

Analisis Sentimen Destinasi Wisata Kuliner di Twitter Menggunakan TF-IDF dan Complement Naïve Bayes pada Dataset Tidak Seimbang

Fakhrana Kurnia Sutrisno¹, Jondri², Kemas Muslim Lhaksana³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹fakhranakurnia@student.telkomuniversity.ac.id, ²jondri@telkomuniversity.ac.id,

³kemasmuslim@telkomuniversity.ac.id

Abstrak

Opini masyarakat terhadap suatu destinasi wisata kuliner sangat bermanfaat bagi pemilik maupun pengunjung tempat tersebut. Maka dari itu dilakukan analisis sentimen terhadap destinasi wisata kuliner di Kota Bandung, yaitu Cuanki Serayu dan Sate DJ. Analisis sentimen diawali dengan mengambil data dari Twitter dan dilabeli secara manual menjadi positif, netral, dan negatif. Data yang sudah dilabeli dilakukan *preprocessing* dan *oversampling* pada data yang tidak seimbang. Dataset dibagi menjadi data *train* dan data *test* dengan perbandingan 70:30. Pelatihan data dilakukan menggunakan metode Complement Naïve Bayes dengan ekstraksi fitur TF-IDF. Dari hasil pengujian diperoleh nilai *f1-score* terbesar sebesar 0,80 dari data yang telah dilakukan *oversampling*.

Kata kunci: analisis sentimen, *oversampling*, TF-IDF, Complement Naïve Bayes, *f1-score*

Abstract

Public's opinion on a culinary tourism destination is very helpful for the owners and the tourists. Therefore, sentiment analysis was performed on culinary tourism destinations in Bandung, namely Cuanki Serayu and Sate DJ. The process begins with crawling the data from Twitter and manually labelling it as positive, neutral, and negative. The data that has been labeled are preprocessed and undersampled on the unbalanced data. Dataset is divided into train data and test data with a ratio of 70:30. Data training was performed using the Complement Naïve Bayes method with TF-IDF feature extraction. From the test results, the greatest *f1-score* value is 0.80 from the data that has been oversampled.

Keywords: sentiment analysis, *oversampling*, TF-IDF, Complement Naïve Bayes, *f1-score*

1. Pendahuluan

1.1. Latar Belakang

Dalam berwisata, banyak hal yang dapat dilakukan. Beberapa contohnya adalah wisata bahari, wisata budaya, wisata Pendidikan, wisata alam, dan wisata kuliner. Wisata kuliner adalah wisata dengan objek tujuannya adalah kuliner. Wisata kuliner memiliki tujuan mencicipi atau mencoba kuliner yang terkenal di daerah yang dituju. Salah satu contoh daerahnya adalah Kota Bandung. Kota Bandung memiliki banyak destinasi wisata kuliner, beberapa contohnya adalah Sate DJ dan Cuanki Serayu. Kedua destinasi wisata kuliner tersebut ramai didatangi pengunjung. Tidak sedikit orang yang menyebut kedua destinasi wisata kuliner tersebut pada media sosial Twitter.

Twitter merupakan layanan *micro blog* yang memiliki aspek jejaring sosial. Pengguna Twitter dapat mengirim pesan secara instan yang disebut dengan *tweet* dengan batasan maksimal 140 karakter dalam satu *tweet*. Selain itu, pengguna Twitter dapat melihat *tweet* mengenai topik-topik tertentu [1].

Di Indonesia, Twitter menjadi media sosial ketiga yang paling banyak digunakan [2]. Banyak orang yang memanfaatkan Twitter untuk bercerita mengenai kesehariannya maupun mengeluarkan opini pribadi mereka. Contohnya adalah opini mengenai destinasi wisata kuliner, yaitu Sate DJ dan Cuanki Serayu.

Opini bersifat subjektif dan berisi mengenai sentimen terhadap suatu topik. Analisis sentimen merupakan analisis sentimen yang ada pada sebuah opini pada sebuah topik atau permasalahan [3]. Dari analisis sentimen dapat diketahui apakah opini tersebut bernilai positif, negatif, atau netral. Analisis sentimen sudah banyak dilakukan dalam penelitian, contohnya adalah analisis sentimen terhadap calon gubernur DKI Jakarta 2017 dengan menggunakan metode Naïve Bayes dan Support Vector Machine [4]. Dalam penelitian tersebut, nilai akurasi menggunakan metode klasifikasi Naïve Bayes memiliki nilai tertinggi. Oleh karena itu, pada penelitian ini analisis sentimen dilakukan menggunakan ekstraksi fitur TF-IDF dan *Naïve Bayes Classifier*. Complement Naïve Bayes digunakan karena memiliki hasil yang baik dibandingkan metode Naïve Bayes lainnya seperti pada penelitian yang telah dilakukan A. Anagaw dan Y. L. Chang [5]. Pada penelitian tersebut dilakukan perbandingan metode Complement Naïve Bayes dengan metode Naïve Bayes yang lainnya. Hasil dari penelitian tersebut menunjukkan bahwa Complement Naïve Bayes memiliki hasil yang baik pada data yang seimbang maupun tidak seimbang.

