan beberapa metode lainnya [7]. LDA adalah algoritma *unsupervised* yang digunakan untuk mengekstraksi topik dari sekumpulan dokumen serta mengasumsikan bahwa setiap dokumen terdiri atas beberapa topik, yang mana setiap topiknya merupakan distribusi probabilitas dari kata yang ada [7]. Penelitian ini akan dilakukan dengan menggunakan bahasa pemrograman Python beserta bantuan dari *library-library* yang tersedia.

2. Metode Penelitian

2.1 Alur Penelitian



Gambar 1. Alur Penelitian

Alur dari penelitian ini ditunjukkan pada Gambar 1. Tahap pertama yang dilakukan adalah pengumpulan data ulasan aplikasi Noice dari Google Play Store. Setelah itu, dilakukan *text preprocessing* untuk membersihkan data ulasan yang telah dikumpulkan. Untuk melakukan analisis sentimen dan pemodelan topik, dilakukan ekstraksi fitur terlebih dahulu untuk merepresentasikan data tekstual dalam bentuk yang dipahami oleh metode yang digunakan. Ekstraksi fitur dilakukan dengan dua metode, yaitu TF-IDF untuk analisis sentimen dan *bag-of-words* untuk pemodelan topik. Setelah itu, dilakukan proses analisis sentimen dengan menggunakan XGBoost. Kemudian, dilakukan proses evaluasi untuk menilai kinerja model analisis sentimen yang dihasilkan. Proses evaluasi dilakukan dengan menggunakan *K-fold cross validation*, *grid search*, dan *confusion matrix*. Setelah dilakukan sentimen analisis, akan dilakukan pemodelan topik pada sentimen positif dan negatif menggunakan metode LDA. Kemudian, proses evaluasi pemodelan topik dilakukan menggunakan *topic coherence* untuk menentukan jumlah topik yang merepresentasikan masing-masing kelas sentimen.

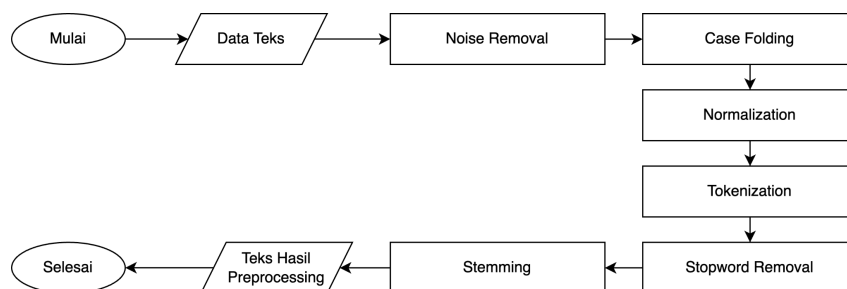
2.2 Pengumpulan Data

Data ulasan aplikasi Noice yang digunakan pada penelitian ini merupakan data sekunder yang di-*scrape* dari Google Play Store. Proses *scraping* dilakukan dengan menggunakan *library* Google-Play-Scraper. Data ulasan aplikasi Noice adalah data teks berbahasa Indonesia dengan jumlah data sebanyak 8341 ulasan. Data ulasan yang diambil adalah ulasan pada tanggal 26

November 2019 hingga 24 September 2022. Kemudian, data tersebut dilabeli dengan memanfaatkan *rating* dari masing-masing ulasan tersebut. Untuk *rating* dengan jumlah bintang 1, 2, dan 3 akan dilabeli sebagai ulasan dengan sentimen negatif. Sedangkan, ulasan dengan *rating* yang jumlah bintangnya adalah 4 dan 5 dilabeli sebagai sentimen positif. Melalui proses *labeling* tersebut, diperoleh sentimen negatif sebanyak 2055 ulasan dan sentimen positif sebanyak 6286 ulasan.

2.3 Text Preprocessing

Adapun beberapa tahapan *preprocessing* yang akan dilakukan pada penelitian ini ditunjukkan pada Gambar 2.



Gambar 2. Alur Text Preprocessing

Langkah pertama dalam *text preprocessing* yang dilakukan adalah *noise removal*, yaitu proses untuk menghilangkan tanda baca, karakter, ataupun angka [8]. Setelah melalui tahapan *noise removal*, dilakukan *case folding* untuk mengubah huruf kapital menjadi huruf kecil. Hal ini dilakukan dengan tujuan agar setiap karakter pada suatu *dataset* direpresentasikan dalam bentuk yang sama, yaitu berupa huruf kecil [9]. Setelah melakukan *case folding* dan setiap kata sudah dalam bentuk yang sama, maka *normalization* dilakukan untuk menormalisasikan kata yang bukan standar, kata yang disingkat, ataupun kata yang mengalami kesalahan penulisan [10]. Hasil dari proses *normalization* akan diterapkan *tokenization* untuk memecah suatu data tekstual menjadi bagian yang lebih kecil atau disebut dengan token. Proses ini akan memecah suatu kalimat menjadi sekumpulan kata [11]. Setelah itu, dilakukan *stopword removal* yang merupakan proses untuk memperoleh kata-kata yang penting dari hasil *tokenization* dengan menghilangkan kata-kata yang tidak memiliki arti [8]. Kemudian, dilakukan *Stemming* yang merupakan proses untuk mengubah suatu kata yang mengandung imbuhan menjadi akar katanya [12]. Proses *text preprocessing* akan dilakukan dengan menggunakan bahasa pemrograman Python.

