

Text Mining of PeduliLindungi Application Reviews on Google Play Store

Irwansyah Saputra¹, Taufik Djatna², Riki Ruli A. Siregar³,
Dinar Ajeng Kristiyanti⁴, Hasbi Rahma Yani⁵, Andri Agung Riyadi⁶

¹Fakultas Teknologi Informasi Universitas Nusa Mandiri, Indonesia

^{1,2,3,4}Departemen Ilmu Komputer FMIPA IPB University Bogor, Indonesia

⁴Fakultas Teknik dan Informatika Universitas Multimedia Nusantara, Indonesia

⁵Fakultas Adab dan Humaniora UIN Imam Bonjol Padang, Indonesia

⁶Fakultas Teknologi Informasi Universitas Nusa Mandiri, Indonesia

Article Info

Article history:

Received Sep 9, 2021

Revised Jun 02, 2022

Accepted Jun 18, 2022

Keywords:

Data Mining

Sentiment Analysis

Algorithm

Naive Bayes Classifier (NBC)

PeduliLindungi

ABSTRACT

The PeduliLindungi application is an application made by the Indonesian government to track and stop the spread of Covid-19. Reviews related to the application are not all good, this is evidenced by the variety of star ratings given by users so that it is difficult in seeing positive or negative sentiment related to the application. This study aims to classify reviews about the application of PeduliLindungi to two cutlasses, namely positive sentiment, and negative sentiment. The classification algorithm used is the Naive Bayes Classifier (NBC) classification. Results show Accuracy 85%, Precision 77,7%, Recall 98%, dan F1-Score 86,7%.

Copyright © 2022 Universitas Indraprasta PGRI.
All rights reserved.

Corresponding Author:

Irwansyah Saputra,

Fakultas Teknologi Informasi

Universitas Nusa Mandiri

Jl. Jatiwaringin No. 2, Cipinang Melayu, Makasar Jakarta Timur

Email: irwansyah.iys@nusamandiri.ac.id

1. PENDAHULUAN

Aplikasi PeduliLindungi merupakan suatu aplikasi yang diciptakan oleh pemerintah Indonesia dalam melakukan pelacakan untuk menghentikan penyebaran Covid-19 [1]. Aplikasi ini dapat diunduh pada *Google Play Store* dan sudah mendapatkan lebih dari 38.000 ulasan dari para penggunanya [2]. Berbagai ulasan yang bersifat teks tersebut dapat diproses menggunakan *data mining* agar didapatkan ekstraksi pengetahuan [3]. Data yang datang dari ulasan di internet memiliki sifat tidak terstruktur, artinya banyak tantangan dan potensi bias dalam data [4]. Namun demikian, terdapat berbagai penelitian yang sudah menggunakan data dari media sosial seperti deteksi serangan dan *cyberbullying* di Twitter [5], melacak penyebaran influenza [6], memantau dan memberikan dukungan selama krisis kemanusiaan [7], deteksi kesehatan pribadi di media sosial [8], reaksi publik terhadap vaksinasi [9] dan memahami ide bunuh diri [10].

Aplikasi PeduliLindungi merupakan aplikasi yang muncul setelah terjadinya pandemi *Covid-19*. *Coronavirus* merupakan suatu kelompok virus yang dapat menyebabkan penyakit pada hewan atau manusia. Beberapa jenis *coronavirus* diketahui menyebabkan infeksi saluran nafas pada manusia mulai dari batuk, pilek hingga yang lebih serius seperti *Middle East Respiratory Syndrome* (MERS) dan *Severe Acute Respiratory Syndrome* (SARS). *Coronavirus* jenis baru yang ditemukan menyebabkan penyakit COVID-19 [11]. *Covid-19* menyebar dengan sangat cepat. Data WHO menunjukkan hingga bulan November 2020 terdapat lebih dari 63