Metode TF-IDF untuk ekstraksi fitur dipilih karena TF-IDF menghitung bobot kata yang muncul pada dokumen. Selain itu, metode TF-IDF mudah dan efisien untuk membantu analisis sentimen. Evaluasi model dilakukan menggunakan Grid Search Cross Validation.

1.2. Topik dan Batasannya

Topik yang dibahas dalam penelitian ini adalah pengaruh *oversampling* dan performa yang dihasilkan dari *oversampling* dataset pada data yang tidak seimbang. Batasan dari penelitian ini adalah dataset diambil dari Twitter dengan kata kunci berupa nama destinasi wisata kuliner terkenal di Kota Bandung, yaitu “sate dj” dan “cuanki serayu”. Pengambilan data Twitter dilakukan menggunakan bahasa pemrograman Python dengan library Tweepy. Dataset yang diperoleh sejumlah 5754 data.

1.3. Tujuan

Tujuan dilakukannya penelitian ini adalah untuk melakukan analisis sentimen opini masyarakat terhadap destinasi wisata kuliner pada media sosial Twitter dengan membandingkan pengaruh dan performa dari skenario-skenario *oversampling* pada dataset yang tidak seimbang menggunakan metode TF-IDF dan Complement Naïve Bayes.

1.4. Organisasi Tulisan

Setelah Bagian 1 yang berisi pendahuluan, jurnal ini dilanjutkan dengan Bagian 2 yang berisi studi terkait jurnal ini. Selanjutnya adalah Bagian 3 yang menjelaskan sistem yang dibangun dalam penelitian ini. Pada Bagian 4 dijelaskan hasil dan evaluasi hasil pengujian dari sistem yang dibangun. Setelah itu, dijelaskan kesimpulan dari penelitian pada Bagian 5.

2. Studi Terkait

2.1. Analisis Sentimen

Analisis sentimen merupakan proses klasifikasi yang mengidentifikasi sentimen pada sebuah teks kemudian menganalisisnya. Analisis sentimen dapat digunakan pada *review* produk, artikel berita, maupun debat politik [6].

Penelitian mengenai analisis sentimen sudah banyak dilakukan sebelumnya, beberapa contohnya adalah Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter [4], Analisis Sentimen dan Klasifikasi Kategori terhadap Tokoh Publik pada Twitter [7], dan Perancangan Sistem Analisis Sentimen Komentar Pelanggan Menggunakan Metode Naive Bayes Classifier [8].

Pada penelitian analisis sentimen terhadap calon gubernur DKI Jakarta 2017, digunakan metode Naive Bayes dan Support Vector Machine. Pengambilan data dilakukan dengan menggunakan tiga kata kunci; AHY, Ahok, dan Anies. Data yang diambil dari masing-masing kata kunci berjumlah 100 data. Hasil akhir dari penelitian tersebut memiliki nilai rata-rata akurasi mencapai 95%, nilai presisi 95%, dan nilai recall 95% untuk klasifikasi data AHY menggunakan Naive Bayes Classifier. Sedangkan untuk metode Support Vector Machine terhadap calon AHY didapat nilai akurasi 90%, nilai presisi 89,9%, dan nilai recall 90%. Nilai akurasi terendah didapat dari metode Naive Bayes pada data dengan kata kunci Ahok dan Anies [4].

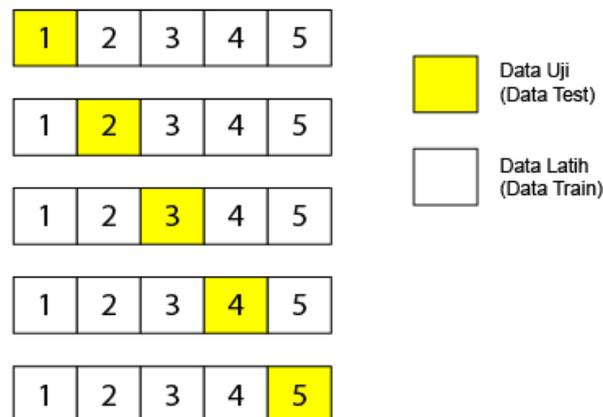
Pada penelitian Analisis Sentimen dan Klasifikasi Kategori terhadap Tokoh Publik pada Twitter [7] dilakukan proses ekstraksi fitur menggunakan TF-IDF dan pengklasifikasian menggunakan metode Naive Bayes Classifier. Penelitian dilakukan menggunakan Naive Bayes dengan fitur *Term Frequency* dan TF-IDF. Hasil akurasi dengan fitur *Term Frequency* adalah 79,91% sedangkan hasil akurasi dengan fitur TF-IDF adalah 79,68% [7].

