

Uji Klasifikasi Algoritma *Naive Bayes Classification* dalam Analisis Sentimen Ulasan Puncak B29 Lumajang

Fadhel Akhmad Hizham^{1*}, Cahyasari Kartika Murni², Marita Qori'atunnadyah³
Informatika, Institut Teknologi dan Bisnis Widya Gama Lumajang, Lumajang, Indonesia
*e-mail *Corresponding Author*: hizhamfadhel@gmail.com

Abstract

B29 Peak is a tourist attraction which is located in Argosari Village, Senduro District, Lumajang Regency, which is the highest peak in the Bromo sand sea area with a height of 2,900 meters above sea level. Based on Google Maps, B29 Peak has the best review score for tourist attraction in Lumajang Regency, with an average score of 4.7 stars and reviewed by 3,368 users until October 25th, 2023. Sentiment analysis was carried out to group the very large number of user reviews as an evaluation in order to correct deficiencies and improve the quality of the tourist attraction. Grouping sentiment analysis is using the Naive Bayes Classification. The result of classification obtained the average score of accuracy is 91.06%, precision 96.57%, recall 94.07% and F1-score 95.31%. While the best result is on the 150 data testing, with the accuracy of 92.67%, precision 97.20%, recall 95.21% and F1-score 96.19%.

Keyword: *Sentimen Analisis; B29 Peak; Naive Bayes Classification*

Abstrak

Puncak B29 merupakan salah satu objek wisata yang terletak di Desa Argosari, Kecamatan Senduro, Kabupaten Lumajang, yang merupakan puncak tertinggi di Kawasan lautan pasir Bromo dengan ketinggian 2.900 meter di atas permukaan laut. Berdasarkan situs *Google Maps*, Puncak B29 memiliki skor ulasan yang terbaik untuk tempat wisata Kabupaten Lumajang, dengan skor ulasan rata-rata 4,7 bintang dan diulas oleh 3.368 pengguna hingga 25 Oktober 2023. Sentimen analisis dilakukan untuk mengelompokkan ulasan pengguna yang jumlahnya sangat banyak sebagai evaluasi agar dapat memperbaiki kekurangan dan meningkatkan kualitas dari tempat wisata tersebut. Pengelompokkan sentimen analisis menggunakan algoritma *Naive Bayes Classification*. Hasil klasifikasi diperoleh nilai rata-rata akurasi sebesar 91,06%, presisi 96,57%, *recall* 94,07% dan *F1-score* 95,31%. Sementara untuk perolehan hasil terbaik terdapat pada pengujian dataset sebanyak 150 data, dengan akurasi sebesar 92,67%, presisi 97,20%, *recall* 95,21% dan *F1-score* 96,19%.

Kata kunci: *Sentimen Analisis; Puncak B29; Naive Bayes Classification*

1. Pendahuluan

Perkembangan teknologi berbasis internet berdampak pada makin luasnya inovasi di berbagai sektor, baik setor industri, sektor ekonomi, maupun sektor pariwisata. Inovasi tersebut ditunjukkan dengan terciptanya berbagai produk yang telah dipasarkan di berbagai kalangan, menjamurnya *content creator* di berbagai sosial media, hingga tempat pariwisata yang disediakan untuk pengunjung. Selain itu, melalui pemanfaatan teknologi, manusia dapat dengan mudah mencari maupun mendapatkan informasi dari sekian banyak inovasi tersebut melalui mesin pencarian internet [1].

Dari sektor pariwisata, perkembangan teknologi berbasis internet dapat dimanfaatkan untuk menggali informasi tentang tempat-tempat pariwisata dan mengolah data yang sudah ada untuk mengukur kualitas melalui perbandingan opini pengunjung terhadap suatu tempat pariwisata [2]. Hasil pengukuran kualitas tersebut dapat digunakan sebagai pertimbangan bagi pengguna untuk menentukan keputusan dalam memilih tempat wisata [3], dan dapat bermanfaat bagi pengelola wisata sebagai bahan evaluasi untuk meningkatkan kualitas dari tempat wisata tersebut agar dapat menarik lebih banyak pengunjung [4].

Evaluasi dapat dilakukan dengan memberikan ulasan maupun komentar dari pengguna maupun pengunjung tempat wisata. Ulasan tersebut dapat diberikan melalui aplikasi *Google Maps* API. Akan tetapi, ulasan yang jumlahnya banyak tentunya akan kesulitan dan memerlukan waktu yang lama apabila harus menganalisis masing-masing ulasan tersebut secara manual [5]. Oleh karena itu, untuk menganalisis ulasan yang berjumlah banyak maka digunakan teknik sentimen analisis. Sentimen analisis merupakan metode yang menerapkan konsep *text mining* untuk memberikan klasifikasi yang mempunyai polaritas yang bersifat positif, negatif, maupun netral dari setiap kalimat atau dokumen [6]. Polaritas tersebut dibuat sebagai acuan untuk pengambilan keputusan dan sebagai evaluasi untuk pengembangan suatu tempat wisata ke depannya [7].