2.4 Bag-of-Words (BOW)

Bag-of-words (BOW) adalah salah satu jenis cara untuk mengekstraksi fitur dari suatu kalimat maupun dokumen. Model BOW adalah model representasi yang akan mentransformasikan suatu data tekstual menjadi vektor dengan panjang yang telah ditentukan melalui proses perhitungan kata-kata yang muncul berulang kali. Pada representasi ini, kalimat maupun dokumen dianggap sebagai kumpulan kata yang tidak berurutan menggunakan vektor dengan ukuran yang telah ditentukan dan mengandung jumlah kemunculan kata yang mengabaikan tata bahasa maupun urutan kata. Penggunaan BOW dapat dilakukan dengan dua tahap, yaitu menentukan kata yang ada pada seluruh kalimat atau dokumen dan menentukan metode penilaian untuk kemunculan setiap kata yang telah diidentifikasi sebelumnya. Metode penilaian tersebut dapat berupa cara yang sederhana seperti menggunakan nilai biner yang menentukan kemunculan suatu kata ataupun menggunakan nilai non-biner yang menunjukkan frekuensi kemunculan kata tersebut [13]. Pada penelitian ini, hasil *text preprocessing* yang berupa kata yang telah di-*tokenize* dari suatu kalimat akan digunakan untuk menghasilkan fitur yang dapat dipahami oleh LDA melalui penerapan *bag-of-words*.

2.5 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) adalah skema pembobotan yang bermanfaat dalam ranah *information retrieval* dan *text mining*. TF-IDF memberikan informasi mengenai pentingnya suatu kata pada dokumen dari suatu *corpus* [14]. *Term frequency* (TF) pada TF-IDF menyatakan kemunculan suatu kata pada dokumen. Sehingga, nilai TF yang tinggi untuk suatu kata menandakan bahwa kata tersebut penting untuk suatu dokumen. Sedangkan,

IDF menyatakan invers dari DF yang digunakan untuk menentukan seberapa penting suatu kata pada sekumpulan dokumen. Nilai IDF yang tinggi menyatakan bahwa kemunculan kata tersebut jumlahnya sedikit sehingga dianggap merupakan kata yang penting [15]. Pada penelitian ini, hasil *text preprocessing* akan digunakan untuk menghasilkan fitur yang dapat digunakan oleh model *machine learning* melalui penggunaan TF-IDF. Untuk suatu kata t pada dokumen d , nilai TF-IDF dapat dihitung melalui persamaan berikut:

$$TF(t, d) = \frac{\text{Jumlah kemunculan kata } t \text{ pada dokumen } d}{\text{Jumlah kata pada dokumen } d} \quad (1)$$

$$IDF(t) = \log\left(\frac{\text{Jumlah dokumen}}{\text{Jumlah dokumen yang mengandung kata } t}\right) \quad (2)$$

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

2.6 Extreme Gradient Boosting (XGBoost)

Extreme gradient boosting (XGBoost) adalah suatu algoritma *machine learning* yang diaplikasikan untuk permasalahan klasifikasi dan regresi. XGBoost merupakan algoritma yang berupa peningkatan dari algoritma *gradient boosting*. XGBoost menerapkan mekanisme *decision tree* untuk membuat *weak learner* menjadi *learner* yang lebih baik. Peningkatan yang dilakukan pada XGBoost adalah penggunaan regularisasi yang bertujuan untuk mengontrol serta mengurangi terjadinya *overfitting* pada model yang dihasilkan agar model tersebut dapat memberikan performa yang lebih baik [16]. Pada penerapan XGBoost dalam analisis sentimen, XGBoost akan menggunakan hasil ekstraksi TF-IDF yaitu matriks dengan bobot kata sebagai fiturnya. Adapun langkah-langkah proses klasifikasi dengan algoritma XGBoost sebagai berikut:

1. Melakukan prediksi awal

Langkah pertama dalam XGBoost adalah membuat prediksi awal, yaitu setiap objek data memiliki probabilitas sebesar 0,5 untuk diklasifikasikan ke dalam suatu kelas [17].