Sedangkan pada penelitian Perancangan Sistem Analisis Sentimen Komentar Pelanggan Menggunakan Metode Naive Bayes Classifier, data yang digunakan adalah data yang diambil dari komentar-komentar *feedback* untuk hotel-hotel pada aplikasi *booking* Agoda. Dari data latih diperoleh 6 kategori, yaitu kamar, makanan, pelayanan, fasilitas, harga, dan umum. Hasil dari akurasi kategori memiliki nilai 77,14%, presisi sentimen sebesar 99,12%, recall sentimen sebesar 72,9%, dan akurasi sentimen sebesar 75,42% [8].

2.2. Grid Search Cross Validation

Grid search merupakan sebuah metode untuk mencari parameter terbaik pada sebuah model *machine learning* sehingga memiliki hasil yang baik. Grid search mencari kombinasi parameter terbaik berdasarkan kemungkinan-kemungkinan parameter pada sebuah model *machine learning* [9].

Cross validation adalah metode yang umum digunakan untuk mengevaluasi kinerja sebuah model. Data dibagi menjadi dua bagian, yaitu data untuk pelatihan model dan data untuk evaluasi [10]. Pada K-Fold Cross Validation, data dibagi rata sebanyak k. Dataset kemudian dibagi menjadi data *train* dan data *test*. Digunakan 1 fold sebagai data *test* sedangkan bagian lainnya digunakan sebagai data *train*. Hal ini terus berulang sebanyak k kali. Ilustrasi K-Fold Cross Validation dapat dilihat pada Gambar 1 dengan contoh k=5.



Gambar 1. Ilustrasi 5-Fold Cross Validation

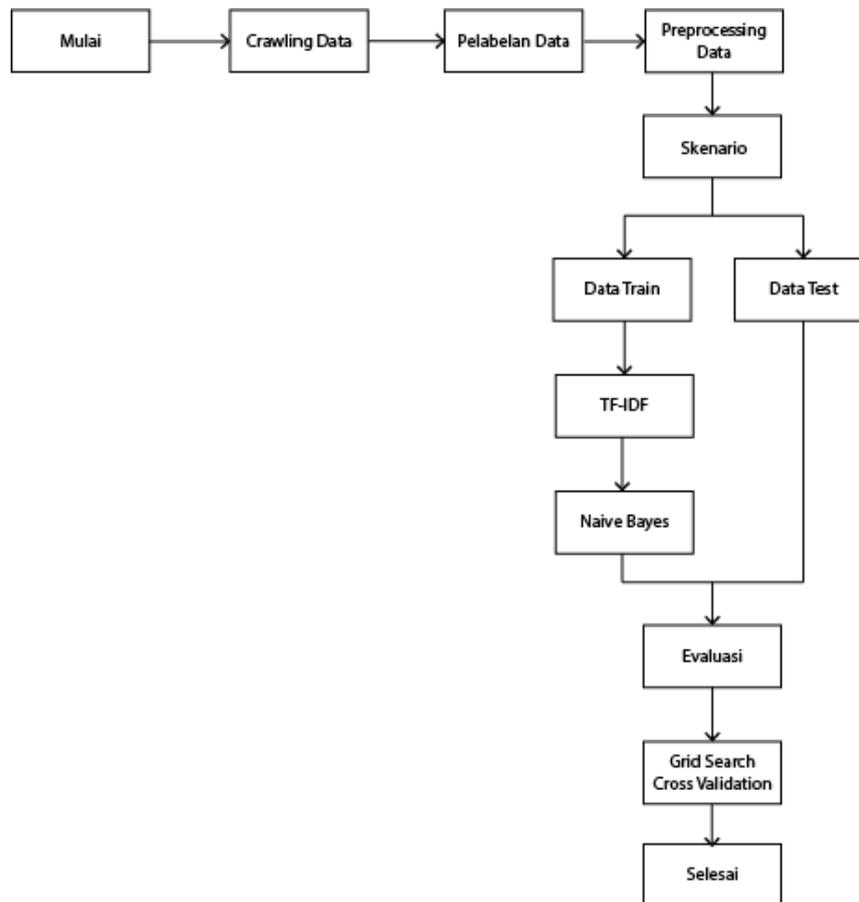
Grid search dan cross validation dapat digunakan secara bersamaan untuk mencari parameter terbaik dalam sebuah model *machine learning* dengan hasil yang lebih stabil dan akurat.

