

ANALISIS SENTIMEN REVIEW APLIKASI PEDULILINDUNGI PADA GOOGLE PLAY STORE MENGGUNAKAN NBC

George Kenneth Locarso

Universitas Bunda Mulia

Jl. Jalur Sutera Barat Kav 7 – 9 Alam Sutera Tangerang 15143

E-mail : georgelocarso@gmail.com

ABSTRACT

Sentiment from the community is an important factor in the government carrying out a policy. By paying attention to feedback from the public, the government can find out what can be changed or improved and what has been achieved. This study aims to build a system that can perform sentiment analysis, so that reviews from the public can be more easily filtered by interested parties. Sentiment analysis will be implemented using the naive Bayes classifier algorithm. By using 1179, the training data and testing data are divided by a ratio of 70:30. From the results of the classification of the testing data, the accuracy is 83.3%.

Keyword : sentiment analysis, naïve bayes classifier, nbc, pedulilindungi

ABSTRAK

Sentimen dari masyarakat merupakan faktor yang penting dalam pemerintah melakukan suatu kebijakan. Dengan memperhatikan feedback dari masyarakat pemerintah dapat mengetahui apa saja yang dapat diubah atau ditingkatkan dan apa yang sudah dicapai. Penelitian ini bertujuan membangun sistem yang dapat melakukan analisis sentimen, sehingga ulasan dari masyarakat dapat lebih mudah disaring oleh pihak yang berkepentingan. Analisis sentiment akan diimplementasikan menggunakan algoritma naïve bayes classifier. Dengan menggunakan 1179, data training dan data testing dibagi dengan perbandingan 70:30. Dari hasil klasifikasi pada data testing, didapatkan akurasi sebesar 83.3%.

Kata kunci : analisis sentimen, naïve bayes classifier, nbc, pedulilindungi

1. PENDAHULUAN

Medial sosial adalah sebuah media yang dapat diakses secara daring, setiap penggunaanya dapat dengan mudah mendapatkan dan membagikan informasi, berdiskusi, dan membuat konten yang dapat di publikasi Bahasa [1]. Banyak masyarakat Indonesia yang menggunakan media sosial untuk

menyampaikan opininya terkait kebijakan kebijakan pemerintah, program-program pemerintah ataupun keluhan keluhan yang ingin disampaikan.

Pada 2 Maret 2020, Presiden Joko Widodo mengumumkan kasus pertama dari penyakit covid-19 [2]. Setelah hampir 11 bulan program vaksin untuk

covid-19 mulai dilaksanakan. Pada tanggal 13 Januari 2021, Presiden Joko Widodo merupakan individu yang menerima vaksin pertama. Pemerintah meluncurkan program pelacakan individu yang akan bepergian ke tempat umum dengan aplikasi PeduliLindungi. Aplikasi ini dapat membantu pemerintah terkait dalam melakukan pelacakan untuk menghentikan penyebaran covid-19. Masyarakat yang akan masuk ke suatu tempat umum wajib melakukan *check-in* pada aplikasi PeduliLindungi dan hanya boleh masuk jika sudah mendapatkan 2 dosis vaksin dan melakukan *check-out* ketika meninggalkan tempat tersebut. Aplikasi PeduliLindungi juga digunakan untuk masyarakat untuk mendaftar vaksinasi, melihat status vaksinasi, melihat sertifikat vaksin, dan melihat status keamanan bepergian tiap individu.

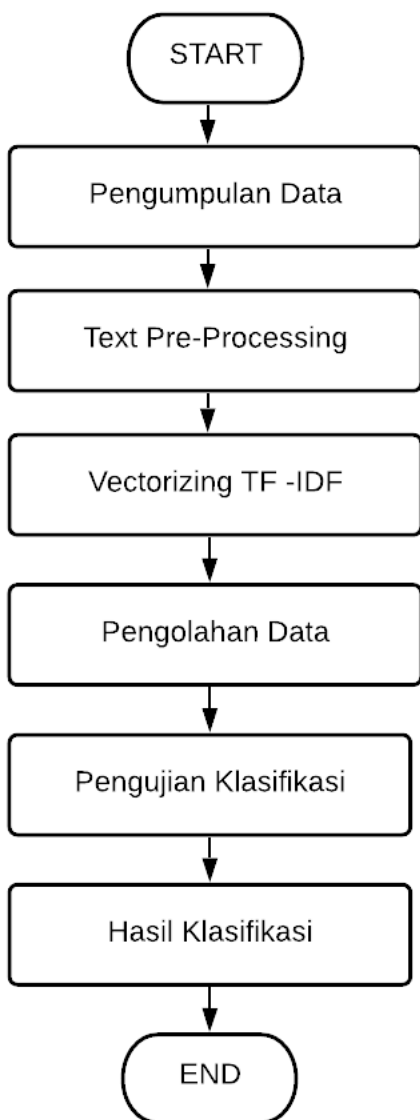
Dengan kehadiran aplikasi PeduliLindungi menimbulkan opini dan kritikan dari masyarakat. Setiap orang memiliki pendapat dan pemikirannya masing-masing yang terdiri dari opini yang pro maupun kontra terhadap aplikasi ini.

