SENAFTI
SEMINAR NASIONAL MAHASISWA
FAKULTAS TEKNOLOGI INFORMASI

5th Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
7 September 2024 – Jakarta, Indonesia
Volume 3, Nomor 2, September 2024 - ISSN 2962-8628 (*online*)

PENERAPAN *TEXT MINING* UNTUK KLASIFIKASI INFORMASI BANJIR DI JAKARTA BERDASARKAN DATA TWITTER MENGGUNAKAN ALGORITMA *NAIVE BAYES*

Owen Meladiar[1], Indra[2*]

[1,2] Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia
Email: [1]2011500895@student.budiluhur.ac.id, [2*]indra@budiluhur.ac.id
(* : corresponding author)

Abstrak- Penelitian ini membahas klasifikasi tweet terkait banjir di DKI Jakarta menggunakan metode *Naïve Bayes Classifier*. Banjir merupakan masalah bencana yang sering terjadi di DKI Jakarta akibat berbagai faktor seperti curah hujan tinggi, tata kelola air yang buruk, dan pembangunan di bantaran sungai. DKI Jakarta memiliki sejarah panjang masalah banjir yang tercatat sejak tahun 1500-an, dengan kejadian signifikan pada tahun-tahun seperti 1621, 1654, 1671, 1918, 1976, 2002, 2007, dan 2013. Penelitian ini menggunakan data tweet yang dikumpulkan dari Twitter pada bulan Mei 2024 dengan jumlah data sebanyak 538 tweet yang dibagi menjadi dua kategori: banjir dan nonbanjir. Metode *text mining* dan klasifikasi *Naïve Bayes* diterapkan untuk mengidentifikasi tweet terkait banjir. Media sosial, khususnya Twitter, dipilih karena sifatnya yang cepat dalam menyebarkan informasi dan kemampuannya untuk menjangkau berbagai lapisan masyarakat. Twitter memungkinkan pengguna untuk berbagi informasi tentang lokasi banjir, kedalaman air, tingkat keparahan, dan kebutuhan bantuan secara real-time. Pengujian dilakukan dengan membagi dataset menjadi data latih dan data uji dengan rasio 80:20. Dari total dataset yang dimiliki, terdapat 334 data kelompok banjir dan 136 data kelompok nonbanjir. Data latih terdiri dari 376 data, sementara data uji terdiri dari 94 data. Hasil pengujian menunjukkan bahwa model klasifikasi memiliki tingkat akurasi 78,72%, *precision* 80,52%, *recall* 92,54%, dan *f1-score* 86,15%. Penelitian ini diharapkan memberikan kontribusi dalam implementasi text mining untuk mitigasi bencana banjir serta memberikan pengetahuan dalam bidang klasifikasi teks bahasa Indonesia. Selain itu, hasil penelitian ini dapat menjadi acuan untuk pengembangan model prediksi yang lebih baik di masa depan, serta memberikan wawasan tentang penggunaan media sosial sebagai sumber informasi dalam penanggulangan bencana. Penelitian ini juga dapat menjadi referensi untuk penelitian lanjutan yang bertujuan meningkatkan akurasi dan efektivitas metode klasifikasi teks dalam konteks bencana banjir di wilayah perkotaan**.**

Kata kunci : klasifikasi, *text mining, naïve bayes*, banjir jakarta.

IMPLEMENTATION OF TEXT MINING FOR FLOOD INFORMATION CLASSIFICATION IN JAKARTA BASED ON TWITTER DATA USING THE NAIVE BAYES ALGORITHM