2. Menghitung *residual*

Setelah melakukan prediksi awal, maka akan dilakukan pembuatan *tree* berdasarkan *residual* dari prediksi awal [17]. *Residual* dapat dihitung melalui persamaan (4).

$$\text{Residual} = \text{nilai observasi} - \text{nilai prediksi} \quad (4)$$

3. Menghitung *gain*

Sebelum menghitung *gain*, perlu dihitung nilai *similarity score* dari setiap *node* melalui persamaan (5) [17].

$$\text{Similarity score} = \frac{(\sum r_i)^2}{\sum |p \times (1-p)| + \lambda} \quad (5)$$

Keterangan:

r_i = residual ke- i , p = probabilitas sebelumnya, λ = parameter regularisasi

Setelah menghitung nilai *similarity score*, maka nilai *gain* dapat dihitung melalui persamaan (6). Fitur dan *split point* terbesar dari fitur tersebut akan dipilih untuk melakukan *splitting* apabila diperoleh nilai *gain* terbesar [17].

$$\text{Gain} = \text{Left Similarity} + \text{Right Similarity} - \text{Root Similarity} \quad (6)$$

4. Menghitung *cover*

Nilai *cover* digunakan untuk menentukan jumlah minimum *residual* pada *leaf* dari suatu *tree* [17]. Nilai *cover* dapat dihitung melalui persamaan (7).

$$\text{Cover} = \sum |p \times (1-p)| \quad (7)$$

5. Menghitung probabilitas kelas

Untuk memprediksi probabilitas kelas dari suatu objek data, maka perlu dihitung *output* dari *node*-nya melalui persamaan (8) [17].

$$Output = \frac{Residuals}{\sum |(p)x(1-p)|+\lambda} \quad (8)$$

Setelah itu, nilai prediksi probabilitas awal yang didefinisikan pada langkah pertama perlu diubah menjadi bentuk $\log(\text{odds})$. Untuk itu, digunakan persamaan (9) [17].

$$\log(\text{odds}) = \log\left(\frac{P}{1-P}\right) \quad (9)$$

Kemudian, akan dihitung $\log(\text{odds})$ dari objek data melalui persamaan (10) [17].

$$Log(\text{odds})Prediction = Log(\text{odds}) + (\varepsilon \times Output) \quad (10)$$

Untuk memperoleh probabilitas kelas dari objek data tersebut, dapat dikonversi kembali $\log(\text{odds})$ tersebut menjadi probabilitas melalui persamaan (11) [17].

$$Probability = \frac{e^{\log\log(\text{odds})}}{1+e^{\log\log(\text{odds})}} \quad (11)$$

Keterangan:

P = prediksi, ε = learning rate

Probabilitas tersebut merupakan probabilitas peluang untuk tergolong ke dalam suatu kelas. Proses perhitungan akan diulang kembali dari langkah ke-2 untuk melanjutkan pembuatan *tree* lainnya [17].

2.7 K-fold Cross Validation

K-fold cross validation metode statistik yang digunakan untuk mempartisi suatu data menjadi bagian *training* dan *testing*. Metode ini digunakan untuk membagi data secara berulang menjadi dua bagian, yaitu bagian *training* dan bagian *testing*. Yang mana, setiap data memiliki peluang untuk menjadi data *testing* [18]. *K-fold cross validation* diterapkan dalam menentukan *hyperparameter* model analisis sentimen untuk memperoleh rata-rata nilai evaluasi yang diujikan dalam berbagai kondisi data.

2.8 Grid Search

Grid search adalah metode yang digunakan untuk mengidentifikasi parameter optimal bagi suatu model agar model tersebut dapat memprediksi data tidak berlabel dengan akurat. Melalui *grid search*, proses pencarian parameter yang sesuai bagi suatu model dapat dilakukan dengan lebih mudah. Metode *grid search* digunakan dengan mendefinisikan rentang nilai suatu *hyperparameter*. Kemudian, *grid search* akan membuat model berdasarkan kombinasi dari rentang nilai yang telah didefinisikan sebelumnya, yang mana rentang *hyperparameter* tersebut disebut sebagai *grid* [19]. Metode ini diterapkan pada proses *training* model analisis sentimen.

2.9 Confusion Matrix

Untuk melakukan evaluasi terhadap model analisis sentimen yang dihasilkan, yaitu dalam penelitian ini merupakan model XGBoost, akan digunakan *confusion matrix*. *Confusion matrix* digunakan untuk menentukan efektifitas pemodelan permasalahan klasifikasi [20]. *Confusion matrix* akan digunakan untuk menilai hasil prediksi dari model analisis sentimen terhadap data ulasan yang diberikan kepada model untuk menilai kinerja dari model tersebut. Adapun bentuk dari *confusion matrix* ditunjukkan pada Tabel 1.