3. Sistem yang Dibangun

3.1. Gambaran Umum Perancangan Sistem

Pembangunan sistem dimulai dengan mengambil data (*crawling* data) pada Twitter dengan kata kunci “sate dj” dan “cuanki serayu”. Setelah itu, dilakukan pelabelan secara manual pada *tweet* dengan 3 kategori; positif, negatif, dan netral. Setelah data selesai dilabeli, dilakukan *preprocessing* data. Data yang telah dilakukan *preprocessing* kemudian dilakukan ekstraksi fitur TF-IDF. Data dibagi menjadi data *test* dan data

train dan diklasifikasi menggunakan *Naïve Bayes Classifier*. Gambaran alur pembangunan sistem dapat dilihat pada Gambar 2.



Gambar 2. Gambar Alur Perancangan Sistem

3.2. Crawling Data

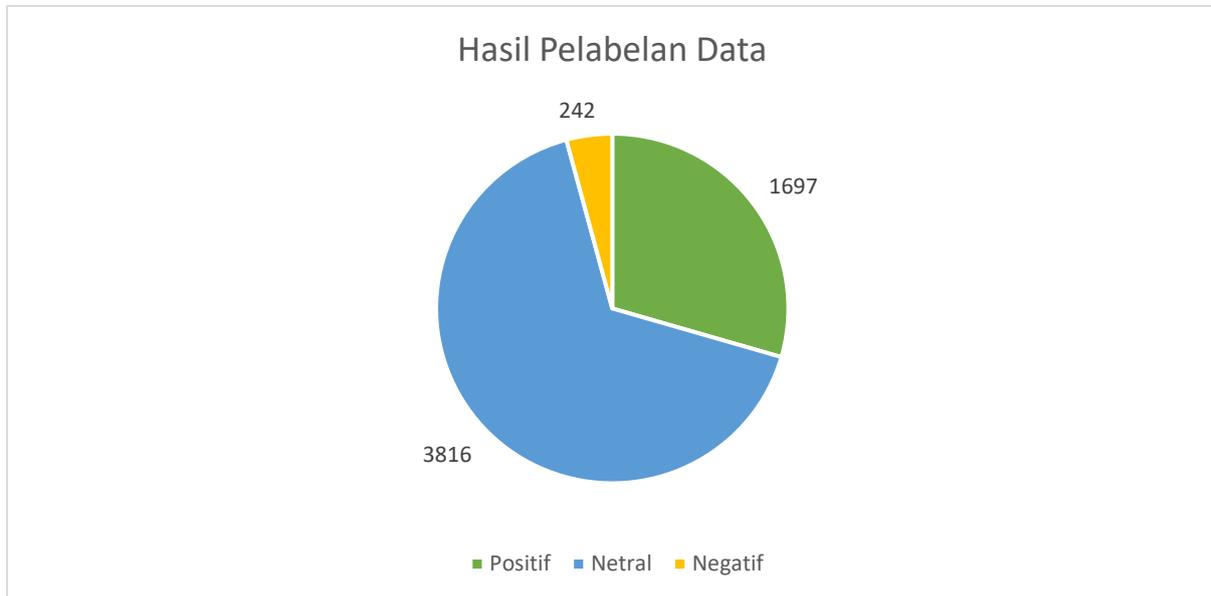
Crawling data dilakukan menggunakan library Tweepy pada bahasa pemrograman Python. Data diambil dari tanggal 1 Januari 2019 hingga 28 Februari 2021. Kata kunci yang digunakan adalah “sate dj” dan “cuanki serayu”. Diperoleh 5754 data dari proses crawling data tersebut. Data yang diambil memiliki atribut ID *tweet*, *username*, tanggal dibuatnya *tweet*, dan isi *tweet*. Data kemudian disimpan dalam file dengan format .csv. Contoh data hasil *crawling* dapat dilihat pada Tabel 1.