Pada penelitian ini, objek yang diambil yaitu tempat wisata Puncak B29 Lumajang. Puncak B29 merupakan salah satu objek wisata yang terletak di Desa Argosari, Kecamatan Senduro, Kabupaten Lumajang, yang merupakan puncak tertinggi di Kawasan lautan pasir Bromo dengan ketinggian 2.900 meter di atas permukaan laut. Objek tersebut diambil sebagai penelitian karena berdasarkan situs *Google Maps*, tempat wisata tersebut memiliki skor ulasan yang terbaik untuk tempat wisata Kabupaten Lumajang, dengan skor ulasan rata-rata 4,7 bintang dan diulas oleh 3.368 pengguna hingga 25 Oktober 2023.

Penelitian yang menerapkan sentimen analisis untuk mengklasifikasikan ulasan atau komentar telah dilakukan oleh beberapa peneliti, dengan berbagai metode klasifikasi. Metode tersebut diantaranya metode *Backpropagation Neural Network* [2], metode *Naive Bayes Classification* [6], metode *K-Nearest Neighbor* [6], dan metode *Decision Tree* [8].

Dari uraian latar belakang tersebut, peneliti menganalisis sentimen dari ulasan tempat wisata puncak B29 Lumajang dengan menggunakan metode klasifikasi *Naive Bayes Classification* (NBC). Perumusan masalah yang dilakukan pada penelitian ini yaitu bagaimana peran sentimen analisis dalam menganalisis ulasan tempat wisata Puncak B29 Lumajang berdasarkan komentar pengguna di *Google Maps*. Tujuan dari penelitian ini yaitu untuk menguji klasifikasi algoritma NBC dalam menganalisis sentimen ulasan dari Puncak B29 Lumajang.

2. Tinjauan Pustaka

Penelitian pertama mengenai sentimen analisis menggunakan *Naive Bayes Classification* yaitu "Analisis Sentimen Penilaian Tempat Tujuan Wisata Kota Tegal Berbasis *Text Mining*" oleh Somantri dan Dairoh. Penelitian ini mengambil ulasan tempat wisata Kota Tegal dari situs *Google Maps* dari tahun 2017 sampai 2018 dengan total sebanyak 120 data *file text* dengan jumlah kata yang berbeda di setiap *filenya*. Penelitian ini membandingkan dua metode yaitu *Naive Bayes Classification* dan *Decision Tree*. Hasil penelitian ini menunjukkan bahwa persentase akurasi untuk metode *Naive Bayes Classification* lebih tinggi dibanding *Decision Tree* yaitu sebesar 73,33% [3].

Penelitian kedua yaitu "Sentiment Analysis Objek Wisata Kalimantan Barat Pada *Google Maps* Menggunakan Metode *Naive Bayes*" yang dilakukan oleh Ahmad Rifa'i, Herry Sujaini, Dian Prawira. Penelitian ini mengambil 10 tempat wisata di Kalimantan Barat dengan masing-masing 50 ulasan per tempat wisata. Hasil penelitian ini menunjukkan bahwa akurasi tertinggi yaitu sebesar 76,00% pada tempat wisata Air Terjun Pancur Aji [4].

Penelitian ketiga yaitu "Perbandingan Metode *Naive Bayes*, KNN Dan *Decision Tree* Terhadap Analisis Sentimen Transportasi KRL *Commuter Line*" yang dilakukan oleh Nova Tri Romadloni, Imam Santoso, dan Sularso Budilaksono. Penelitian ini mengambil ulasan pengguna transportasi KRL *Commuter Line*. Pada penelitian tersebut, hasilnya diperoleh bahwa penggunaan metode *Naive Bayes* menghasilkan akurasi sebesar 80,00%, precision 66,67%, sensitifity 100,00%, dan specificity 66,67% [6].

Penelitian keempat yaitu "Analisis Sentimen Ulasan Villa di Ubud Menggunakan Metode *Naive Bayes*, *Decision Tree*, dan K-NN" yang dilakukan oleh Ni Luh Wiwik Sri Rahayu Ginantra, C. P. Yanti, G. D. Prasetya, Ida Bagus Gede Sarasvananda, dan I Komang Arya Ganda Wiguna. Penelitian dilakukan dengan mengambil data pada *Google Maps* API sebanyak 2894 ulasan yang dibagi menjadi 2024 data *training* dan 867 data uji. Hasil penelitian diperoleh untuk metode *Naive Bayes* memperoleh akurasi sebesar 86,66%, *Decision Tree* sebesar 87,78%, dan k-NN sebesar 91,26% [9].

Penelitian kelima yaitu "Sentimen Analisis pada Data Ulasan Aplikasi KAI Access di *Google Play Store* menggunakan Metode *Multinomial Naive Bayes*. Jurnal Smart Teknologi" yang dilakukan oleh Machrus Izunnahdi, Ginanjar Aburrahman, dan Ari Eko Wardoyo. Penelitian ini

menggunakan data sebanyak 7500 ulasan yang terbagi menjadi sentimen positif dan sentimen negatif yang sebelumnya telah divalidasi oleh tenaga ahli pihak PT KAI daerah Banyuwangi. Penelitian ini menggunakan skenario dan partisi data *K-fold Cross Validation* pada pemodelan dengan nilai *k-fold* 2, 3, 4, 5, 6, 8 dan 10. Hasil pengujian pada penelitian memperoleh tingkat akurasi sebesar 89%, tingkat presisi sebesar 92% dan tingkat *recall* sebesar 85% [10].