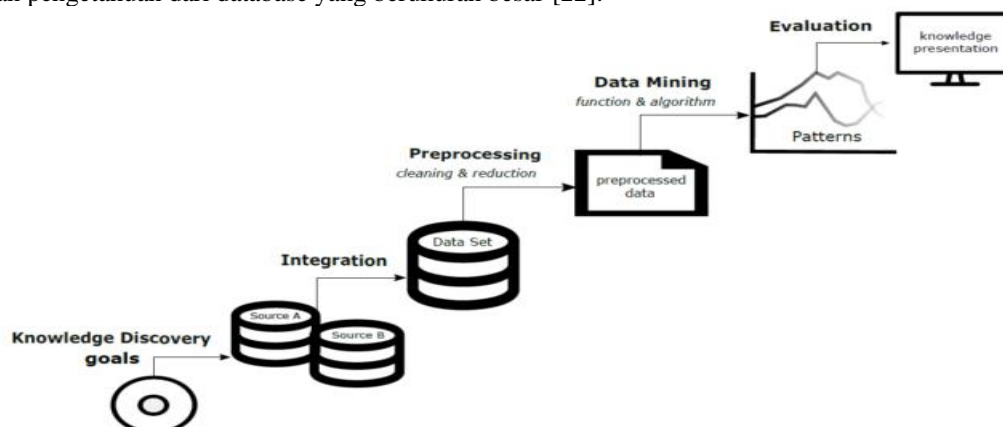
juta kasus terkonfirmasi *Covid-19* dari seluruh dunia, lebih dari 1 juta kasus kematian terkait virus tersebut [12]. Khususnya di Asia Tenggara, Indonesia merupakan negara terparah *Covid-19*, memiliki lebih dari 500 ribu kasus terkonfirmasi, terdiri dari 17.081 kasus meninggal dan 458.880 kasus sembuh [13]. Deteksi awal terhadap orang yang dicurigai terdampak *Covid-19* akan mencegah penyebarannya semakin luas, karena kontak orang tersebut dengan orang lain yang sehat menjadi terbatas [14].

Salah satu cara untuk mencegah dan mengatasi *Covid-19* adalah dengan melacaknya. Aplikasi PeduliLindungi dapat digunakan untuk melakukan tugas itu. Namun, ulasan yang terdapat pada aplikasi tersebut tidak sepenuhnya baik, terlihat dari peringkat bintang yang diberikan oleh para pengguna. Banyaknya data ulasan menyulitkan untuk membedakan antara sentimen yang bersifat positif dan negatif [15][16][17] sehingga perlu dilakukan proses analisis terhadap ulasan pada aplikasi tersebut [18].

Pada penelitian ini, pendekatan metode *data mining* dilakukan untuk mengekstraksi pengetahuan dan mencari pola dari dataset yang sudah disiapkan menggunakan teknik tertentu [19]. Teknik pembelajaran mesin yang digunakan adalah algoritma klasifikasi *Naive Bayes Classifier* (NBC). Algoritma NBC merupakan salah satu algoritma yang masuk ke dalam Top 10 terbaik versi Konferensi Internasional IEEE tentang data mining pada tahun 2006 [20]. Laporan tersebut diperkuat oleh berbagai penelitian tentang teks yang menggunakan algoritma NBC seperti penelitian yang dilakukan oleh Haiyi Zhang dan Di Li yang menyatakan bahwa algoritma ini sangat sederhana namun menunjukkan kelebihan dalam klasifikasi teks. Selain itu, algoritma ini juga memberikan hasil yang memuaskan pada tugas klasifikasi email spam [21]. Tugas klasifikasi dilakukan untuk menentukan sentimen opini positif dan negatif dari para pengguna aplikasi PeduliLindungi. Setelah hasil didapatkan, kemudian akan dievaluasi akurasi, presisi dan recall dari pemodelan tersebut.

2. METODE

Metodologi penelitian yang digunakan pada penelitian ini adalah *Knowledge Discovery on Database*. KDD adalah metode yang diusulkan oleh Fayyad pada tahun 1996, merupakan proses mengekstraksi informasi baru dan pengetahuan dari database yang berukuran besar [22].



Gambar 2. Alur Proses Model KDD [22]

3.1. Knowledge Discovery Goals

Tahapan pertama dalam mengerjakan kegiatan data mining adalah menentukan tujuan berdasarkan permasalahan yang dimiliki. Penelitian ini bertujuan untuk menganalisis sentimen ke dalam dua kelas yaitu positif dan negatif. Proses pemodelan dilakukan menggunakan algoritma *Naive Bayes Classifier*.