Tabel 1. Contoh Data Hasil Crawling

ID tweet	Tanggal dibuat	username	tweet
1310788821252857857	2020-09-29 03:49:42	ridhandani	@briankhrisna dabest emang cuanki serayu
1309683885823766528	2020-09-26 02:39:05	yurichasanah	@rizkidwika bandung pagi-pagi, cobain yg namanya sate jando! Siang2 udah paling ok makan cuanki serayu. Dah sore ni... https://t.co/YfJTrQACbH
1309540990064054272	2020-09-25 17:11:16	ronfirmansyah	Cuanki Serayu selalu jd destinasi kuliner wajib gue kalo ke Bdg. Ga pernah bosan aja, gatau kenapa. https://t.co/CMI3GmE255
1309534691972784129	2020-09-25 16:46:15	moelyantii	@bernadetteri makan sate dj juga dong
1309520978045595648	2020-09-25 15:51:45	monadeby	@aipsun Karesep aku eta cuanki serayu 🤩

3.3. Pelabelan Data

Pelabelan data dilakukan oleh 4 orang secara manual menjadi label positif, negatif, atau netral. Dari hasil pelabelan diperoleh data seperti pada diagram seperti pada Gambar 3.



Gambar 3. Diagram Hasil Pelabelan Data

Sementara untuk contoh data dan labelnya dapat dilihat pada Tabel 2.

Tabel 2. Contoh Pelabelan Data

Tweet	Label
Huuu sate dj ternyata enak parahhhh kamana wae aku teh. Makasih yah @yuliatrifadilah	Positif
@askmenfess cuanki serayu is a must.	Positif
@aipsun Lu belum pernah ke Cuanki Serayu?	Netral
Cuanki Serayu buka ga?	Netral
@salsefira cuanki serayu b aja, porsinya dikit bgt tp harganya mahal sist	Negatif

3.4. Preprocessing Data

Preprocessing data dilakukan untuk membersihkan data hasil crawling. Berikut adalah tahapan yang dilakukan pada preprocessing data

1. Case folding, URL & Punctuation Removal, dan Tokenizing
Case folding adalah proses mengubah seluruh huruf pada *tweet* menjadi huruf kecil. *URL & punctuation removal* dilakukan untuk menghapus URL, *username* Twitter, simbol, tanda baca, angka, dan huruf yang berlebih dalam sebuah kata. Terakhir, *tokenizing* dilakukan untuk memecah kalimat menjadi per kata.
2. Normalization
Normalization atau normalisasi adalah mengubah kata-kata yang ditulis menggunakan bahasa sehari-hari dan bahasa gaul menjadi bahasa formal.
3. Foreign Words Translation
Foreign words translation adalah proses menerjemahkan bahasa asing menjadi Bahasa Indonesia yang baku.

4. Stopwords Removal
Stopwords removal merupakan tahapan menghapus *stopwords* atau kata umum yang dianggap tidak memiliki makna.
5. Stemming
Stemming merupakan proses mengubah kata menjadi kata dasarnya. Stemming dilakukan menggunakan *library* Sastrawi.

Contoh preprocessing data dapat dilihat pada Tabel 3.

Tabel 3. Contoh Preprocessing Data

Proses	Hasil Proses
Data	Huuu sate dj ternyata enak parahhhh kamana wae aku teh. Makasih yah @yuliatrifadilah
Case Folding, URL & Punctuation Removal, Tokenizing	hu sate dj ternyata enak parah kamana wae aku teh makasih yah
Normalization	hu sate dj ternyata enak parah kamana wae aku teh terima kasih ya
Foreign Words Translation	hu sate dj ternyata enak parah ke mana saja aku teh terima kasih ya
Stopwords Removal	sate dj ternyata enak parah mana aku teh terima kasih
Stemming	sate dj nyata enak parah mana aku teh terima kasih

3.5. Skenario

Skenario *oversampling* dilakukan untuk membuat performa lebih baik pada dataset tidak seimbang menggunakan RandomOverSampler yang terdapat dalam *package* imbalanced learn. *Oversampling* adalah sebuah metode untuk menyeimbangkan dataset dengan cara membuat data buatan dari data yang minoritas sehingga jumlah data yang minoritas tersebut menjadi lebih seimbang dengan data yang lainnya [11].

Oversampling data dilakukan pada label negatif yang sebelumnya berjumlah 242 data. Terdapat enam skenario pemodelan; skenario 1 menggunakan data yang tidak di-*oversampling*, skenario 2 menggunakan data setelah *oversampling* data berlabel negatif menjadi 500 data, skenario 3 menggunakan data setelah *oversampling* data berlabel negatif menjadi 1000 data, skenario 4 menggunakan data setelah *oversampling* data berlabel negatif menjadi 1500 data, skenario 5 menggunakan data setelah *oversampling* data berlabel negatif menjadi 2000 data, dan skenario 6 menggunakan data setelah *oversampling* data berlabel negatif menjadi 2500 data. Dataset tiap skenario dibagi menjadi data *train* dan data *test* dengan perbandingan 70:30. Dataset tersebut digunakan pada tahap-tahap selanjutnya dan dibandingkan hasilnya satu sama lain.