Abstract*- This research discusses the classification of tweets related to floods in DKI Jakarta using the Naïve Bayes Classifier method. Flooding is a recurring disaster problem in DKI Jakarta, caused by various factors such as high rainfall, poor water management, and development along riverbanks. DKI Jakarta has a long history of flood issues dating back to the 1500s, with significant events occurring in years such as 1621, 1654, 1671, 1918, 1976, 2002, 2007, and 2013. This study uses tweet data collected from Twitter in May 2024, comprising 538 tweets categorized into two groups: flood and non-flood. Text mining techniques and the Naïve Bayes classification method were applied to identify flood-related tweets. Social media, especially Twitter, was chosen due to its rapid dissemination of information and ability to reach various segments of society. Twitter enables users to share real-time information about flood locations, water depth, severity, and aid requirements. The testing was conducted by dividing the dataset into training and testing data with a ratio of 80:20. From the total dataset, there were 334 tweets in the flood group and 136 tweets in the non-flood group. The training data consisted of 376 tweets, while the testing data included 94 tweets. The testing results showed that the classification model achieved an accuracy of 78.72%, precision of 80.52%, recall of 92.54%, and an F1-score of 86.15%. This research is expected to contribute to the implementation of text mining for flood disaster mitigation and provide insights into Indonesian text classification. Additionally, the findings of this study can serve as a reference for developing better prediction models in the future and offer insights into the use of social media as a source of information in disaster management. This research can also be a reference for future studies aimed at improving the accuracy and effectiveness of text classification methods in the context of urban flood disasters*.

Keywords*: Classification, Text Mining, Naïve Bayes, Jakarta Flood.*

1. PENDAHULUAN

Banjir adalah ketika banyak air meluap ke daratan yang biasanya kering karena curah hujan yang tinggi, lelehan salju, atau masalah lain yang menyebabkan air tidak dapat diserap dengan cepat oleh tanah atau dialirkan melalui saluran air. Banjir dapat terjadi dengan cepat atau secara bertahap.

Bencana banjir adalah masalah yang dihadapi oleh Provinsi DKI Jakarta. Faktor-faktor yang menyebabkan banjir termasuk curah hujan yang tinggi, permukaan tanah yang lebih rendah dibandingkan muka air laut, lokasi di cekungan yang dikelilingi perbukitan dengan sedikit resapan air, pembangunan bangunan di sepanjang bantaran sungai, sampah yang menghambat aliran sungai, dan kurangnya tutupan lahan di daerah hulu sungai. DKI Jakarta pernah banjir sekitar tahun 1500-an. Data menunjukkan banjir pada tahun 1621, 1654, 1671, 1918, 2976, 2002, 2007, dan 2013.

Di media sosial, masyarakat cenderung berbagi informasi tentang bencana yang mereka alami. Di sisi lain, orang-orang yang terkena banjir juga berbagi informasi tentang kejadian yang mereka alami, seperti lokasi banjir, kedalaman air, tingkat keparahan bencana, dan jumlah bantuan yang diperlukan. Media sosial dapat digunakan untuk mendapatkan informasi di lapangan dengan menggunakan publik sebagai sumbernya.

Sixdegree.com, situs jejaring sosial pertama, muncul pada tahun 1997. Setelah itu, pada tahun 1999, lahirlah Blogger, sebuah situs blog pribadi. Pengguna dapat membuat halaman web di situs web mereka sendiri melalui fitur ini. Kelebihannya adalah pengguna dapat menulis tentang topik apa pun, bahkan untuk pemerintahan atau individu. Friendster adalah situs jejaring sosial pertama yang didirikan pada tahun 2002. Pada tahun 2003, Linkedln mengikutinya. Situs ini bagus untuk mencari pekerjaan dan bersosialisasi. Selain itu, situs MySpace muncul pada tahun yang sama. Tahun berikutnya, Facebook, situs jejaring sosial yang masih bertahan hingga saat ini, muncul. Dua tahun kemudian muncul Twitter, yang unik karena penggunanya hanya dapat mengirimkan pesan sepanjang 140 karakter.

Twitter memutuskan untuk berkonsentrasi pada memberikan konten yang langsung dan segar kepada penggunanya. Selain itu, Twitter mengubah kategori dirinya dari layanan toko aplikasi mobile menjadi aplikasi berita, yang didukung oleh realitas jejaring sosialnya. Banyak perusahaan media, baik penyiaran, cetak, maupun berbasis internet, memiliki akun *Twitter*.