Tabel 1. Confusion Matrix

	Hasil Prediksi	
Positif		Negatif

Kelas Seharusnya	Positif	TP (<i>True Positive</i>)	FN (<i>False Negative</i>)
	Negatif	FP (<i>False Positive</i>)	TN (<i>True Negative</i>)

Dari *confusion matrix*, dapat diperoleh empat metrik evaluasi yaitu akurasi, *precision*, *recall*, dan *F1-score*. Keempat metrik evaluasi tersebut merupakan metrik yang banyak digunakan [21]. *Confusion matrix* diterapkan pada hasil prediksi dari model analisis sentimen untuk memperoleh akurasi, *precision*, *recall*, dan *F1-score* dari model analisis sentimen. Adapun persamaan yang digunakan untuk menghitung nilai akurasi, *precision*, *recall*, dan *F1-score* adalah sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP + TN}{TP + FN} \quad (14)$$

$$F1 - Score = \frac{2 \times recall \times precision}{recall + precision} \quad (15)$$

2.10 Latent Dirichlet Allocation (LDA)

LDA adalah model probabilistik yang dikenal sebagai salah satu algoritma pemodelan topik yang populer [7] LDA dapat digunakan untuk memahami arti semantik dari suatu teks sehingga dapat mengidentifikasi topik utamanya [22]. LDA menggunakan asumsi dasar bahwa setiap topik direpresentasikan oleh kata yang serupa dan setiap dokumen dapat direpresentasikan oleh beberapa topik. LDA bertujuan untuk memetakan setiap dokumen ke suatu topik yang mengandung kata-kata pada dokumen tersebut. Karena LDA menganggap setiap dokumen sebagai model *bag-of-words*, maka urutan kata atau perubahan konten pada dokumen tidak perlu dilakukan [23]. Pada penelitian ini, LDA akan diterapkan pada representasi ulasan yang dihasilkan melalui penerapan *bag-of-words*.

2.11 Topic Coherence

Topic coherence digunakan untuk menghitung nilai dari satu topik dengan menghitung derajat kemiripan dari aspek semantik antara kata-kata dengan hasil *scoring* yang tinggi pada suatu topik. Hasil perhitungan tersebut digunakan untuk membedakan topik yang dapat dipahami secara semantik dan topik yang hanya berupa artifak dari *statistical inference* [24]. Terdapat berbagai *coherence measure* yang digunakan dalam menghitung *topic coherence*, beberapa diantaranya adalah *c_v*, *c_p*, *c_uci*, *c_umass*, *c_npmi*, dan *c_a*. Pada penelitian ini, digunakan *c_v* yang berbasis *sliding window*, *one-set segmentation* terhadap kata-kata dengan *scoring* yang tinggi, dan *indirect confirmation measure* yang menggunakan *normalized pointwise mutual information* (NPMI) dan *cosine similarity* [25]. Pada penelitian ini, *topic coherence* diterapkan pada model LDA yang telah dihasilkan untuk menentukan jumlah topik terbaik berdasarkan nilai *topic coherence* yang dihasilkan.

3. Hasil dan Pembahasan

3.1 Text Preprocessing

Tahap pertama yang dilakukan pada *dataset* adalah menerapkan *text preprocessing* pada seluruh ulasan yang ada di *dataset* yang telah dikumpulkan. Adapun proses *text preprocessing* yang dilakukan ditunjukkan pada Tabel 2.

Tabel 2. Tahap *Text Preprocessing*

Langkah <i>Preprocessing</i>	Teks
Kondisi Awal Teks	Aplikasi yg menarik, bisa mendengar cerita" yg seru juga
<i>Noise Removal</i>	Aplikasi yg menarik bisa mendengar cerita yg seru juga
<i>Case Folding</i>	aplikasi yg menarik bisa mendengar cerita yg seru juga
<i>Normalization</i>	aplikasi yang menarik bisa mendengar cerita yang seru juga
<i>Tokenization</i>	['aplikasi', 'yang', 'menarik', 'bisa', 'mendengar', 'cerita', 'yang', 'seru', 'juga']
<i>Stopword Removal</i>	['aplikasi', 'menarik', 'mendengar', 'cerita', 'seru']
<i>Stemming</i>	['aplikasi', 'tarik', 'dengar', 'cerita', 'seru']

Pada penelitian ini, proses *text preprocessing* dilakukan dengan menggunakan bahasa pemrograman Python. Proses ini dilakukan dengan memanfaatkan *library* NLTK untuk melakukan *tokenizing* dan *stopword removal* serta *library* Sastrawi untuk melakukan *stemming*. Untuk proses normalisasi, penulis menggunakan kamus kata yang digunakan pada penelitian sebelumnya [26]. Setelah menerapkan keseluruhan tahapan *text preprocessing* pada 8341 ulasan, akan dilakukan proses penghapusan data yang hasil *text preprocessing*-nya tidak lagi menyisakan kata apapun. Dari hasil penghapusan data tersebut, diperoleh data sebanyak 8155 ulasan yang akan diolah pada tahapan berikutnya, yang mana sebanyak 2034 dari data tersebut adalah ulasan negatif dan sebanyak 6121 adalah ulasan positif.