3.2. Data Integration

Dataset yang digunakan pada penelitian ini memiliki jenis teks ulasan yang didapatkan dari hasil scrape pada aplikasi PeduliLindungi di *Google Play Store*. Dataset terdiri dari 496 data dengan 248 data positif dan 248 data negatif. Pelabelan dilakukan berdasarkan peringkat bintang pada ulasan, yang kemudian diubah menjadi kategori positif dan negatif.

3.3. Data Preprocessing

Sebelum dilakukan proses pemodelan, dataset dibersihkan terlebih dahulu dengan teknik *praprocessing*. Tahapan yang dilakukan adalah *case folding*, *tokenization*, *stopword*, dan *stemming* menggunakan *sastrawi*. Setelah selesai, selanjutnya dilakukan proses pembobotan dengan TF-IDF untuk melihat bobot dari masing-masing kata yang terdapat di dalam dataset.

3.3.1. Case Folding

Case folding merupakan teknik yang digunakan untuk menghilangkan tanda baca dan mengubah teks ke dalam bentuk *lowercase*. Sehingga yang diterima adalah huruf sesuai dengan abjad dimulai dari a sampai z, selainnya dianggap sebagai *delimiter*. Pada *Python*, prosesnya dapat dilihat pada Gambar 3.

```
In [ ]: # proses case folding
import re
def casefolding(comment):
    comment = comment.lower()
    comment = comment.strip(" ")
    comment = re.sub(r'[?${}!@_:"(){}+.,]', '', comment)
    return comment
data['comment'] = data['comment'].apply(casefolding)
data.head()
```

```
Out[34]: <bound method NDFrame.head of
0      smoga bermanfaat tapi kenapa otp not validbegi...  POSITIF
1  menginstal aplikasi ini sama pentingnya dengan...  POSITIF
2                                problem otp salah sudah  POSITIF
3                                niat baik harus diapresiasi  POSITIF
4      saya tidak menerima otp padahal nomor saya uda...  POSITIF
..
491                                ngebug mulu nga fungsi  NEGATIF
492                                force close setelah sign up  NEGATIF
493                                nge lag need to fix the bugs ya  NEGATIF
494  apa gunanya kalo setiap aplikasinya di buka ma...  NEGATIF
495  ko force close terus sih udh uninstallinstall ...  NEGATIF

[496 rows x 2 columns]>
```

Gambar 3. *Case Folding* dan Hasilnya Menggunakan *Python*

3.3.2. Tokenization

Tahapan ini merupakan proses pemotongan *string* masukan berdasarkan tiap kata yang menyusunnya. Proses *tokenization* menggunakan *Python* dapat dilihat pada Gambar 4.

```
In [ ]: # proses tokenizing
def token(comments):
    nstr = comments.split(' ')
    dat = []
    a = -1
    for hu in nstr:
        a = a + 1
        if hu == '':
            dat.append(a)
    p = 0
    b = 0
    for q in dat:
        b = q - p
        del nstr[b]
        p = p + 1
    return nstr
data['comment'] = data['comment'].apply(token)
data.head()
```

```
Out[35]:
comment  VALUE
0  [smoga, bermanfaat, tapi, kenapa, otp, not, va...  POSITIF
1  [menginstal, aplikasi, ini, sama, pentingnya, ...  POSITIF
2  [problem, otp, salah, sudah, benar]  POSITIF
3  [niat, baik, harus, diapresiasi]  POSITIF
4  [saya, tidak, menerima, otp, padahal, nomor, s...  POSITIF
```