3.6. TF-IDF

TF-IDF atau *Term Frequency – Inverse Document Frequency* merupakan sebuah perhitungan yang biasa digunakan untuk menghitung bobot kata pada sebuah dokumen. Lebih besar frekuensi kemunculan sebuah kata pada suatu dokumen, kata tersebut lebih memungkinkan untuk menjadi kata yang mewakili dokumen tersebut [12]. Kata yang umum pada satu atau beberapa bagian dokumen cenderung memiliki nilai TF-IDF yang besar daripada kata umum seperti preposisi [13]. Hal ini disebabkan karena kata-kata tersebut termasuk dalam *stopwords* yang kemudian akan dihapus untuk meningkatkan kinerja program.

Perhitungan TF-IDF dapat menggunakan rumus berikut.

$$w_d = f_{w,d} * \log\left(\frac{|D|}{f_{w,D}}\right) \quad (1)$$

Dengan w adalah sebuah kata, D adalah seluruh dokumen, d adalah sebuah dokumen pada D , $f_{w,d}$ adalah frekuensi kemunculan kata w pada dokumen d , dan $f_{w,D}$ adalah frekuensi kemunculan kata w pada seluruh dokumen.

Ekstraksi fitur menggunakan TF-IDF dilakukan dengan parameter $min_df=0,01$, $max_df=1,0$, dan $max_features=None$. Parameter min_df digunakan untuk menghilangkan kata yang frekuensi kemunculannya kurang dari jumlah tertentu, max_df digunakan untuk menghilangkan kata yang frekuensi kemunculannya lebih dari jumlah tertentu, dan $max_features$ digunakan untuk membatasi jumlah kata yang diolah pada TF-IDF. None berarti semua nilai TF-IDF yang diperoleh tidak akan diabaikan. Data sebelumnya

dibagi menjadi data *train* dan data *test* terlebih dahulu dengan rasio 70:30. Ilustrasi pembobotan menggunakan TF-IDF dapat dilihat pada Tabel 4.

Tabel 4. Ilustrasi TF-IDF

kata	TF					IDF = log(D/df)	TF * IDF				
	D1	D2	D3	D4	D5		D1	D2	D3	D4	D5
enak	1	0	0	1	0	log(5/2) = 0,397	0,397	0	0	0,397	0
murah	0	0	0	1	2	log(5/3) = 0,222	0	0	0	0,222	0,444
suka	0	1	1	1	0	log(5/3) = 0,222	0	0,222	0,222	0,222	0

3.7. Naïve Bayes Classifier

Naïve Bayes Classifier merupakan metode klasifikasi yang menggunakan probabilitas. Naïve Bayes Classifier dalam analisis sentimen menghitung probabilitas kelas berdasarkan distribusi kata pada sebuah dokumen. Naïve Bayes Classifier menggunakan Teorema Bayes untuk memprediksi kelas sebuah *feature* [6].

Jika pada Multinomial Naïve Bayes perhitungan dilakukan dengan menghitung probabilitas kemunculan sebuah kata pada suatu kelas, pada Complement Naïve Bayes dilakukan dengan menghitung probabilitas kemunculan sebuah kata selain pada kelasnya [14].

Pada penelitian ini dilakukan pengklasifikasian menggunakan metode Complement Naïve Bayes pada setiap data *train* masing-masing skenario. Perhitungan Complement Naïve Bayes berdasarkan pembobotan TF-IDF menggunakan rumus berikut [14]

$$\hat{\theta}_{ci} = \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}} \tag{2}$$

$$w_{ci} = \log \hat{\theta}_{ci} \tag{3}$$

$$l(t) = \arg \min_c \sum_i t_i w_{ci} \tag{4}$$

Perhitungan dilakukan berdasarkan seluruh dokumen *j* bukan di kelas *c*. d_{ij} adalah nilai TF-IDF sebuah kata *i* pada dokumen *j* dan *y* adalah label atau kelas. α_i adalah parameter *smoothing* dan $\alpha = \sum_i \alpha_i$ atau jumlah seluruh α_i . $t = (t_1, \dots, t_n)$ adalah dokumen pada data *test* dan t_i adalah jumlah kata *i*. Pada Complement Naïve Bayes, pelabelan dilakukan berdasarkan nilai yang paling rendah. Ilustrasi perhitungan Complement Naïve Bayes berdasarkan nilai TF-IDF pada tabel Tabel 5 sebagai berikut

Tabel 5. Contoh Nilai TF-IDF pada Perhitungan Complement Naïve Bayes

Dokumen	Kata					Kelas
	enak	makan	suka	aku	mahal	
D1	0,7	0	0,6	0,7	0	Positif
D2	0	0,6	0	0,5	0	Netral
D3	0	0,6	0	0,6	0,7	Negatif