3.2 Ekstraksi Fitur

Pada penelitian ini, proses ekstraksi fitur pada *dataset* yang telah dilakukan *preprocessing* akan dilakukan dengan menggunakan TF-IDF dan *bag-of-words*. Hasil ekstraksi fitur menggunakan TF-IDF akan dijadikan sebagai *input* dari model analisis sentimen. Dalam proses pemodelan topik, LDA akan menggunakan *document-term matrix* (DTM) yang disusun menggunakan *bag-of-words* dan *dictionary* yang mengandung setiap kata pada *corpus*. Hasil penerapan TF-IDF dapat dilihat pada gambar 3, yaitu berupa *sparse matrix* yang berisikan bobot TF-IDF dengan jumlah baris sebanyak jumlah ulasan dan jumlah kolom sebanyak jumlah kata yang menjadi fiturnya. Sedangkan, hasil penerapan ekstraksi fitur dapat dilihat pada gambar 4. Pada hasil penerapan *bag-of-words* dengan menggunakan *library* Gensim, diperoleh *list* yang berisikan pasangan indeks dari suatu kata beserta jumlah kemunculannya pada setiap kalimat atau ulasan pada *dataset* yang digunakan.

```
[41]: print("bentuk matriks: ", tf_x_train.shape)
      print("matriks: ", tf_x_train)
bentuk matriks: (6524, 4337)
matriks: (0, 2761) 0.6946802473480708
(0, 3563) 0.719318673429533
(1, 1881) 0.1558024387418681
(1, 3946) 0.17302712753359992
(1, 2830) 0.13007074235777372
(1, 260) 0.30601061614313796
(1, 3871) 0.19810649152894022
(1, 390) 0.33909292564436494
(1, 679) 0.24207102258848268
```

Gambar 3. Hasil Ekstraksi TF-IDF

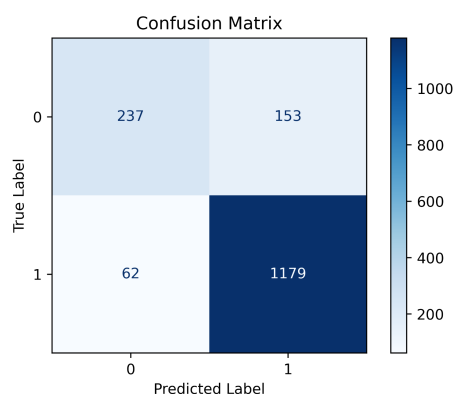
```
[47]: review_dtm
[47]: [[(0, 1)],
      [(1, 1)],
      [(2, 1)],
      [(3, 1), (4, 1), (5, 1)],
      [(6, 1), (7, 1), (8, 1), (9, 1)],
      [(10, 1),
      (11, 1),
      (12, 1),
      (13, 1),
      (14, 1),
      (15, 1),
      (16, 1),
      (17, 1)].
```

Gambar 4. Hasil Ekstraksi *Bag-of-Words*

3.3 Analisis Sentimen Dengan XGBoost

Setelah melalui fase *text preprocessing* dan pembobotan, analisis sentimen dapat dilakukan dengan menggunakan hasil ekstraksi fitur TF-IDF. Proses analisis sentimen akan dilakukan dengan menggunakan algoritma XGBoost melalui *library* XGBoost. Karena XGBoost memiliki banyak *hyperparameter*, maka akan digunakan *grid search* beserta *K-fold cross validation* untuk menentukan kombinasi *hyperparameter* yang dapat memberikan akurasi paling optimal. *Hyperparameter* yang akan disesuaikan pada penelitian ini adalah tiga dari tujuh *hyperparameter* yang dinyatakan memiliki pengaruh yang signifikan terhadap performa model XGBoost, yaitu “*max_depth*” yang berkaitan dengan kedalaman maksimum dari *tree*, “*learning_rate*” yang berkaitan dengan *step size*, dan “*max_leaves*” yang berkaitan dengan jumlah maksimum *node* yang ditambahkan [27]. Untuk *hyperparameter* “*max_depth*”, akan digunakan rentang nilai 3, 4, 5, 7, dan 9. Untuk *hyperparameter* “*learning_rate*”, akan digunakan rentang nilai 0,1, 0,3, dan 0,5. Untuk *hyperparameter* “*max_leaves*”, akan digunakan rentang nilai 0, 3, 5, 7, dan 10.

Dataset akan dibagi menjadi dua bagian, yaitu data *training* sebesar 80% dan data *testing* sebesar 20%. Proses *grid search* dan *K-fold cross validation* akan dilakukan terlebih dahulu pada data *training* untuk memperoleh kombinasi *hyperparameter* terbaik berdasarkan akurasi dari model yang dihasilkan. Proses *grid search* dan *K-fold cross validation* dilakukan dengan menggunakan *GridSearchCV* pada *library scikit-learn*. Pada proses tersebut, model XGBoost dengan performa terbaik dihasilkan dengan menggunakan kombinasi *hyperparameter* *learning_rate* bernilai 0,3, *max_depth* bernilai 7, dan *max_leaves* bernilai 0. Pada kombinasi *hyperparameter* tersebut, diperoleh akurasi sebesar 86,1% dengan menggunakan 5 *fold* pada data *training*. *Confusion matrix* dari model dengan kombinasi *hyperparameter* terbaik ditunjukkan pada Gambar 5. Pada model tersebut, diperoleh akurasi, *precision*, *recall*, dan *F1 score* yang berturut-turut bernilai 86,8%, 83,9%, 77,9%, dan 80,2%. Adapun hasil klasifikasi dari model yang dihasilkan dan dapat dilihat pada tabel 3.