Dilihat dari permasalahan yang ada, diperlukan solusi seperti analisis sentimen terhadap komentar dari masyarakat sehingga dapat diketahui informasi sentimen terkait aplikasi PeduliLindungi yang diluncurkan oleh pemerintah Indonesia. Agar pemerintah Indonesia dapat memperhatikan masalah-masalah yang dialami oleh para pengguna aplikasi. Peneliti akan menggunakan data ulasan atau komentar dari aplikasi PeduliLindungi pada Google Play Store. Pada penelitian ini penulis akan menggunakan

algoritma *Naïve Bayes Classifier* (NBC).

2. METODOLOGI

Gambar 1 menunjukkan proses penelitian dilakukan dengan melakukan proses pengumpulan data, *text pre-processing*, *vectorizing TF-IDF*, pengolahan data, pengujian klasifikasi, dan hasil klasifikasi.



Gambar 1. Proses Penelitian

2.1. Pengumpulan Data

Pada tahap pengumpulan data, penulis menggunakan data ulasan aplikasi PeduliLindungi dari GooglePlay Store.

Tabel 1. Contoh Dataset

No.	Comment	Text
1	pedulilindungi menjadi filter hanya bagi yg telah di vaksin yang dapat diberikan akses untuk dapat leluasa beraktifitas, sehingga akan mengurangi ketersebaran covid 19	POSITIF
2	saat mengisi e-hac dan memilih di kolom kendaraan yg digunakan, pilih airplane tapi ketika akan disubmit malah berubah jadi car.	NEGATIF
3	sangat membantu saya dan aplikasi ini sangat mudah di gunakan	POSITIF
4	terimakasih dgn pedulilindungi diri ini kita bisa lebih waspada disekitar kita..	POSITIF
5	saya tidak bisa mengunduh sertifikat vaksin padahal sudah vaksin kedua	NEGATIF

2.2. Text Pre-Processing

Text Pre-processing adalah tahap memproses data agar dapat digunakan untuk proses analisa. Proses *pre-processing* text berupa *cleaning, case-folding, tokenizing, stemming, filtering* [3].

2.2.1. Casefolding

Tahap *case-folding* mengubah kata yang berhuruf besar menjadi kata

Sebelum Replacement	Setelah Replacement
['aplikasi', 'ini', 'merugikan', 'kesalahan', 'pihak', 'pemerintah', 'yang', 'tdk', 'input', 'data', 'tapi', 'masyarakat', 'yang', 'dirugikan', 'tdk', 'bs', 'kemana', 'krn', 'datanya', 'tdk', 'ada', 'di', 'aplikasi', 'ini']	['aplikasi', 'ini', 'merugikan', 'kesalahan', 'pihak', 'pemerintah', 'yang', 'tidak', 'input', 'data', 'tapi', 'masyarakat', 'yang', 'dirugikan', 'tidak', 'bisa', 'kemana', 'karena', 'datanya', 'tidak', 'ada', 'di', 'aplikasi', 'ini']

berhuruf kecil dan menghapus tanda baca.

Sebelum Tokenizing	Setelah Tokenizing
aplikasi ini merugikan kesalahan pihak pemerintah yang tdk input data tapi masyarakat yang dirugikan tdk bs kemana krn datanya tdk ada di aplikasi ini	['aplikasi', 'ini', 'merugikan', 'kesalahan', 'pihak', 'pemerintah', 'yang', 'tdk', 'input', 'data', 'tapi', 'masyarakat', 'yang', 'dirugikan', 'tdk', 'bs', 'kemana', 'krn', 'datanya', 'tdk', 'ada', 'di', 'aplikasi', 'ini']

Tabel 2. Contoh Proses Casefolding

2.2.2. Cleaning

Tahap *cleaning* akan dilakukan pembuangan atau pembersihan pada dokumen yang berisi url, angka, dan

simbol-simbol yang tidak diperlukan dalam pengolahan data.

Tabel 3. Contoh Proses Cleaning

2.2.3. Tokenizing

Tahap *tokenizing* melakukan pemisahan kalimat menjadi kata kata yang terpisah.