Misalkan mencari kelas sebuah kalimat dengan kata yang dimiliki adalah ‘aku’, ‘makan’, dan ‘suka’ perhitungannya seperti berikut

$$P(\text{positif} | \text{aku}, \text{makan}, \text{suka}) = \log\left(\frac{0,6+1}{0,6+0,7+0+3}\right) + \log\left(\frac{0+1}{0,6+0,7+0+3}\right) + \log\left(\frac{0,7+1}{0,6+0,7+0+3}\right) = -1,465$$

$$P(\text{netral} | \text{aku}, \text{makan}, \text{suka}) = \log\left(\frac{0,5+1}{0,5+0+0,6+3}\right) + \log\left(\frac{0,6+1}{0,5+0+0,6+3}\right) + \log\left(\frac{0+1}{0,5+0+0,6+3}\right) = -0,293$$

$$P(\text{negatif} | \text{aku}, \text{makan}, \text{suka}) = \log\left(\frac{0,6+1}{0,6+0,6+0+3}\right) + \log\left(\frac{0,6+1}{0,6+0,6+0+3}\right) + \log\left(\frac{0+1}{0,6+0,6+0+3}\right) = -1,461$$

Karena nilai $P(\text{positif} | \text{aku}, \text{makan}, \text{suka})$ memiliki nilai terkecil sebesar -1,465 maka kalimat tersebut masuk ke dalam kelas positif.

4. Evaluasi

Evaluasi dilakukan dengan *classification report*. *Classification report* terdiri dari *precision*, *recall*, *f1-score*, dan *accuracy*. *Precision*, *recall*, *f1-score*, dan *accuracy* dihitung dengan rumus sebagai berikut

$$precision = \frac{TP}{TP + FP} \quad (5)$$

$$recall = \frac{TP}{TP + FN} \quad (6)$$

$$f1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (7)$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

Dengan TP (True Positive) adalah data positif yang diprediksi positif, TN (True Negative) adalah data negatif yang diprediksi negatif, FP (False Positive) adalah data negatif yang diprediksi positif, dan FN (False Negative) adalah data positif yang diprediksi negatif. Nilai yang digunakan dalam mengevaluasi adalah *f1-score macro average* karena *f1-score macro average* menghitung nilai masing-masing kelas sehingga cocok digunakan pada data yang tidak seimbang. Perbandingan skor tiap skenario dapat dilihat pada Tabel 6.

Tabel 6. Perbandingan Skor Classification Report

Skenario	Precision	Recall	F1-score	Accuracy
No oversampling	0,60	0,70	0,63	0,76
Oversampling data negatif 500	0,68	0,77	0,71	0,77
Oversampling data negatif 1000	0,72	0,77	0,74	0,77
Oversampling data negatif 1500	0,77	0,79	0,78	0,79
Oversampling data negatif 2000	0,79	0,79	0,78	0,80
Oversampling data negatif 2500	0,81	0,80	0,80	0,83

Dari Tabel 6 dapat dilihat bahwa skenario 6 memiliki nilai *f1-score* terbesar sebesar 0,80. Jika dibandingkan dengan data asli tanpa *oversampling* (skenario 1), hasil ini menunjukkan bahwa sistem yang dibangun sudah cukup bagus untuk memprediksi data. Nilai *f1-score* sebesar 0,80 dapat terjadi karena data berlabel negatif sudah cukup seimbang dengan data berlabel positif dan netral. Selain itu, *oversampling* berpengaruh dalam performa sistem dengan data yang tidak seimbang. Semakin seimbang data maka sistem dapat memprediksi dengan lebih baik.

Setelah evaluasi dilakukan *hyperparameter tuning* untuk mencari tahu parameter-parameter terbaik menggunakan Grid Search Cross Validation dengan 10-fold cross validation. Parameter yang dicari adalah *min_df*, *max_df*, *max_features* pada *TfidfVectorizer* dan *alpha* pada *ComplementNB*. Hasil Grid Search Cross Validation dapat dilihat pada Tabel 7.

Tabel 7. Hasil Grid Search Cross Validation

Skenario	Parameter				F1-score
	min_df	max_df	max_features	alpha	
No oversampling	0,001	0,7	2000	0,7	0,665
Oversampling data negatif 500	0,0001	0,7	3000	1,0	0,743
Oversampling data negatif 1000	0,0001	0,7	3000	1,0	0,775
Oversampling data negatif 1500	0,0001	0,7	3000	1,0	0,788
Oversampling data negatif 2000	0,0001	0,7	3000	1,0	0,798
Oversampling data negatif 2500	0,0001	0,7	2000	0,7	0,794

Dari hasil Grid Search Cross Validation dapat dilihat bahwa skenario 5 memiliki nilai *f1-score* terbesar sebesar 0,798 dengan parameter *min_df*=0,0001, *max_df*=0,7, *max_features*=3000, dan *alpha*=1,0.