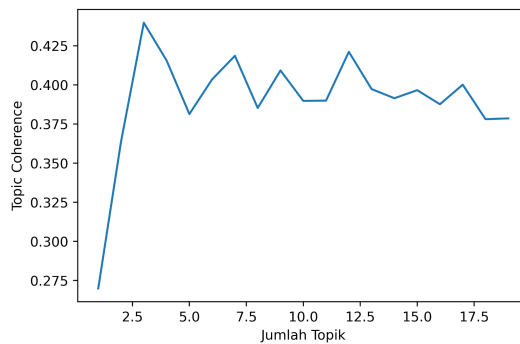
Gambar 5. *Confusion Matrix* Dari Model Yang Dihasilkan

Tabel 3. Hasil Klasifikasi

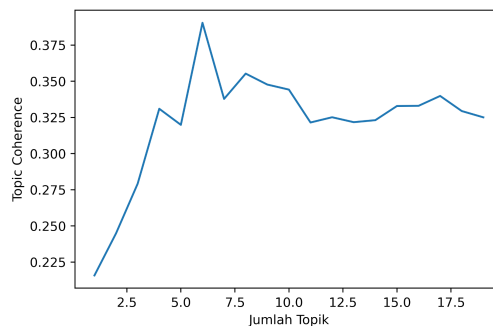
Ulasan	Sentimen
Gabisa dipake, "Aduh error" katanya.	Negatif
bagus isi podcastnya	Positif

3.4 Pemodelan Topik Dengan LDA

Pada penelitian ini, pemodelan topik melalui LDA dilakukan pada masing-masing sentimen ulasan, yaitu ulasan dengan sentimen positif dan ulasan dengan sentimen negatif. Hal tersebut dilakukan untuk memperoleh informasi mengenai kelebihan maupun kekurangan dari aplikasi Noice melalui topik-topik yang dibahas pada masing-masing kelas sentimen. Dalam menentukan jumlah topik yang sesuai, digunakan *topic coherence* untuk menentukan jumlah topik yang ideal pada masing-masing kelas sentimen. Pada tahap ini, penulis menggunakan *library* Gensim untuk mengimplementasikan LDA.



Gambar 6. Perbandingan *Topic Coherence* Pemodelan LDA Pada Sentimen Positif



Gambar 7. Perbandingan *Topic Coherence* Pemodelan LDA Pada Sentimen Negatif

Ditunjukkan pada gambar 6, pada ulasan dengan sentimen positif, jumlah topik yang menghasilkan *topic coherence* terbaik adalah sebanyak tiga topik dengan *topic coherence* yang bernilai 0,439. Untuk ulasan dengan sentimen negatif, ditunjukkan pada gambar 7 bahwa jumlah topik yang menghasilkan *topic coherence* terbaik adalah sebanyak enam topik dengan *topic coherence* yang bernilai 0,390. Hasil penerapan pemodelan topik pada ulasan dengan sentimen positif dan negatif ditunjukkan pada Tabel 4 dan 5.

Tabel 4. Hasil Pemodelan Topik Pada Ulasan Dengan Sentimen Positif

Topik	Kata
1	"noice", "keren", "bagus", "podcast", "nya", "banget", "dengar", "aplikasi", "suka", "radio"
2	"mantap", "aplikasi", "hiya", "keren", "banget", "pokok", "nya", "seru", "gara", "wow"
3	"cok", "muslim", "dengar", "podcast", "musuh", "masyarakat", "unduh", "aplikasi", "radio", "good"

Pada ulasan dengan sentimen positif, topik pertama dapat diinterpretasikan sebagai topik yang membahas kepuasan pengguna terhadap aplikasi Noice beserta konten *podcast* dan radionya melalui kata “keren”, “suka”, ataupun “bagus”. Pada topik kedua, diinterpretasikan bahwa topik tersebut membahas mengenai kepuasan pengguna terhadap aplikasi Noice melalui kata “mantap”, “keren”, ataupun “seru” serta alasan yang menyebabkan pengguna menggunakan aplikasi Noice. Pada topik ketiga, diinterpretasikan bahwa topik tersebut membahas mengenai pengguna yang menggunakan aplikasi Noice untuk mendengarkan konten “Musuh Masyarakat” yang dibawakan oleh Coki Pardede dan Tretan Muslim.