Twitter, dengan penggunanya yang berasal dari berbagai kalangan dan lapisan masyarakat, memungkinkan berbagai macam pendapat disampaikan pada setiap topik berita. Ini ditunjukkan oleh banyaknya pengguna *Twitter* yang berkicau tentang masalah banjir di wilayah DKI Jakarta

Berdasarkan hasil penelitian sebelumnya yang dilakukan oleh (Tasya & Putri) [1] diketahui bahwa setelah dilakukan analisis terhadap parameter faktor tingkat kerawanan banjir didapatkan hasil akurasi sebesar 99,187%. Nilai akurasi tersebut dapat dinyatakan sangat tinggi sebagai model prediksi.

Data yang dipakai dalam studi ini merupakan data tweet yang diperoleh dari Twitter melalui proses pengumpulan data. Tweet-tweet tersebut dibagi menjadi dua kelompok, yakni banjir dan nonbanjir. Hal ini bertujuan agar sistem klasifikasi dengan algoritma Naïve Bayes dapat membantu dalam memberikan pemahaman tentang pengelompokan teks di dalam tweet.

2. METODE PENELITIAN

Penelitian ini menerapkan teknik t*ext mining* melalui beberapa tahapan untuk mencapai tingkat akurasi yang tinggi. Tahapan-tahapan tersebut meliputi pengumpulan data, *preprocessing*, pelabelan, pembagian data, pemodelan, pengujian, dan visualisasi data. Dalam studi ini, peneliti menggunakan algoritma *Naïve Bayes*.

2.1 Pengumpulan Data

Pengumpulan Data adalah proses untuk mendapatkan *dataset* berupa *tweet* yang kemudian diolah untuk masuk ke tahap *preprocessing*[2]. Proses pengumpulan data pada penelitian ini menggunakan teknik *web scrapping* dengan menggunakan kata kunci “banjir jakarta”.

2.2 *Pre-Processing*

Tujuan dari *preprocessing* adalah untuk meningkatkan hasil analisis *text mining* dengan membuat data lebih mudah untuk dikelolah atau digunakan.[3]

Tahapan *text preprocessing* membersihkan data berita dari elemen yang tidak penting, membentuk pola kata yang sama, dan mengurangi volume kata untuk mempermudah klasifikasi. Ini meningkatkan efisiensi dan ketepatan komputasi.[4]

2.3 *Labelling*

Labelling adalah proses memberikan klasifikasi berdasarkan karakteristik yang terdapat dalam sebuah kalimat dalan dokumen. Pada penelitian ini, proses *labelling* diterapkan pada setiap *tweet* yang melewati proses *preprocessing*, dengan memberikan label banjir dan nonbanjir. Sebagian besar, pemberian label pada proses klasifikasi masih dilakukan secara manual oleh tim ahli pada dataset yang cukup besar. Oleh karena itu, proses pengklasifikasian dalam penelitian ini digunakan untuk menggantikan proses pelabelan manual pada dataset berskala besar.[5]

2.4 *Split* Data

Pada tahap ini data tweet yang sudah diberi label dibagi menjadi dua bagian dengan rasio 80:20, yaitu data latih sebesar 80% dan data uji sebesar 20% dari jumlah total data. Data latih digunakan sebagai dataset latih model, sedangkan data uji digunakan untuk menguji data serta mendapatkan nilai akurasi dari algoritma yang digunakan.[6]

2.5 *Modelling*

Untuk menghasilkan sebuah model pelatihan, ada beberapa langkah yang harus dilakukan, langkah-langkah tersebut antara lain pemilihan data pelatihan, tokenisasi, pengumpulan kata-kata, menghitung *vector token*, menghitung *term frequency* & *inverse document frequency* (TF-IDF), dan perkalian antara *term frequency* (TF) dengan *inverse document frequency* (IDF), dan pemodelan Naïve Bayes. [2]

Metode TF-IDF menggunakan transformasi data latih ke dalam bentuk TF-IDF untuk memberikan bobot pada data latih. Setelah data diubah menjadi TF-IDF, algoritma Multinomial Naive Bayes digunakan untuk melakukan proses pelatihan, dan hasilnya disimpan dan dibuat menjadi model latih dalam bentul file ekstensi (.model).[7]