5. Kesimpulan dan Saran

Dalam penelitian ini data sebanyak 5754 dilabeli secara manual menjadi 242 data negatif, 3816 data netral, dan 1697 data positif. Data tersebut dibuat menjadi enam skenario. Dari hasil skenario-skenario tersebut skenario 6 memiliki nilai *f1-score* tertinggi, yaitu 0,80. Hasil Grid Search Cross Evaluation

menunjukkan bahwa skenario 5 memiliki nilai *f1-score* tertinggi sebesar 0,798 dengan parameter $\text{min_df}=0,0001$, $\text{max_df}=0,7$, $\text{max_features}=3000$, dan $\text{alpha}=1,0$.

Berdasarkan hasil pengujian dan analisis yang telah dilakukan dapat disimpulkan bahwa klasifikasi menggunakan metode Complement Naïve Bayes dan TF-IDF memiliki performa yang cukup baik dalam analisis sentimen. Data yang tidak seimbang berpengaruh dalam performa sistem, sehingga melakukan *oversampling* pada data yang tidak seimbang dapat membuat performa sistem lebih baik. Semakin seimbang dataset yang digunakan maka performa sistem klasifikasi semakin baik.

Saran untuk penelitian selanjutnya adalah melakukan analisis sentimen dengan data yang seimbang sehingga memiliki performa yang lebih baik. Selain itu, dapat dilakukan *preprocessing* data yang lebih baik terutama pada proses normalisasi dan *stopwords removal*.

Daftar Pustaka

- [1] Twitter. 2009. [Online]. Available at: <https://www.britannica.com/topic/Twitter>. [Accessed 22 October 2020].
- [2] Fachrizal, R. 2020. Inilah Media Sosial yang Paling Banyak Diakses Pengguna 3 Indonesia. [Online] Available at: <https://infokomputer.grid.id/read/122220303/inilah-media-sosial-yang-paling-banyak-diakses-pengguna-3-indonesia?page=all>. [Accessed 17 January 2021].
- [3] Liu, B. 2010. Handbook of natural language processing. Florida: CRC Press.
- [4] Buntoro, G. A. 2017. Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. INTEGER: Journal of Information Technology, 2:1.
- [5] Anagaw, A., & Chang, Y. L. 2019. A new complement naïve Bayesian approach for biomedical data classification. Journal of Ambient Intelligence and Humanized Computing. 10:10 3889-3897.
- [6] Medhat, W., Hassan, A., & Korashy, H. 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal. 5:4 1093-1113.
- [7] Hidayatullah, A. F., & Azhari, A. S. 2015. Analisis sentimen dan klasifikasi kategori terhadap tokoh publik pada twitter. In Seminar Nasional Informatika (SEMNASIF). 1:1.
- [8] Sipayung, E. M., Maharani, H., & Zefanya, I. 2016. Perancangan Sistem Analisis Sentimen Komentar Pelanggan Menggunakan Metode Naive Bayes Classifier. Jurnal Sistem Informasi (JSI). 8:1 958-965.
- [9] Abas, M. A. H., Ismail, N., Ali, N. A., Tajuddin, S., & Tahir, N. M. 2020. Agarwood oil quality classification using support vector classifier and grid search cross validation hyperparameter tuning. Int. J. 8.
- [10] Yadav, S., & Shukla, S. 2016. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. 2016 IEEE 6th International conference on advanced computing (IACC). 78-83.
- [11] Triyanto, A. Y., & Kusumaningrum, R. 2017. Implementasi Teknik Sampling untuk Mengatasi Imbalanced Data pada Penentuan Status Gizi Balita dengan Menggunakan Learning Vector Quantization. Jurnal Ilmu Pengetahuan dan Teknologi Komunikasi. 39-50.
- [12] Yun-tao, Z., Ling, G., & Yong-cheng, W. 2005. An improved TF-IDF approach for text classification. Journal of Zhejiang University-Science A. 6:1 49-55.
- [13] Ramos, J. 2003. Using tf-idf to determine word relevance in document queries. Proceedings of the first instructional conference on machine learning. 242:1 29-48.
- [14] Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. 2003. Tackling the poor assumptions of naive bayes text classifiers. Proceedings of the 20th international conference on machine learning (ICML-03). 616-623.