Tabel 4. Contoh Proses Tokenizing

Sebelum Casefolding	Setelah Casefolding
Aplikasi ini merugikan, kesalahan pihak pemerintah yang TDK input data tapi masyarakat yang dirugikan TDK bs kemana2 krn datanya tdk ada di aplikasi ini.	aplikasi ini merugikan, kesalahan pihak pemerintah yang tdk input data tapi masyarakat yang dirugikan tdk bs kemana2 krn datanya tdk ada di aplikasi ini.

2.2.4. Replacement

Tahap *replacement* merupakan tahap menyaring kata-kata yang tidak baku, berdasarkan kamus kata yang disediakan penulis.

Tabel 5. Contoh Proses Replacement

Sebelum Cleaning	Setelah Cleaning
aplikasi ini merugikan, kesalahan pihak pemerintah yang tdk input data tapi masyarakat yang dirugikan tdk bs kemana2 krn datanya tdk ada di aplikasi ini.	aplikasi ini merugikan kesalahan pihak pemerintah yang tdk input data tapi masyarakat yang dirugikan tdk bs kemana krn datanya tdk ada di aplikasi ini

2.2.5. Stemming

Tahap *Stemming* mengubah kata-kata yang berimbuhan menjadi kata dasar.

Tabel 6. Contoh Proses *Stemming*

2.2.6. Filtering Stopwords

Pada proses *filtering* atau *Stopword Removing*, menghapus kata-kata yang tidak penting atau tidak bermakna untuk analisis sentimen. Akan disediakan sebuah kumpulan kata-kata tidak penting yang jika ditemukan di dalam dokumen yang sedang diproses maka kata tersebut akan dihapus dari dokumen.

Tabel 7. Contoh Proses *Filtering*

2.3. TF-IDF

Pembobotan kata bertujuan memberikan bobot pada setiap kata yang terdapat dalam, kalimat/dokumen yang sudah melalui tahap pre-processing. Untuk mendapatkan nilai TF-IDF, akan dilakukan perhitungan *Term Frequency* (TF), dan *Inverse Document Frequency* (IDF). *Term Frequency* (TF) merupakan proses untuk menghitung jumlah frekuensi suatu kata terdapat dalam keseluruhan dokumen dan memberikan nilai untuk kata tersebut [4].

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d} , & \text{if } tf_{t,d} > 0 \\ 0, & \text{if } tf_{t,d} = 0 \end{cases}$$

(1)

Term (*t*) ditentukan dengan menghitung berapa kali term muncul pada keseluruhan dokumen. Dokumen (*d*) merupakan jumlah total semua kata yang terdapat dalam dokumen [5]. Lalu untuk mendapatkan bobot tf, adalah dengan melakukan $1 + \log_{10}$. Jika nilai $t = 0$ maka bobot $tf = 0$.

Sebelum Stemming	Setelah Stemming
['aplikasi', 'ini', 'merugikan', 'kesalahan', 'pihak', 'pemerintah', 'yang', 'tidak', 'input', 'data', 'tapi', 'masyarakat', 'yang', 'dirugikan', 'tidak', 'bisa', 'kemana', 'karena', 'datanya', 'tidak', 'ada', 'di', 'aplikasi', 'ini']	['aplikasi', 'ini', 'rugi', 'salah', 'pihak', 'perintah', 'yang', 'tidak', 'input', 'data', 'tapi', 'masyarakat', 'yang', 'rugi', 'tidak', 'bisa', 'mana', 'karena', 'data', 'tidak', 'ada', 'di', 'aplikasi', 'ini']

Sebelum Filtering	Setelah Filtering
['aplikasi', 'ini', 'rugi', 'salah', 'pihak', 'perintah', 'yang', 'tidak', 'input', 'data', 'tapi', 'masyarakat', 'yang', 'rugi', 'tidak', 'bisa', 'mana', 'karena', 'data', 'tidak', 'ada', 'di', 'aplikasi', 'ini']	['aplikasi', 'ini', 'rugi', 'salah', 'pihak', 'perintah', 'yang', 'tidak', 'input', 'data', 'tapi', 'masyarakat', 'yang', 'rugi', 'tidak', 'bisa', 'mana', 'karena', 'data', 'tidak', 'ada', 'di', 'aplikasi', 'ini']

Inverse Document Frequency (IDF) merupakan proses ketika suatu term banyak muncul maka, nilai bobot dari term tersebut akan dikurangi.

$$idf_t = \log_{10} \left(\frac{N}{df_t} \right)$$

(2)

Bobot dari idf didapatkan dengan melakukan \log_{10} (jumlah kalimat/jumlah kalimat yang mengandung $term$ [6].