Tabel 5. Hasil Pemodelan Topik Pada Ulasan Dengan Sentimen Negatif

Topik	Kata
1	"mati", "bagus", "dengar", "radio", "bikin", "malas", "politik", "screen", "podcastnya", "kayak"
2	"aplikasi", "noice", "ya", "nya", "tolong", "dengar", "baik", "detik", "buka", "suka"
3	"musik", "baik", "putar", "dengar", "app", "tolong", "unduh", "error", "putus", "podcast"
4	"lagu", "error", "aplikasi", "nya", "podcast", "baru", "dengar", "banget", "koneksi", "buka"
5	"cok", "muslim", "podcast", "suka", "mulu", "dengar", "gue", "koneksi", "tingkat", "mati"
6	"podcast", "nya", "putar", "nih", "kuota", "jam", "payah", "makan", "hidup", "habis"

Pada ulasan dengan sentimen negatif, topik pertama membahas mengenai permasalahan aplikasi Noice yang mati sendiri ketika sedang digunakan serta hal-hal yang menyebabkan pengguna merasa malas dalam menggunakan aplikasi Noice. Pada topik kedua, diinterpretasikan bahwa topik ini membahas mengenai permintaan perbaikan permasalahan yang ada pada aplikasi beserta masalah-masalah yang sering dihadapi pengguna baik permasalahan dalam membuka aplikasi maupun dalam pemutaran kontennya. Pada topik ketiga, diinterpretasikan bahwa pengguna merasa kurangnya koleksi serta permasalahan konten musik pada Aplikasi Noice, adanya *error*, permasalahan dalam mengunduh aplikasi ataupun konten, serta adanya permasalahan konten yang putus-putus ketika pengguna mendengarkan suatu konten. Pada topik keempat, diinterpretasikan bahwa topik ini membahas mengenai kurangnya koleksi serta permasalahan lagu di aplikasi Noice, keluhan terkait *error* di aplikasi Noice, permasalahan dalam membuka aplikasi, dan permasalahan koneksi dalam mendengarkan konten di aplikasi Noice. Pada topik kelima, diinterpretasikan bahwa pengguna mengunduh aplikasi hanya karena adanya konten milik Coki Pardede dan Tretan Muslim, adanya

permasalahan koneksi ketika mengakses konten pada aplikasi Noice, serta permintaan untuk meningkatkan aplikasi Noice agar menjadi lebih baik. Pada topik keenam, diinterpretasikan bahwa topik tersebut membahas mengenai permasalahan pemutaran konten, pengiriman kode verifikasi yang lama, serta aplikasi Noice yang banyak memakan kuota.

4. Kesimpulan

Dari 8341 ulasan aplikasi Noice yang diperoleh melalui Google Play Store, sebanyak 6286 dari ulasan tersebut merupakan sentimen positif dan sebanyak 2055 merupakan sentimen negatif. Dari proses analisis sentimen yang dilakukan, diperoleh model XGBoost yang mampu mengklasifikasikan sentimen pada ulasan aplikasi Noice dengan akurasi, *precision*, *recall*, dan *F1-score* yang bernilai 86,8%, 83,9%, 77,9%, dan 80,2%. Model XGBoost tersebut menggunakan *hyperparameter* *learning_rate* yang bernilai 0,3, *max_depth* yang bernilai 7, dan *max_leaves* yang bernilai 0.

Berdasarkan pemodelan topik yang telah dilakukan pada ulasan dengan sentimen positif dan negatif melalui LDA, diperoleh jumlah topik yang menghasilkan *topic coherence* terbaik pada ulasan dengan sentimen positif yaitu sebanyak 3 topik. Sedangkan, jumlah topik yang menghasilkan *topic coherence* terbaik pada ulasan dengan sentimen negatif adalah sebanyak 6 topik. Pada sentimen positif, ulasan umumnya berisikan pujian terhadap aplikasi, konten yang disukai oleh penulis ulasan, serta alasan pengguna mengunduh aplikasi Noice. Sedangkan, topik yang dibahas pada sentimen negatif membahas mengenai permasalahan yang lebih spesifik seperti permasalahan koneksi, permasalahan pembukaan aplikasi, permasalahan pemutaran atau pengunduhan konten, penggunaan kuota, pengiriman kode verifikasi, ataupun aplikasi yang mati dengan sendirinya.

Referensi

- [1] Sortlist, "COVID-19 Boosts Audio Consumption by 76.2%: 2021 Survey", 26 Agustus 2022. [Online]. Tersedia: <https://www.sortlist.com/datahub/reports/covid-19-boosts-audio-consumption-by-76-2-2021-survey/>. 29 September 2022.
- [2] IndoTelko, "Noice masuk 50 besar startup TOP Indonesia versi Tech In Asia", 22 Februari 2022. Tersedia: <https://www.indotelko.com/read/1645464215/noice-masuk-50-besar-startup-top-indonesia-versi-tech-in-asia>. 1 Oktober 2022.
- [3] C. A. Melton, O. A. Olusanya, N. Ammar, dan A. Shaban-Nehad, "Public sentiment analysis and topic modeling regarding COVID-19 vaccines on the Reddit social media platform: A call to action for strengthening vaccine confidence," *Journal of Infection and Public Health*, vol. 14, no. 10, 2021.
- [4] M. Muhajid, E. Lee, F. Rustam, P. B. Washington, S. Ullah, A. A. Reshi, dan I. Ashraf, "Sentiment Analysis and Topic Modeling on Tweets about Online Education during COVID-19," *Applied Sciences*, vol. 11, no. 18, 2021.
- [5] M. N. Aziz, A. Firmanto, A. M. Fajrin, dan R. V. Hari Ginardi, "Sentiment Analysis and Topic Modelling for Identification of Government Service Satisfaction," *2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 125-130, 2018.
- [6] H. Ullah, B. Ahmad, I. Sana, A. Sattar, A. Khan, S. Akbar, dan M. Z. Asghar, "Comparative study for machine learning classifier recommendation to predict political affiliation based on online reviews," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 3, p. 251– 264, 2021.
- [7] R. Albalawi, T. H. Yeap, M. Benyoucef, "Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis", *Frontiers in Artificial Intelligence*, vol. 3, no. 42, 2020.
- [8] Y. D. Kirana dan S. A. Faraby, "Sentiment Analysis of Beauty Product Reviews Using the K-Nearest Neighbor (KNN) and TF-IDF Methods with Chi-Square Feature Selection," *Journal of Data Science and Its Applications*, vol. 4, no. 1, p. 31 - 42, 2021.
- [9] A. R. L. Pratama dan W. Astuti, "Implementation of Enhance Confix Stripping Stemmer Algorithm for Multiclass Dataset Classification in News Text using K-Nearest Neighbor," *Journal of Data Science and Its Applications*, vol. 4, no. 1, p. 11 - 17, 2021.