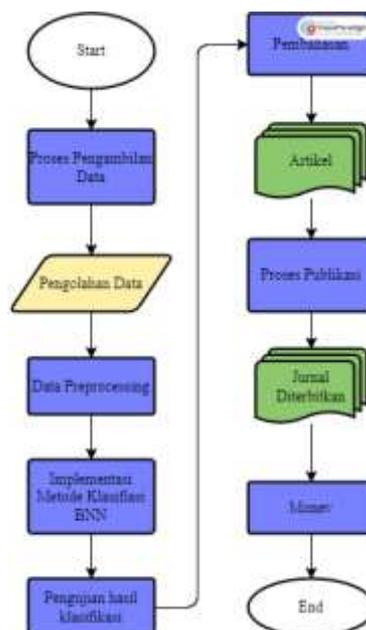
Penelitian keenam yaitu “Penerapan *Naïve Bayes Classifier*, *K-Nearest Neighbor* (KNN) dan *Decision Tree* untuk Menganalisis Sentimen pada Interaksi Netizen dan Pemerintah” yang dilakukan oleh M. Khairul Anam, Bunga Nanti Pikir, Muhammad Bambang Firdaus, Susi Erlinda, dan Agustin. Data yang digunakan adalah tweet dengan jumlah dataset sebanyak 150 tweets. Data tersebut kemudian dianalisis agar menjadi informasi. Analisa dilakukan menggunakan metode data mining yaitu *Naive Bayes Classification*, *k-NN*, dan *Decision Tree*. Hasil akurasi yang didapatkan cukup beragam yaitu dari metode *Naive Bayes* sebesar 100%, metode KNN sebesar 98,25%, dan metode *decision tree* sebesar 62,28% [11].

Penelitian ketujuh yaitu “Text Mining Sentimen Analisis Pengguna Aplikasi *Marketplace* Tokopedia Berdasar Rating dan Komentar Pada *Google Play Store*” yang dilakukan oleh Saiful Ulya, Achmad Ridwan, Widya Cholid, dan Fida Maisa Hana. Penelitian ini mengambil 1000 *record* ulasan pengguna Tokopedia, dan setelah melakukan *cleaning data* menjadi 996 *record*. Hasil klasifikasi diperoleh untuk akurasi *Naive Bayes* sebesar 89,13%, *Decision Tree* sebesar 83,51% dan *Deep Learning* sebesar 84,91% [8].

Penelitian kedelapan yang berkaitan dengan sentimen analisis untuk sektor wisata yaitu “*Sentiment Analysis Using Backpropagation Method to Recognize the Public Opinion*” yang dilakukan oleh I Komang Arya Ganda Wiguna, Putu Sugiartawan, I Gede Iwan Sudipa, dan I Putu Yudi Pratama. Penelitian ini mengambil data berupa ulasan hotel yang diperoleh dari situs www.kaggle.com dengan jumlah data sebanyak 515.739 *record*. Penelitian ini menggunakan metode klasifikasi *Backpropagation Neural Network*. Hasil penelitian diperoleh bahwa akurasi terbaik sebesar 71,630% dengan satu buah lapisan tersembunyi yang mengandung 10 neuron pada lapisan tersebut dan *learning rate* 0,000002 [12].

Berdasarkan beberapa penelitian tersebut, dapat disimpulkan bahwa penggunaan *Naive Bayes Classification* dapat dilakukan untuk klasifikasi sentimen analisis dengan tingkat akurasi yang cukup akurat. Pada penelitian ini, yang membedakan dengan penelitian sebelumnya yaitu terdapat pada jumlah data beserta objek penelitiannya, yaitu penelitian ini berobjek pada tempat wisata Puncak B29, yang diperoleh ulasannya pada situs *Google Maps*.

3. Metodologi



Gambar 1. Diagram Alir Penelitian

Penelitian ini dimulai dengan pengumpulan data, preprocessing data, pembobotan kata dengan TF-IDF, klasifikasi dengan menggunakan *Naive Bayes Classifier*, dan pengujian klasifikasi. Proses pada penelitian ini bersifat eksperimental untuk mengevaluasi algoritma *Naive Bayes Classification* dalam melakukan analisis sentimen. Diagram perancangan sistem ditunjukkan pada Gambar 1.

3.1. Pengumpulan Data

Penelitian ini menggunakan data teks hasil dari ulasan pengguna pada Google Maps. Ulasan yang digunakan untuk dataset yaitu ulasan pada tempat wisata Puncak B29. Pengambilan dataset di Google Maps disebut dengan *scrapping* atau *crawling*. *Crawling* bertujuan untuk mengumpulkan data berupa teks dari ulasan yang ditulis oleh wisatawan pada Google Maps. Pengambilan data dengan mengambil kategori ulasan yang paling relevan, terbaik, dan terburuk.

Pada penelitian ini, pengambilan data dilakukan dengan bantuan situs *Outscraper*. Jumlah ulasan yang digunakan untuk dataset yaitu sebanyak 500 ulasan dengan skor ulasan yang bervariasi dari 1 sampai 5 bintang, dengan rincian 3 ulasan bintang 1, 2 ulasan bintang 2, 9 ulasan bintang 3, 74 ulasan bintang 4, dan 412 ulasan bintang 5.

3.2. Text Preprocessing

Tahapan pertama dalam text mining yaitu *text preprocessing*, yang merupakan tahapan penting dalam analisis sentimen karena tahapan ini sangat menentukan hasil analisis yang dihasilkan. Preprocessing yaitu proses pembersihan, penyaringan, dan pengolahan data ataupun mengubah format data agar data dapat digunakan dalam tahapan selanjutnya dengan optimal. Tahapan preprocessing ditunjukkan pada Tabel 1 [13].