Bobot setiap kata untuk menghitung TF-IDF adalah dengan menggunakan persamaan berikut :

$$w_{t,d} = tf_{t,d} \times idf_t \quad (3)$$

2.4. Pengolahan Data

Algoritma NBC adalah metode yang melakukan klasifikasi berdasarkan data statistik yang digunakan untuk melakukan klasifikasi probabilitas keanggotaan suatu $class$. NBC memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan pada data dalam jumlah yang besar [7]. Rumus NBC memiliki bentuk umum sebagai berikut (1) :

$$P(X) = \frac{P(H)P(H)}{P(X)}$$

(4)

Keterangan :

X = Data dengan $class$ yang belum diketahui

H = Hipotesis data X merupakan suatu $class$ spesifik

$P(H|X)$ = Probabilitas hipotesis H berdasarkan kondisi X

$P(H)$ = Probabilitas hipotesis H

$P(X|H)$ = Probabilitas hipotesis X berdasarkan kondisi H

$P(H)$ = Probabilitas hipotesis H

2.5. Pengujian Klasifikasi Confusion Matrix

Confusion Matrix adalah merupakan matriks yang berisi nilai prediksi benar dan prediksi salah. Dalam pengujian keakuratan hasil pencarian akan dievaluasi nilai $recall$, $precision$, dan $accuracy$ [8]. Dimana $precision$ mengevaluasi kemampuan sistem untuk menemukan peringkat yang paling relevan, dan didefinisikan sebagai sebagian dari dokumen yang di ambil dan benar-benar relevan terhadap query. $Recall$ mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dari koleksi dokumen dan didefinisikan sebagai sebagian dari dokumen yang relevan terhadap query. $Accuracy$ merupakan perbandingan kasus yang diidentifikasi benar dengan jumlah seluruh kasus dan $error rate$ merupakan kasus yang diidentifikasi salah dengan jumlah seluruh kasus [9]. Tabel 1 memperlihatkan 4 bagian dari *confusion matrix*, terdiri dari *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN).

Tabel 8. Contoh Confusion Matrix

CONFUSION MATRIX	Prediction Positive	Prediction Negative
Actual Positive	TP	FP
Actual Negative	FN	TN

TP, FP, TN, FN merupakan parameter yang digunakan untuk menghitung nilai $accuracy$, $precision$, dan $recall$ [10]. Pada persamaan

(5), (6), (7) digunakan untuk menilai hasil dari model yang sedang di uji.

Berikut persamaan untuk mendapatkan nilai *accuracy* :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Berikut persamaan untuk mendapatkan nilai *precision* :

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (6)$$

Berikut persamaan untuk mendapatkan nilai *recall* :

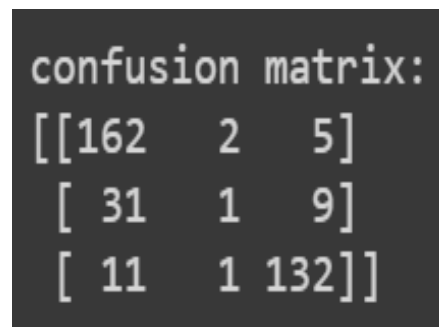
$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (7)$$

3. HASIL DAN PEMBAHASAN

Data ulasan yang didapatkan dari *review* aplikasi pada platform GooglePlay Store. Data komentar terdiri dari 1179 data yang terdiri dari 513 komentar positif, 144 komentar netral dan 522 komentar negatif. Data komentar kemudian dibagi menjadi 70% data latih sejumlah 825, dan 30% data uji sejumlah 354. Data terdiri dari 2 kolom yaitu kolom yang berisi komentar dan kolom yang berisi label yang diberikan pada komentar. Terdapat 3 jenis label yang akan diberikan pada setiap komentar yaitu, Positif, Netral, dan Negatif.

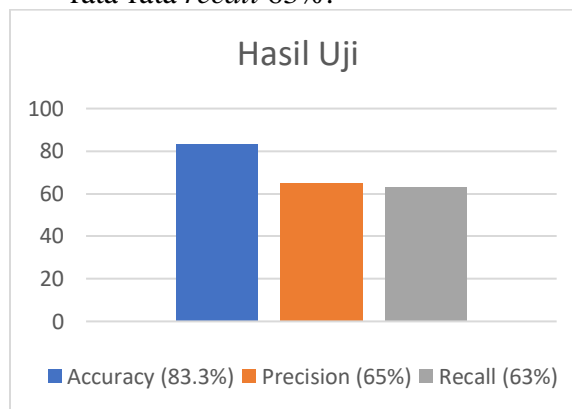
Hasil penelitian diukur dengan *confusion matrix*, berdasarkan pada gambar 2, sistem berhasil mengklasifikasi 162 komentar positif dengan benar, 1 komentar netral dengan benar, dan 132 komentar negatif dengan benar.