2.6 Pengujian

Pada penelitian ini, dilakukan pengujian dengan dua cara yaitu pengujian sistem dan pengujian metode. Aplikasi yang dibuat yaitu Naive Bayes Classifier, untuk mengukur keakuratan penghitungan suatu data yaitu banjir dan nonbanjir, kemudian metode yang digunakan adalah *confusion matrix*.[8] Ada empat istilah dalam *confusion matrix* yang menjelaskan hasil pengukuran kinerja klasifikasi, yaitu *True Negative (TN), False Positive (FP), True Positive (TP), dan False Negative (FN)*.[9]

$$P(Y|Z) = \frac{P(Y)P(Z|Y)}{P(Z)} \quad (1)$$

Keterangan :

P(Y) : Probabilitas kejadian Y
P(Z) : Probabilitas kejadian Z
P(Y|Z) : Probabilitas kejadian Y berdasarkan kejadian Z
P(Z|Y) : Probabilitas kejadian Z berdasarkan kejadian Y

Confussion matrix adalah tabel matriks yang menampilkan deskripsi kinerja model klasifikasi pada rangkaian data uji *(testing)* yang nilai sebenarnya telah diketahui.[10]

Table 1. *Confession Matrix*

		Predicted Class	
		Banjir	NonBanjir
Actual Class	Banjir	(*True positive* = TP)	(*False positive* = FP)
	NonBanjir	(*False negative*) = FN)	(*True negative*= TN)

Keterangan:

TP (*True positive*) : Data yang bernilai positif, yang terklasifikasi dengan benar oleh sistem.
FN(*False negative*) : Data yang bernilai positif, tetapi terklasifikasi salah oleh sistem.
FP (*False positive*) : Data yang bernilai negatif, tetapi terklasifikasi salah oleh sistem.
TN (*True negative*) : Data yang bernilai negatif, yang terklasifikasi benar oleh sistem.

Seperti yang sudah dijelaskan, pengukuran tingkat akurasi, *precision*, *recall* dan *f1-score* dapat diketahui melalui *Confusion matrix* dengan penjelasan rumus sebagai berikut:[10]

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$\textit{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\textit{Recall} = \frac{TP}{TP+FN} \quad (4)$$

$$\textit{F1-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Peneliti ini menggunakan data yang diambil dari media sosial Twitter atay X sebagai dataset. Pengumpulan data tweet dilakukan melalui *Tweet Harvest* yang didapatkan dari GitHub (https://github.com/helmisatria/tweet-harvest). Kata kunci yang digunakan untuk meng-*crawling* data dari twitter atau X adalah banjir jakarta. Total jumlah tweet yang terkumpul mencapai 538 tweet. *Crawling* data dilakukan dari bulan Mei 2024 hingga bulan Juni 2024.
Data hasil *crawling* ini kemudian disimpan dalam format excel untuk diproses pelabelan otomatis dan validasi manual oleh pakar.

Table 2. Sample Data Banjir

No	Tweet	Label
1.	@andhikalfariz Pak Heru Budi menangani banjir dan macet di jakarta	Banjir
2.	Mengatasi Banjir Jakarta	Banjir
3.	Keruk Lumpur Sungai Efektif Kurangi Banjir di Jakarta https://t.co/yBMiKhXCAh	NonBanjir
4.	@Tan_Mar3M Kalo saya jadi Gubernur DKI Jakarta mudah mengatasi banjir &	NonBanjir

3.2 *Pre-Processing*

Pada Langkah ini, data mentah yang diperoleh dari proses *crawling* diubah secara signifikan menjadi data yang telah melalui proses pembersihan untuk memungkinkan pengolahan oleh sistem. Berikut adalah informasi mengenai tahapan *preprocessing*.