Gambar 4. *Tokenization* dan Hasilnya Menggunakan *Python*

3.3.3. Stopword

Kata yang dianggap tidak penting akan dihilangkan dalam proses ini. Tahapan ini termasuk ke dalam proses *filtering*. Terdapat dua metode yang dapat digunakan yaitu *stoplist* untuk membuang kata kurang penting, atau *wordlist* untuk menyimpan kata yang penting. Dalam penelitian ini, metode yang digunakan adalah *stoplist/stopword*. Contoh *stopword* adalah “yang”, “di”, “dan”, “dari” dan lain sebagainya. Proses *stopword* menggunakan *Python* dapat dilihat pada Gambar 5.

```
In [ ]: # proses filtering
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

def stopword_removal(comments):
    filtering = stopwords.words('indonesian')
    x = []
    data = []
    def myFunc(x):
        if x in filtering:
            return False
        else:
            return True
    fit = filter(myFunc, comments)
    for x in fit:
        data.append(x)
    return data
data['comment'] = data['comment'].apply(stopword_removal)
data.head(10)

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

Out[36]:
```

	comment	VALUE
0	[smoga, bermanfaat, otp, not, valid, begitu, dk...	POSITIF
1	[menginstal, aplikasi, mencuci, tangan, stay, ...	POSITIF
2	[problem, otp, salah, beres]	POSITIF
3	[niat, apresiasi]	POSITIF
4	[menerima, otp, nomor, udah, solusi]	POSITIF
5	[i, hope, this, app, will, work, as, it, is]	POSITIF
6	[stuck, otp, berulang, kali, kirim, otp, diterm...	POSITIF
7	[hebat]	POSITIF
8	[ayo, dukung, program, pemerintah, 1]	POSITIF
9	[install, ga, fiturnya, ditambah, odp, /pdp, r...	POSITIF

Gambar 5. *Stopword* dan Hasilnya Menggunakan *Python*

3.3.4. *Stemming*

Tahapan ini merupakan proses mengembalikan suatu kata kepada bentuk dasarnya. Jika dalam Bahasa Indonesia, proses *stemming* dapat digunakan untuk menghilangkan imbuhan yang menyertai kata dasarnya. *Stemming* Bahasa Indonesia yang digunakan sebagai acuan adalah *Library Sastrawi* yang disiapkan untuk Bahasa Pemrograman *Python* [23]. Seluruh proses *stemming* yang terjadi sesuai dengan ketentuan dari *Library Sastrawi*, maka luaran yang didapatkan sudah sesuai dengan algoritma pada library tersebut. Selain itu, jika terdapat kalimat tidak baku pada dataset maka ia takkan berubah. Proses *stemming* pada *Python* dapat dilihat pada Gambar 6.

```
In [ ]: # proses stemming
!pip install Sastrawi
from sklearn.pipeline import Pipeline
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

def stemming(comments):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    do = []
    for w in comments:
        dt = stemmer.stem(w)
        do.append(dt)
    d_clean = []
    d_clean = " ".join(do)
    print(d_clean)
    return d_clean
data['comment'] = data['comment'].apply(stemming)

data.to_csv('data_clean.csv', index=False)
data_clean = pd.read_csv('data_clean.csv', encoding='latin1')
data_clean.head()
```

```
Requirement already satisfied: Sastrawi in /usr/local/lib/python3.6/dist-packages (1.0.1)
smoga manfaat otp not valid begitu kirim sbg partisipan
menginstal aplikasi cuci tangan stay at home sah physical distancing cepat putus rantai sebar c-19 salam sehat bahagia tuhan
yesus kati serta amin
problem otp salah beres
niat apresiasi
terima otp nomor udah solusi
i hope this app will work as it is
stuck otp berulang kali kirim otp diterimacoba uninstall verifikasi otp
hebat
ayo dukung program perintah 1
install ga fiturnya tambah odp pdp rumah alarm bunyi rumah tahu odp pdp jarak 10meter
good initiative
good job
orang goblok yg otak nya serang corona bilang aplikasi ga sesuai harap real-time mapping data rs rujuk lha untuk nya ituso st
upid
semangat developer nya saran tambah peta episentrum covid19 cegah obat semangat developer nya
apk is the best
terima kasih upaya cegah sebar covid-19 masalah otp nya tambah 0 awal nomor hp 08111111 tulis 8111111 aja hasil
```