Total klasifikasi yang benar dilakukan 295 dari 354 komentar.



Gambar 2. Hasil *Confusion Matrix*

Accuracy, *precision*, dan *recall* dari hasil prediksi dapat dihitung dengan data yang berasal dari *confusion matrix*. Hasil Uji yang terdapat pada gambar 2 menunjukkan *accuracy* sebesar 83.3%, rata-rata *precision* 65%, dan rata-rata *recall* 63%.



Gambar 3. Hasil uji sistem

4. KESIMPULAN

Berdasarkan hasil dari pengujian pada sistem, maka dapat diambil kesimpulan sebagai berikut :

1. Algoritma NBC dapat melakukan klasifikasi dengan baik untuk memberikan label positif, negatif, dan netral dengan akurasi 83.3%.

2. Sistem kurang akurat dalam memberikan klasifikasi netral karena dari keseluruhan data didominasi oleh data dengan klasifikasi positif dan negatif.

5. SARAN

Berdasarkan kesimpulan di atas maka dapat dikemukakan beberapa saran yang diharapkan dapat menjadi masukan bagi kemajuan sistem yang akan datang. Beberapa saran dari penulis yaitu :

1. Penelitian selanjutnya dapat dilakukan dengan dataset yang lebih besar dan dengan variasi komentar yang jumlahnya memiliki perbandingan yang tidak terlalu jauh.

2. Pada tahap *preprocessing* dapat mengubah kata kata yang salah penulisannya menjadi kata yang baku.

DAFTAR PUSTAKA

- [1] M. Yusi Kamhar and E. Lestari, "Pemanfaat Sosial Media Youtube Sebagai Media Pembelajaran Bahasa Indonesia DI Perguruan Tinggi," *Intel. J. Ilmu Pendidik.*, vol. 1, no. 2, pp. 1–7, 2019, doi: 10.33366/ilg.v1i2.1356.
- [2] A. I. Almuttaqi, "Kekacauan Respons terhadap COVID-19 di Indonesia," *Habibie Cent. Insights*, vol. 1, no. 13, pp. 1–7, 2020, [Online]. Available: <http://habibiecenter.or.id/img/publication/66f28c42de71fefe1c6fcdee37a5c1a6.pdf>.
- [3] M. Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J. & Nithya, "Preprocessing Techniques for Text Mining Preprocessing Techniques for Text Mining," *Int. J. Comput. Sci. Commun. Networks*, vol. 5, no. October 2014, pp. 7–16, 2015.
- [4] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, "Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach," 2018, [Online]. Available: <http://arxiv.org/abs/1809.08651>.
- [5] A. Deolika, K. Kusriani, and E. T. Luthfi, "Analisis Pembobotan Kata Pada Klasifikasi Text Mining," *J. Teknol. Inf.*, vol. 3, no. 2, p. 179, 2019, doi: 10.36294/jurti.v3i2.1077.
- [6] B. Das and S. Chakraborty, "An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation," 2018, [Online]. Available: <http://arxiv.org/abs/1806.06407>.
- [7] H. Annur, "Klasifikasi Masyarakat Miskin Menggunakan Metode Naive Bayes," *Ilk. J. Ilm.*, vol. 10, no. 2, pp. 160–165, 2018, doi: 10.33096/ilkom.v10i2.303.160-165.
- [8] M. Zarlis and E. Budhiarti Nababan, "Analisis Perbandingan Akurasi dalam Identifikasi Autism dengan SVM dan Naive Bayes," *Jl. Thamrin No*, vol. 17, no. 2, pp. 4567789–4567790, 2016, [Online]. Available: <https://mikroskil.ac.id/ejurnal/index.php/jsm/article/view/384>.
- [9] R. Hermawan, "KLASIFIKASI SENTIMENT ANALYSIS PSBB DI INDONESIA PADA TWITTER MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE

DENGAN FEATURE
SELECTION,” 2020.

- [10] D. Normawati and S. A. Prayogi,
“Implementasi Naïve Bayes Classifier
Dan Confusion Matrix Pada Analisis
Sentimen Berbasis Teks Pada
Twitter,” *J. Sains Komput. Inform.*,
vol. 5, no. 2, pp. 697–711, 2021.