Tabel 1. Tahapan *Text Preprocessing*

Tahapan	Penjelasan
Case Folding	Proses menghilangkan tanda baca, emoji, dan mengubah teks menjadi huruf kecil (penyamaan case dalam sebuah dokumen).
Tokenizing	Proses memecah string input menjadi kata per kata yang menyusunnya dalam sebuah kalimat.
Stopword	Proses menghilangkan kata-kata yang terlalu sering muncul yang tidak memiliki makna serta tidak berpengaruh apapun dalam analisis sentimen [14]. Proses menggunakan <i>library</i> Sastrawi pada <i>Python</i> [15].
Stemming	Proses pengembalian kata-kata yang berimbuhan menjadi kata dasar. Proses menggunakan <i>library</i> Sastrawi pada <i>Python</i> [15].

3.3. Pembobotan Kata

Setelah data dilakukan *preprocessing text*, maka selanjutnya yaitu memberikan bobot kata per kata pada jumlah kemunculan suatu kata dalam dokumen atau *record*. Metode pembobotan kata ini menggunakan metode TF-IDF [9]. *Term Frequency* (TF) digunakan untuk mengukur tingkat keseringan dari suatu kata yang muncul dalam satu dokumen [16]. Sementara *Inverse Document Frequency* (IDF) digunakan untuk mengukur tingkat kepentingan dari kata tersebut. Rumus untuk menghitung TF-IDF ditunjukkan pada Persamaan 1.

$$tf.idf(t, d, D) = (1 + \log(f_{t,d})) \left(\log \frac{N}{df_t} \right) \quad (1)$$

Keterangan:

$f_{t,d}$ = banyaknya kata t pada dokumen d

N = total banyaknya dokumen dalam sekumpulan dokumen

df_t = banyaknya dokumen yang terdapat kata t , apabila kata tersebut bernilai tidak ada pada seluruh dokumen (0) maka nilainya adalah 1

3.4. Klasifikasi *Naive Bayes*

Pada penelitian ini, klasifikasi yang digunakan yaitu algoritma *Naive Bayes Classification* untuk melakukan klasifikasi sentimen berlabel positif dan negatif. Algoritma tersebut menghitung probabilitas dengan menjumlahkan frekuensi serta campuran nilai dari dataset. Dengan kata lain,

kehadiran atau ketiadaan salah satu fitur dari suatu kelas tidak berhubungan dengan fitur lainnya [17]. Rumus untuk pengklasifikasian metode *Naïve Bayes* ditunjukkan pada persamaan 2.

$$P(Y | X) = \frac{P(Y) \prod_{i=1}^q P(X_i|Y)}{P(X)} \tag{2}$$

Sementara *Naïve Bayes* dengan fitur kontinu (*Gaussian Naïve Bayes*) ditunjukkan pada persamaan 3.

$$P(X | Y) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3}$$

Keterangan:

- $P(X | Y)$ = Probabilitas data dengan atribut X pada kelas Y (*posterior probability*)
- $P(Y)$ = probabilitas awal kelas Y (*prior probability*)
- σ = standar deviasi
- μ = rata-rata atribut dengan fitur kontinu
- $\prod_{i=1}^q P(X_i|Y)$ = probabilitas independen kelas Y dari semua fitur dalam vektor X

Proses implementasi metode *Naïve Bayes* dalam sentimen analisis, tiap dokumen direpresentasikan dengan atribut “ $X_1, X_2, X_3, \dots, X_n$ ”, untuk X_1 adalah kata pertama, X_2 adalah kata kedua, dan seterusnya. Pada proses klasifikasi, terdapat sejumlah kelas yang telah ditentukan pada proses pelatihan. Proses pelatihan probabilitas kelas $P(Y_j)$ dan probabilitas kata terhadap kelas $P(X_i | Y_j)$ dihitung dengan rumus seperti pada persamaan 4 dan 5.

$$P(Y_j) = \frac{|docs\ j|}{|sample|} \tag{4}$$

$$P(X_i | Y_j) = \frac{n_k+1}{n+|word|} \tag{5}$$

Keterangan:

- $|docs\ j|$ = jumlah dokumen setiap kelas j
- $|sample|$ = jumlah semua kelas
- n_k = jumlah frekuensi kemunculan setiap kata dalam dokumen
- n = jumlah frekuensi kemunculan kata dari setiap kelas
- $|word|$ = jumlah semua kosakata dari semua kelas

Setelah proses pelatihan selesai, maka selanjutnya yaitu melakukan proses klasifikasi data uji. *Naïve Bayes Classifier* digunakan untuk mencari nilai probabilitas tertinggi dari semua kelas berdasarkan data *train* yang dibentuk. Probabilitas tertinggi disebut sebagai nilai *posterior probability* tertinggi dari semua kelas atau yang disebut V_{MAX} . Persamaan V_{MAX} ditunjukkan pada persamaan 6.

$$V_{MAX} = \underset{Y_j \in V}{\arg\ max} \frac{P(x_1, x_2, x_3, \dots, x_n | Y_j) P(Y_j)}{P(x_1, x_2, x_3, \dots, x_n | Y_j)} \tag{6}$$

Keterangan:

- $P(x_1, x_2, x_3, \dots, x_n | Y_j)$ = probabilitas munculnya kata x di dalam dokumen kelas J
- $P(Y_j)$ = probabilitas dokumen kelas J di dalam semua dokumen
- $P(x_1, x_2, x_3, \dots, x_n)$ = probabilitas pola independen kata x_n

Nilai $P(x_1, x_2, x_3, \dots, x_n)$ adalah konstan untuk semua kelas Y_j , sehingga persamaan 6 dapat disederhanakan menjadi persamaan 7 atau 8

$$V_{MAX} = \underset{Y_j \in V}{\arg\ max} P(x_1, x_2, x_3, \dots, x_n | Y_j) P(Y_j) \tag{7}$$

$$V_{MAX} = \underset{Y_j \in V}{\arg\ max} \prod_{i=1}^q P(X_i | Y_j) P(Y_j) \tag{8}$$

3.5. Pengujian Klasifikasi

Pada penelitian ini, pengujian klasifikasi dilakukan dengan mengimplementasikan metode *Naïve Bayes Classification (NBC)*, yang kemudian ditentukan performa klasifikasinya dengan menggunakan *confusion matrix*. Dari *confusion matrix*, diperoleh hasil pengukuran akurasi, presisi, dan *recall*. Akurasi merupakan penilaian tentang keakuratan klasifikasi yang diperoleh dari perbandingan data positif dan negatif yang terklasifikasi benar dengan seluruh data. Presisi merupakan penilaian tentang relevansi data yang dicari dengan informasi yang dibutuhkan. *Recall* merupakan penilaian tentang keberhasilan sistem dalam menemukan kembali informasi. *Confusion matrix* disajikan dalam Tabel 2 [18].

Tabel 2. *Confusion Matrix*

Kelas Prediksi	Kelas Aktual	
	<i>True</i>	<i>False</i>
<i>True</i>	TP (<i>True Positive</i>)	FP (<i>False Positive</i>)
<i>False</i>	FN (<i>False Negative</i>)	TN (<i>True Negative</i>)

True Positive (TP) artinya apabila data yang diprediksi bernilai positif dan sesuai dengan nilai aktual (positif). *False Positive (FP)* artinya jika data yang diprediksi tidak sesuai dengan nilai aktual. *False Negative (FN)* artinya jika yang diprediksi bernilai negatif dan aktualnya positif. *True Negative (TN)* artinya jika benar antara prediksi negatif dan aktualnya negatif. Pengukuran nilai akurasi, presisi, *recall*, dan *F-Measure* berdasarkan *confusion matrix* dirumuskan pada persamaan 9 sampai 12:

$$\text{Akurasi} = \frac{TP + TN}{TP + FN + FP + TN} \quad (9)$$

$$\text{Presisi} = \frac{TP}{TP + FN} \quad (10)$$

$$\text{Recall} = \frac{TP}{TP + FP} \quad (11)$$

$$\text{F-Measure} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (12)$$

4. Hasil dan Pembahasan

Tahapan awal sebelum mengimplementasikan metode yaitu dengan melakukan pengumpulan data ulasan Puncak B29 Lumajang yang diperoleh dari *Google Maps API*. Pengumpulan data dilakukan dengan melakukan *scrapping* data dengan menggunakan situs *Outscraper*. Hasilnya, diperoleh sebanyak 500 ulasan dengan skor ulasan yang bervariasi dari 1 sampai 5 bintang, dengan rincian 3 ulasan bintang 1, 2 ulasan bintang 2, 9 ulasan bintang 3, 74 ulasan bintang 4, dan 412 ulasan bintang 5. Pada penelitian ini, klasifikasi berdasarkan ulasan dibagi menjadi 2 klasifikasi, yaitu sentimen positif untuk ulasan bintang 4 dan 5, dan sentimen negatif untuk ulasan bintang 1 sampai 3.

Setelah data dikumpulkan, maka tahapan selanjutnya yaitu *text preprocessing*. Pada tahapan ini, *text preprocessing* dilakukan dengan empat langkah, yaitu *case folding*, *tokenizing*, *stopwords*, dan *stemming*. Seluruh proses *text preprocessing* dilakukan dengan menggunakan bahasa *Python*. Pada penelitian ini, dilakukan impor dataset dengan library *pandas* untuk selanjutnya dilakukan serangkaian tahapan *text preprocessing*.

Langkah pertama yaitu *case folding*, yang dilakukan dengan mengubah seluruh data menjadi huruf kecil (*lowercase*) dan menghilangkan karakter selain a-z seperti simbol, angka, dan emoji. Langkah kedua yaitu *tokenizing*, yang dilakukan dengan memisahkan kata per kata seluruh data. Langkah ketiga yaitu *stopwords*, yang dilakukan dengan menghapus kata-kata yang terlalu umum dengan menggunakan library Sastrawi pada *Python*. Langkah keempat yaitu *stemming*, yang dilakukan dengan mengelompokkan kata-kata yang mempunyai kata dasar dan arti yang serupa namun memiliki makna yang berbeda. *Stemming* juga menggunakan library Sastrawi pada *Python*.