Gambar 6. *Stemming* dan Hasilnya Menggunakan *Python*

3.3.5. Pembobotan Kata dengan TF-IDF

TF-IDF merupakan teknik pembobotan yang populer dalam *text mining*. Terdapat dua proses yang berjalan dalam tahapan ini yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). *Term Frequency* (TF) merupakan frekuensi kemunculan sebuah term dalam dokumen yang terkait. Semakin tinggi jumlah kemunculan term dalam dokumen, semakin tinggi pula bobotnya. *Inverse Document Frequency* (IDF) merupakan sebuah perhitungan yang menunjukkan hubungan ketersediaan sebuah *term* dalam seluruh dokumen. Semakin sedikit jumlah dokumen yang mengandung suatu *term* tertentu, maka nilai IDF akan menjadi semakin besar [24].

$$tf - idf_{t,d} = tf_d * idf_t \tag{1}$$

Selanjutnya, proses pembobotan TF-IDF yang dilakukan pada penelitian ini dapat dilihat pada Gambar 7.

```
In [ ]: # proses TF-IDF
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer

tvec = CountVectorizer()
X_cVec = tvec.fit_transform(data_clean['comment'].values.astype('U'))
print(X_cVec)
h_tfidf = TfidfTransformer()
x_tfidf = h_tfidf.fit_transform(X_cVec)
print(x_tfidf)
X = data_clean.comment
Y = data_clean.VALUE

(0, 1017) 1
(0, 656) 1
(0, 815) 1
(0, 761) 1
(0, 1201) 1
(0, 575) 1
(0, 963) 1
(0, 834) 1
(1, 685) 1
(1, 86) 1
(1, 254) 1
(1, 1072) 1
(1, 1038) 1
```

Gambar 7. Pembobotan TF-IDF Menggunakan Python

3.4. Data Mining

Tahapan ini merupakan proses inti dari penelitian. Setelah dataset dibersihkan, kemudian diolah menggunakan teknik *data mining*. Sebelum proses pemodelan, dataset dipartisi menjadi dua bagian, yaitu 80% untuk menjadi data latih dan 20% menjadi data uji. Algoritma yang digunakan adalah *Naive Bayes Classifier*. Proses ini dapat dilihat pada Gambar 8.

```
In [10]: # performa algoritma naive bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

clf = MultinomialNB().fit(X_train, y_train)
predicted = clf.predict(X_test)
print("MultinomialNB Accuracy: ", accuracy_score(y_test,predicted))
print("MultinomialNB Precision: ", precision_score(y_test,predicted, average="binary", pos_label="NEGATIF" ))
print("MultinomialNB Recall: ", recall_score(y_test,predicted, average="binary", pos_label="NEGATIF"))
print("MultinomialNB f1_score: ", f1_score(y_test,predicted, average="binary", pos_label="NEGATIF"))

print(f'confusion matrix:\n {confusion_matrix(y_test, predicted)}')
print('-----\n')
print(classification_report(y_test, predicted, zero_division=0))

MultinomialNB Accuracy: 0.85
MultinomialNB Precision: 0.7777777777777778
MultinomialNB Recall: 0.98
MultinomialNB f1_score: 0.8672566371681417
confusion matrix:
[[49  1]
 [14 36]]

-----
              precision    recall  f1-score   support

  NEGATIF      0.78      0.98      0.87         50
  POSITIF      0.97      0.72      0.83         50

 accuracy          0.88
 macro avg         0.88      0.85      0.85         100
 weighted avg     0.88      0.85      0.85         100
```

Gambar 8. Algoritma Naive Bayes Menggunakan Python

3.5. Evaluation

Evaluasi dilakukan untuk melihat hasil performansi dari model yang sudah dibuat. Parameter performa yang akan digunakan adalah *Accuracy*, *Precision*, *Recall* dan *F1-Score* [25]. *Confusion matrix* menghasilkan nilai yang dapat dilihat pada Table I.