Table 3. Sampel proses *preprocessing*

Tahapan *Preprocessing*	Hasil
Tweet Asli	@andhikalfariz Pak Heru Budi menangani banjir dan macet di jakarta
Case Folding + Cleansing	pak heru budi menangani banjir dan macet di jakarta
Tokenization	[“pak”, “heru”, “budi”, “menangani”, “banjir”, “dan”, “macet”, “di”, “jakarta”]
Replace Slang Word	[“pak”, “heru”, “budi”, “menangani”, “banjir”, “dan”, “macet”, “di”, “jakarta”]
Remove Stop Word	[“heru”, “budi”, “menangani”, “banjir”, “macet”, “jakarta”]
Stemming	[“heru”, “budi”, “tangan”, “banjir”, “macet”, “jakarta”]

3.3 *Split* data

split data adalah dimana data dibagi secara acak dengan rasio 80 (data latih): 20 (data uji). Tahapan ini penting untuk memastikan bahwa model yang dibuat tidak terganggu karna tidak terganggu ketidakseimbangan data, maka dari itu data dapat diuji secara valid. Dalam penelitian ini, 470 *tweet* dibagi menjadi 376 *tweet* data latih dan 94 *tweet* data uji. Kemudian data disimpan didalam basis data untuk diolah pada tahapan *modelling*.

3.4 *Modelling*

Tahap *Modelling* merupakan langkah yang dilakukan setelah *tweet* melewati proses *preprocessing, labelling,* dan *split data.* Tahap ini bertujuan untuk memperoleh model pelatihan. Terdapat beberapa proses utama dalam tahap ini, yaitu seleksi data latih, tokenisasi, pengumpulan kata, menghitung *vektor token*, penghitungan *Term Frequency (TF),* perhitungan *Inverse Document Frequency (IDF),* mengalikan *Term Frequency* dengan *Inverse Document Frequency*, pemodelan *Multinomial Naïve Bayes*, serta menyimpan hasilnya dalam ekstensi (model). Tahapan tahapan pembobotan TF-IDF dapat dilihat lebih rinci sebagai berikut.

a. Perhitungan *Term Frequency-Inverse Document Frequency (TF-IDF)*

Menghitung nilai TF-IDF adalah langkah selanjutnya setelah mendapatkan nilai IDF, nilai *TF-IDF* adalah perkalian antara nilai TF dan nilai IDF.

Table 4. Perhitungan *(TF-IDF)*

Kata	TF - IDF D1	TF - IDF D2	TF - IDF D3	TF- IDF D4
pak	0,066	0	0	0
heru	0,066	0	0	0
budi	0,066	0	0	0
menangani	0,066	0	0	0
banjir	0	0	0	0
dan	0,066	0	0	0
macet	0,066	0	0	0
di	0,033	0	0,037	0
jakarta	0	0	0	0
mengatasi	0	0,100	0	0,030
keruk	0	0	0,075	0
lumpur	0	0	0,075	0
sungai	0	0	0,075	0
efektif	0	0	0,075	0
kurangi	0	0	0,075	0
kalo	0	0	0	0,060
saya	0	0	0	0,060
jadi	0	0	0	0,060
gubernur	0	0	0	0,060
dki	0	0	0	0,060
mudah	0	0	0	0,060
amp	0	0	0	0,060

b. *Modelling Naïve Bayes*

Setelah memperoleh hasil pembobotan TF-IDF diatas, langkah selanjutnya adalah menghitung jumlah tweet yang termasuk dalam kelas banjir dan nonbanjir. Setelah memperoleh probabilitas prior dari setiap kelas, langkah berikutnya adalah menghitung probabilitas likelihood. Langkah ini melibatkan perhitungan setiap kata dalam kelas

banjir dan nonbanjir dengan rumus (Jumlah kemunculan kata di kelas positif + 1) / (Total jumlah kata dalam kelas + Jumlah kata unik dalam kelas). Berikut adalah tabel yang menunjukkan probabilitas *likelihood* kata dalam kelas banjir dan nonbanjir.