Setelah dilakukan *text preprocessing*, maka selanjutnya yaitu pemberian bobot per kata dari dataset, yang dilakukan dengan menggunakan TF-IDF. Dari total 500 ulasan pada dataset

yang dilakukan *text preprocessing*, sebanyak 9.214 kata yang diperoleh hingga tahapan *stemming*, dan terdapat 2.001 kata berbeda apabila kata yang terduplikasi dihilangkan. Pembobotan TF-IDF dilakukan dengan menghitung nilai TF yang diperoleh dari banyaknya kemunculan kata dari setiap ulasan. Hasil dari penghitungan nilai TF ditunjukkan pada Tabel 3.

Tabel 3. Hasil Penghitungan TF

Data	TF (Puncak)	TF (Jalan)	TF (Ojek)	TF (Motor)	TF (Pandang)
1	0	0	0	0	0
2	0	0	2	1	0
3	1	0	4	0	0
4	2	0	2	0	2
5	2	0	2	0	2
6	0	1	1	0	0
7	0	1	3	0	0
8	2	0	0	0	0
9	5	4	2	0	0
10	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮
500	1	1	1	1	1

Setelah mencari nilai TF, maka selanjutnya yaitu mencari nilai DF. Nilai DF ditentukan dengan menghitung banyaknya kemunculan kata dari setiap dataset yang lebih dari 0. Nilai DF terdapat pada Tabel 4.

Tabel 4. Nilai DF

DF (Puncak)	DF (Jalan)	DF (Ojek)	DF (Motor)	DF (Pandang)
155	151	146	116	145

Setelah mencari nilai DF, maka selanjutnya yaitu mencari nilai IDF dari masing-masing kata pada kata kunci tersebut. Nilai IDF berfungsi sebagai penyeimbang kata yang terlalu sering muncul pada semua dataset dengan menghitung log basis 10 dari jumlah dataset dibagi dengan nilai DF-nya. Makin besar DF-nya, makin kecil nilai IDF-nya, demikian sebaliknya. Berdasarkan nilai DF pada Tabel 3, maka nilai IDFnya terdapat pada Tabel 5.

Tabel 5. Nilai IDF

IDF (Puncak)	IDF (Jalan)	IDF (Ojek)	IDF (Motor)	IDF (Pandang)
0,5086	0,5199	0,5346	0,6345	0,5376

Setelah mencari nilai IDF, maka selanjutnya yaitu mencari nilai TF-IDF. Nilai TF-IDF diperoleh dari hasil perkalian antara nilai TF dan nilai IDF. Sebagai contoh nilai TF pada kata "puncak" pada dataset nomor 3 yaitu 2, sementara nilai IDF pada kata "puncak" yaitu 0,5086, maka nilai TF-IDFnya yaitu $2 \times 0,5086 = 1,0172$. Hasil penghitungan TF-IDF terdapat pada Tabel 6.

Tabel 6. Hasil Penghitungan TF-IDF

Data	TF-IDF (Puncak)	TF-IDF (Jalan)	TF-IDF (Ojek)	TF-IDF (Motor)	TF-IDF (Pandang)
1	0	0	0	0	0
2	0	0	1,0692	0,6345	0
3	0,5086	0	2,1384	0	0
4	1,0172	0	1,0692	0	1,0752
5	1,0172	0	1,0692	0	1,0752
6	0	0,5199	0,5346	0	0
7	0	0,5199	1,6038	0	0
8	0,5086	0	0	0	0
9	2,5432	2,0799	1,0692	0	0
10	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮
500	0,5086	0,5199	0,5346	0,6345	0,5376

Dataset yang digunakan untuk implementasi klasifikasi berupa jumlah atribut sebanyak 2.001 atribut yang berisikan kata-kata yang telah dilakukan *text preprocessing* dengan nilai masing-masing atribut sesuai dengan pembobotan kata dengan TF-IDF yang telah dilakukan. Sementara klasifikasi dibagi menjadi 2 kelas, yaitu kelas positif untuk ulasan bintang 4 dan 5 dan kelas negatif untuk ulasan bintang 1 sampai 3. Dalam penelitian ini, terdapat 14 dataset dengan kelas negatif dan 486 dataset dengan kelas positif.

Implementasi untuk klasifikasi menggunakan bantuan dari bahasa pemrograman *Python*. Tahapan awal sebelum dilakukan klasifikasi yaitu mengimpor dataset yang telah dibuat dan dikonversikan ke dalam format ekstensi *.csv* ke dalam kode program *Python*. Untuk mengimpor dataset, diperlukan *library* *pandas* dengan mengimpor *library* tersebut pada kode program. Kode program impor dataset terdapat pada Tabel 7.

Tabel 7. Kode Program Input Dataset

Baris	Kode
1	<code>import pandas as pd</code>
2	<code>import numpy as np</code>
3	
4	<code>puncak = pd.read_csv("D:\dataset 500 (1).csv")</code>
5	<code>puncak.info()</code>
6	<code>puncak</code>

Berdasarkan Tabel 7, pada baris 4-5 akan menghasilkan dataset yang diimpor. Dataset yang ditampilkan pada *Jupyter Notebook* hanya menampilkan 60 baris dataset (dari nomor 0 sampai 29 dan nomor 470 sampai 499) dan 20 kolom berisi 19 atribut dan 1 kelas, sehingga tidak semua isi dataset ditampilkan.

Implementasi klasifikasi dengan metode *Naïve Bayes* dilakukan dengan memanggil *library sklearn* pada *Python*. Jenis metode *Naïve Bayes* yang digunakan yaitu *Gaussian Naïve Bayes* karena seluruh atributnya merupakan atribut kontinu. Kode program untuk implementasi metode *Naïve Bayes Classification* ditunjukkan pada Tabel 8.