- [10] P. Z. Amalia dan E. Winarko, "Aspect-Based Sentiment Analysis on Indonesian Restaurant Review Using a Combination of Convolutional Neural Network and Contextualized Word Embedding," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 3, p. 285 - 294, 2021.
- [11] H. S. Hota, D. K. Sharma, dan N. Verma, "Lexicon-based sentiment analysis using Twitter data: a case of COVID-19 outbreak in India and abroad," *Data Science for COVID-19*, p. 275 - 295, 2021.
- [12] W. Bourequat dan H. Mourad, "Sentiment Analysis Approach for Analyzing iPhone Release using Support Vector Machine," *International Journal of Advances in Data and Information Systems*, vol. 4, no. 1, p. 36 - 44, 2021.
- [13] A. Beheshti, S. Ghodrathnama, M. Elahi, dan H. Farhood, *Social Data Analytics*, Florida: CRC Press, 2022, p. 67.
- [14] S. Kalra, L. Li, dan H. R. Tizhoosh, "Automatic Classification of Pathology Reports using TF-IDF Features," *arXiv preprint arXiv:1903.07406*, 2019.
- [15] SW. Kim dan JM. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Human-centric Computing and Information Sciences*, vol. 9, no. 30, 2019.
- [16] A. M. AL-Shatnwai dan M. Faris, "Predicting Customer Retention using XGBoost and Balancing Methods," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 11, no. 7, 2020
- [17] H. Manoharan dan Dr. J. Dhilipan, "Diabetes Mellitus Prediction on Class Balanced Data Set using XGBoost Algorithms," *Journal of Xi'an Shiyou University, Natural Science Edition*, vol. 18, no. 1, 2022.
- [18] N. Qomariah, "Sentiment Analysis on Coffee Consumer Perceptions on Social Media Twitter Using Multinomial Naïve Bayes" *Journal of Intelligent Computing and Health Informatics*, vol. 2, no. 1, 2021.
- [19] S. G. C G dan B. Sumathi, "Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, 2020.
- [20] A. Nurkholis, D. Alita, A. Munandar, "Comparison of Kernel Support Vector Machine Multi-Class in PPKM Sentiment Analysis on Twitter," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 6, no. 2, p. 227 - 233, 2022.
- [21] C. Sitaula dan T. B. Shahi, "Multi-channel CNN to classify nepali covid-19 related tweets using hybrid features," *arXiv preprint arXiv:2203*, 2022.
- [22] G. Papadia, M. Pacella, V. Giliberti, "Topic Modeling for Automatic Analysis of Natural Language: A Case Study in an Italian Customer Support Center," *Algorithms*, vol. 15, no. 6, 2022.
- [23] T. D. Tandjung dan D. H. Fudholi, "Topic Modeling with Latent-Dirichlet Allocation for the Discovery of State-of-the-Art in Research: A Literature Review," *Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology*, vol. 54, no. 8, 2022.
- [24] S. Devi, C. Dhavale, L. Moharkar, S. Khanvilkar, "Impact of Online Education and Sentiment Analysis from Twitter Data using Topic Modeling Algorithms," *International Journal of Applied Sciences and Smart Technologies*, vol. 4, no. 1, p. 21 - 34, 2022.
- [25] S. Mifrah dan E. H. Benlahmar, "Topic Modeling Coherence: A Comparative Study between LDA and NMF Models using COVID'19 Corpus," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, p. 5756 - 5761, 2020.
- [26] M. O. Ibrohim dan I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter," *Association for Computational Linguistics*, p. 46 - 57.
- [27] Y. Wang dan X. S. Ni, "A XGBoost risk model via feature selection and Bayesian hyperparameter optimization," *arXiv preprint arXiv:1901.08433*, 2019.