Table 5. *Probabilitas Likelihood*

Kata	Probabilitas *Likelihood* Banjir	Probabilitas *Likelihood* NonBanjir
pak	(1+1) / (9+18) = 0,074	(0+1) / (9+18) = 0,037
heru	(1+1) / (9+18) = 0,074	(0+1) / (9+18) = 0,037
budi	(1+1) / (9+18) = 0,074	(0+1) / (9+18) = 0,037
menangani	(1+1) / (9+18) = 0,074	(0+1) / (9+18) = 0,037
banjir	(2+1) / (12+22) = 0,111	(2+1) / (12+22) = 0,111
macet	(1+1) / (9+18) = 0,074	(0+1) / (9+18) = 0,037
jakarta	(2+1) / (12+22) = 0,111	(2+1) / (12+22) = 0,111
mengatasi	(1+1) / (9+18) = 0,074	(1+1) / (9+18) = 0,074
keruk	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
lumpur	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
sungai	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
efektif	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
kurangi	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
saya	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
jadi	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
gubernur	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
dki	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074
mudah	(0+1) / (9+18) = 0,037	(1+1) / (9+18) = 0,074

3.5 Klasifikasi *Naïve Bayes*

Berikut adalah penjelasan mengenai tahap klasifikasi *Naïve Bayes*:

a. Persiapan Data

Persiapan data melibatkan pemilihan model pelatihan. Model pelatihan yang dipilih digunakan sebagai dasar untuk mengklasifikasikan data uji yang tersedia. Data uji yang digunakan adalah contoh data uji yang tercantum dalam tabel berikut.

Table 6. Data uji

Data Uji	Tweet	Label
Uji 1	jakarta banjir ikn banjir trus bedanya apa	Banjir

b. Prediksi Klasifikasi *Naïve Bayes*

Untuk memprediksi kelas dari data uji tersebut, kita bisa mengestimasi probabilitas posterior untuk setiap kelas menggunakan pendekatan Naive Bayes.

P(banjir|Uji 1) = P(banjir|jakarta) * P(banjir|banjir) * P(banjir |ikn) * P(banjir|trus) * P(banjir|bedanya) * P(banjir|apa)

P(banjir|Uji 1) = 0,111 * 0,111 * 0,037 * 0,037 * 0,037 * 0,037 = 0,000000023230573

P(nonbanjir|Uji 1) = P(nonbanjir|jakarta) * P(nonbanjir|banjir) * P(nonbanjir |ikn) * P(nonbanjir|trus) * P(nonbanjir|bedanya) * P(nonbanjir|apa)

P(nonbanjir|Uji 1) = 0,085 * 0,085 * 0,028 * 0,028 * 0,028 * 0,028 = 0,000000004895794

Table 7. Hasil Prediksi *Naive Bayes*

Data Uji	Tweet	Label	Probabilitas	Label Prediksi
Uji 1	jakarta banjir ikn banjir trus bedanya apa	Banjir	Banjir = 0,000000023230573 NonBanjir = 0,000000004895794	Banjir

3.6 Pengujian

Pengujian adalah bagian penting dari setiap proses pengembangan sistem karena memungkinkan untuk menilai, menganalisis, dan memahami seberapa akurat atau kesesuaian hasil yang telah dicapai oleh sistem yang dirancang. Penelitian ini menguji akurasi, presisi, dan recall algoritma Naive Bayes untuk memprediksi label pada data uji. Sebanyak 94 data uji telah diprediksi, dan hasilnya direpresentasikan menggunakan *Confusion Matrix*.

Table 8. Pengujian *Confusion Matrix*

Confusion Matrix		Predicted Values	
		Banjir	NonBanjir
Actual Values	Banjir	62	5
	NonBanjir	15	12

Berdasarkan *Confusion Matrix* diatas, perhitungan rumus untuk menghitung nilai akurasi, presisi, *recall*, dan f1 score dapat dilakukan berdasarkan yang sudah dijelaskan di sub bab (2.6). Rumus tersebut dapat ditemukan dalam table 2.1.

Table 9. Pengujian Accuracy, Precision, Recall, F1 Score

	Pengujian	
Accuracy	$\frac{62+12}{62+12+15+5}$	0,7872 78,72%
Precision	$\frac{62}{62+15}$	0,8052 80,52%
Recall	$\frac{62}{62+3}$	0,9254 92,54%
F1 Score	$2 \times \frac{0,8052 \times 0,9254}{0,8052+0,9254}$	0,8615 86,15%

Berdasarkan hasil pengujian di atas, dapat diketahui bahwa hasil pengujian menggunakan Algoritme *Naïve Bayes* yaitu memperoleh nilai akurasi 78,72%, *precision* 80,52%, *recall* 92,54%, dan *f1-score* 86,15%.

4. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan pada klasifikasi teks banjir daerah jakarta menggunakan metode *Naïve Bayes* melalui media sosial twitter dengan jumlah datas sebanyak 470 data dan menggunakan perbandingan 80:20 lalu data tersebut dipecah menjadi 2 bagian, yaitu data latih dan data uji. Masing-masing data tersebut memiliki nilai yang berbeda, data latih sebanyak 376 data dan data uji sebanyak 94 data. Dari total dataset yang dimiliki, terdapat 327 data kelompok banjir dan 143 data kelompok nonbanjir. Pengujian ini dilakukan menggunakan metode *Naïve Bayes* dengan perbandingan 80:20 dan mendapat nilai akurasi 78,72%, *precision* 80,52%, *recall* 92,54%, dan *f1-score* 86,15%.

DAFTAR PUSTAKA

[1] Y. Tasya and R. A. Putri, "Klasifikasi Tingkat Kerawanan Banjir Wilayah Medan Menggunakan Metode Naive Bayes Dan Algoritma J48," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 6, no. 2, 2023, doi: 10.31539/intecoms.v6i2.7392.

[2] F. Farhan, T. Triase, and A. M. Harahap, "Penggunaan Algoritma Naive Bayes Dalam Text Mining Untuk Klasifikasi Pasal UU ITE," *J-SISKO TECH (Jurnal Teknol. Sist. Inf. dan Sist. Komput. TGD)*, vol. 6, no. 2, pp. 314-322, 2023, doi: 10.53513/jsk.v6i2.7896.

[3] A. E. Wibowo, et al, "Text Mining: Sistem Prediksi Cyberbullying pada Platform Twitter menggunakan Logistic Regression, KNN, dan Naive Bayes," *J. Rekayasa Elektro Sriwij.*, vol. 4, no. 1, pp. 17-23, 2023, doi: 10.36706/jres.v4i1.56.

[4] S. M. Habib, et al, "Klasifikasi Berita Menggunakan Metode Naïve Bayes Classifier," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 2, pp. 248-258, 2022, doi: 10.32672/jnkti.v5i2.4191.

[5] A. Z. M. S. Widodo, A. P. Kusuma, and W. D. Puspitasari, "Analisis Algoritma Naive Bayes Classifier (Nbc) Pada Klasifikasi Tingkat Minat Barang Di Toko Violet Cell," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 87-94, 2023, doi: 10.36040/jati.v7i1.5692.

[6] A. R. Isnain, et al, "Analisis Perbandingan Algoritma LSTM dan Naive Bayes untuk Analisis Sentimen," *J. Edukasi dan Penelit. Inform.*, vol. 8, no. 2, pp. 299-303, 2022, doi: 10.26418/jp.v8i2.54704.

[7] N. Widiastuti, A. Hermawan, and D. Avianto, "Implementasi Metode Naïve Bayes Untuk Klasifikasi Data Blogger," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 8, no. 3, pp. 985-994, 2023, doi: 10.29100/jipi.v8i3.3713.

[8] A. Deolika, K. Kusrini, and E. T. Luthfi, "Analisis Pembobotan Kata Pada Klasifikasi Text Mining," *J. Teknol. Inf.*, vol. 3, no. 2, pp. 179-184, 2019, doi: 10.36294/jurti.v3i2.1077.

[9] A. R. Harungguan, H. Napitupulu, and F. Firdaniza, "Analisis Sentimen Dengan Metode Klasifikasi Naïve Bayes dan Seleksi Fitur Chi-Square," *In Search*, vol. 22, no. 2, pp. 92-99, 2023, doi: 10.37278/insearch.v22i2.762.

[10] N. S. Kustanto, N. Gusriani, and F. Firdaniza, "Analisis Sentimen dengan Metode Klasifikasi Naïve Bayes Dan Seleksi Fitur Information Gain," *In Search*, vol. 21, no. 2, pp. 134-144, 2022, doi: 10.37278/insearch.v21i2.524.