Tabel 8. Kode Program Implementasi *Naïve Bayes*

Baris	Kode
1	<code>from sklearn.model_selection import train_test_split</code>
2	<code>x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.4, random_state = 42)</code>
3	<code>y_test</code>
4	
5	<code>from sklearn.naive_bayes import GaussianNB</code>
6	<code>modelNB = GaussianNB()</code>
7	<code>nbtrain = modelNB.fit(x_train, y_train)</code>
8	
9	<code>y_pred = nbtrain.predict(x_test)</code>
10	<code>y_pred</code>
11	
12	<code>np.array(y_test)</code>

Berdasarkan kode program di atas, baris 1-3 melakukan pembagian dataset dengan *library sklearn.model_selection*. Perintah *test_size* 0.4 pada baris 2 artinya dataset akan dibagi menjadi 40% data testing dan 60% data training. Baris 5-7 yaitu program akan melakukan klasifikasi dengan *Gaussian Naïve Bayes*, dengan memanggil *library sklearn.naive_bayes*. Baris 9-12 yaitu program akan menampilkan kelas prediksi (*y_pred*) berdasarkan penerapan metode *Naïve Bayes* dan kelas aktual (*y_test*) dari data test yang dimasukkan ke dalam kode program

Uji performansi dilakukan untuk mengevaluasi hasil dari klasifikasi yang diimplementasikan dengan menggunakan BNN melalui pemetaan pada *confusion matrix*. Berdasarkan *confusion matrix*, maka dapat diketahui nilai dari akurasi, presisi, *recall*, dan *F1-score*. Pada penelitian ini, uji performansi dilakukan dengan pembagian dataset 40:60 (200 data uji), 30:70 (150 data uji) dan 20:80 (100 data uji), masing-masing dengan jumlah iterasi sebanyak 10, 25, dan 50 iterasi. Hasil pengujian akurasi, presisi, *recall*, dan *F1-score* untuk masing-masing pengujian terdapat pada Tabel 9.

Tabel 9. Hasil Pengujian Klasifikasi

Jumlah Data Test Set	Evaluasi			
	Akurasi	Presisi	Recall	F1-Score
100	90,00%	95,74%	93,75%	94,74%
150	92,67%	97,20%	95,21%	96,19%
200	90,50%	96,77%	93,26%	94,99%
Rata-rata	91,06%	96,57%	94,07%	95,31%

Berdasarkan Tabel 9, dapat disimpulkan bahwa untuk mengklasifikasi sentimen ulasan Puncak B29 dengan algoritma *Naïve Bayes Classification* (NBC) menghasilkan rata-rata akurasi sebesar 91,06%, presisi 96,57%, *recall* 94,07% dan *F1-score* 95,31%. Sementara untuk perolehan hasil terbaik terdapat pada pengujian dataset sebanyak 150 data, dengan akurasi sebesar 92,67%, presisi 97,20%, *recall* 95,21% dan *F1-score* 96,19%. Berdasarkan hasil tersebut, algoritma NBC mampu mengklasifikasi sentimen ulasan Puncak B29 dengan akurasi yang sangat baik, sesuai dengan penelitian terdahulu yang dilakukan oleh [3], [8], dan [11] dengan menggunakan algoritma NBC dan membandingkannya dengan algoritma klasifikasi lain, yang hasilnya algoritma NBC memperoleh skor akurasi yang paling tinggi.

5. Simpulan

Sentimen analisis ulasan Puncak B29 dengan menggunakan *Naïve Bayes Classification* (NBC) dilakukan dengan 3 tahapan, yang diawali dengan proses *scrapping data* untuk mengumpulkan data ulasan pada *Google Maps* dengan bantuan situs *Outscraper*, dan diperoleh sebanyak 500 data ulasan dari berbagai skor ulasan. Tahapan selanjutnya yaitu *text preprocessing* untuk menghapus tanda baca, menyaring kata-kata yang terlalu umum dan mengubah isi ulasan menjadi kata dasar, lalu memberikan bobot dari masing-masing kata dengan TF-IDF. Setelah dilakukan *text preprocessing* dan pembobotan kata, maka tahapan selanjutnya yaitu implementasi algoritma NBC. Setelah dilakukan implementasi, selanjutnya yaitu mengevaluasi hasil klasifikasi metode NBC, dengan menggunakan *confusion matrix* untuk mengukur akurasi, presisi, *recall*, dan *F1-score*. Klasifikasi sentimen ulasan Puncak B29 dengan algoritma NBC menghasilkan rata-rata akurasi sebesar 91,06%, presisi 96,57%, *recall* 94,07% dan *F1-score* 95,31%. Sementara untuk perolehan hasil terbaik terdapat pada pengujian dataset sebanyak 150 data, dengan akurasi sebesar 92,67%, presisi 97,20%, *recall* 95,21% dan *F1-score* 96,19%.

Daftar Referensi

- [1] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemandangan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)," *JTIK (Jurnal Teknologi Informasi dan Ilmu Komputer)*, vol. 8, no. 1, pp. 147–156, 2021, doi: 10.25126/jtiik.202183944.
- [2] T. Hendrawati and C. P. Yanti, "Analysis of Twitter Users Sentiment against the Covid-19 Outbreak Using the Backpropagation Method with Adam Optimization," *Journal of Electrical, Electronics and Informatics*, vol. 5, no. 1, pp. 1–4, 2021, doi: 10.24843/JEEI.2021.v05.i01.p01.
- [3] O. Somantri and Dairoh, "Analisis Sentimen Penilaian Tempat Tujuan Wisata Kota Tegal Berbasis Text Mining," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 5, no. 2, pp. 191–196, 2019, doi: 10.26418/jp.v5i2.32661.
- [4] A. Rifa, H. Sujaini, D. Prawira, and J. H. Hadari Nawawi, "Sentiment Analysis Objek Wisata Kalimantan Barat Pada Google Maps Menggunakan Metode Naive Bayes," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 7, no. 3, pp. 400–407, 2021, doi: 10.26418/jp.v7i3.48132.
- [5] S. Masturoh and A. B. Pohan, "Sentiment Analysis Against the Dana E-Wallet on Google Play Reviews Using The K-Nearest Neighbor Algorithm," *Pilar Nusa Mandiri Journal of Computing and Information System*, vol. 17, no. 1, pp. 53–58, 2021, doi: 10.33480/pilar.v17i1.2182.
- [6] N. T. Romadloni, I. Santoso, and S. Budilaksono, "Perbandingan Metode Naive Bayes, KNN Dan Decision Tree Terhadap Analisis Sentimen Transportasi KRL Commuter Line," *IKRA-ITH Informatika*, vol. 3, no. 2, pp. 1–9, 2019.
- [7] N. Habibah, E. Budianita, M. Fikry, and I. Iskandar, "Analisis Sentimen Mengenai Penggunaan E-Wallet Pada Google Play Menggunakan Lexicon Based dan K-Nearest

- Neighbor,” *JURIKOM (Jurnal Riset Komputer)*, vol. 10, no. 1, pp. 192–200, 2023, doi: 10.30865/jurikom.v10i1.5429.
- [8] S. Ulya, A. Ridwan, W. Cholid Wahyudin, and F. M. Hana, “Text Mining Sentimen Analisis Pengguna Aplikasi Marketplace Tokopedia Berdasar Rating dan Komentar Pada Google Play Store,” *Jurnal Bisnis Digital dan Sistem Informasi*, vol. 3, no. 2, pp. 33–40, 2022.
- [9] N. L. W. S. R. Ginantra, C. P. Yanti, G. D. Prasetya, I. B. G. Sarasvananda, and I. K. A. G. Wiguna, “Analisis Sentimen Ulasan Villa di Ubud Menggunakan Metode Naive Bayes, Decision Tree, dan K-NN,” *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 3, pp. 205–215, Dec. 2022, doi: 10.23887/janapati.v11i3.49450.
- [10] M. Izunahdi, G. Aburrahman, and A. E. Wardoyo, “Sentimen Analisis pada Data Ulasan Aplikasi KAI Access di Google Play Store menggunakan Metode Multinomial Naive Bayes,” *Jurnal Smart Teknologi*, vol. 4, no. 2, pp. 192–198, 2023.
- [11] M. K. Anam, B. N. Pikir, and M. B. Firdaus, “Penerapan Naive Bayes Classifier, K-Nearest Neighbor (KNN) dan Decision Tree untuk Menganalisis Sentimen pada Interaksi Netizen dan Pemerintah,” *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 1, pp. 139–150, Nov. 2021, doi: 10.30812/matrik.v21i1.1092.
- [12] I. K. A. G. Wiguna, P. Sugiartawan, I. G. I. Sudipa, and I. P. Y. Pratama, “Sentiment Analysis Using Backpropagation Method to Recognize the Public Opinion,” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 16, no. 4, pp. 423–434, 2022, doi: 10.22146/ijccs.78664.
- [13] A. Rahman, Wiranto, and A. Doewes, “Online News Classification Using Multinomial Naive Bayes,” *ITSMART: Jurnal Ilmiah Teknologi dan Informasi*, vol. 6, no. 1, 2017.
- [14] B. U. Manalu, “Analisis Sentimen Pada Twitter Menggunakan Text Mining,” *Teknologi Informasi Fak. ILKOM UNSUT*, 2014.
- [15] D. N. Chandra, G. Indrawan, and I. N. Sukajaya, “Klasifikasi Berita Lokal Radar Malang Menggunakan Metode Naive Bayes Dengan Fitur N-Gram,” *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 10, no. 1, 2016.
- [16] Y. Pratama, K. Januar Miraswan, and S. Selatan, “Sentiment Analysis Using Pseudo Nearest Neighbor and TF-IDF Text Vectorizer,” *Sriwijaya Journal of Informatic and Applications*, vol. 4, no. 2, pp. 20–26, 2023.
- [17] B. Krithiga, P. Sabari, I. Jayasri, and I. Anjali, “Early detection of coronary heart disease by using naive bayes algorithm,” in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1717/1/012040.
- [18] D. D. Saputra, B. Pratama, Y. Akbar, and W. Gata, “Penerapan Text Mining untuk Assignment Complaint Handling Customer Terhadap Divisi Tterkait menggunakan Metode Decision Tree Algoritma C4.5 (Studi Case: PT. XL AXIATA, TBK),” *Journal CKI On SPOT*, vol. 11, no. 2, pp. 207–216, 2018.