TABLE I. CONFUSION MATRIX

	Actual Positive	Actual Negative
Predic. Positive	TP	FP
Predic. Negative	FN	TN

TP adalah *True Positive*, yaitu hasil prediksi positif sesuai dengan kenyataan positif. FP adalah *False Positive*, yaitu hasil prediksi positif tidak sesuai dengan kenyataannya negatif. FN adalah *False Negative*, yaitu hasil prediksi negatif tidak sesuai dengan kenyataannya positif. Terakhir, TN adalah *True Negative*, yaitu hasil prediksi negatif sesuai dengan kenyataan negatif.

Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Maka, *accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Dengan kata lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya).

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \quad (2)$$

Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari semua kelas positif yang telah di prediksi dengan benar, berapa banyak data yang benar-benar positif.

$$Precision = \frac{TP}{FP+TP} \times 100\% \quad (3)$$

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (4)$$

F1 Score merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan.

$$F1 = 2 * (Recall * Precision) / (Recall + Precision) \quad (5)$$

3. HASIL DAN PEMBAHASAN

Bagian ini akan menjelaskan hasil yang didapatkan dan menganalisis performansinya. Hasil yang akan dievaluasi adalah Akurasi, *Precision*, *Recall* dan *F1-Score*.

TABLE II. EVALUASI HASIL PEMODELAN NAÏVE BAYES

<i>Naïve Bayes Classifier</i>			
<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
85%	77,7%	98%	86,7%

Berdasarkan hasil yang didapatkan, pada Tabel II nilai akurasi algoritma Naïve Bayes adalah 85,00%. Hasil presisi dari algoritma Naïve Bayes adalah 77,70%. Hasil recall algoritma Naïve Bayes sebesar 98,00%. Kemudian hasil Nilai F-1 sebesar 86,70%. Secara keseluruhan, berdasarkan hasil di atas, kinerja algoritma Naïve Bayes dalam mengklasifikasikan ulasan mengenai aplikasi PeduliLindungi menunjukkan klasifikasi yang sangat baik.

Hal yang menarik dari penelitian ini adalah baik algoritma Naïve Bayes mampu memprediksi data tweet positif dengan nilai 98% true positive. Bahkan jika dilihat berdasarkan data, ada ketidakseimbangan antara data yang berlabel positif dan negatif. Hal ini biasa terjadi karena proses pengumpulan data menggunakan teknik data crawling di Twitter tentunya akan menampilkan data apa adanya yang sesuai dengan kata kunci yang digunakan dan realita pada dataset. Faktanya, lebih banyak opini publik yang setuju dengan penerapan aplikasi PeduliLindungi di masa wabah Covid-19 dibandingkan dengan yang tidak setuju dengan penerapan aplikasi PeduliLindungi. Pada penelitian selanjutnya algoritma *Naïve Bayes* dengan algoritma klasifikasi lainnya sehingga dapat diketahui perbandingan kinerja dari algoritma-algoritma tersebut dan mana yang lebih baik pada proses klasifikasi teks dalam kasus sentimen analisis seperti pada pekerjaan ini.

4. PENUTUP

Penelitian ini dilakukan untuk mengklasifikasikan sentimen analisis dari *review* para pengguna Aplikasi pencegahan dan pelacakan penyebaran Covid 19: PeduliLindungi. Ulasan dari pengguna dirangkum dengan metode data mining dan prediksi label pada dataset menggunakan algoritma *Naïve Bayes Classifier* (NBC). Hasil Menunjukkan *Accuracy* 85%, *Precision* 77,7%, *Recall* 98%, dan *F1-Score* 86,7%.

Penelitian selanjutnya, untuk dapat mengetahui perbandingan kinerja dari berbagai algoritma, dilakukan analisis tidak hanya dengan algoritma NBC saja, namun juga dilakukan dengan algoritma lain seperti

Decision Tree dan K-NN. sehingga melalui proses tersebut dapat ditemukan algoritma mana yang lebih baik pada proses klasifikasi teks dalam kasus sentimen analisis.

UCAPAN TERIMAKASIH

Terima kasih kepada para dosen dari IPB University yaitu Professor Imas Sukaesih Sitanggang, Professor Taufik Djatna, dan Dr. Annisa yang telah membimbing sehingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] PeduliLindungi, "Apa itu PeduliLindungi?," 2020. <https://pedulilindungi.id/>.
- [2] K. Kominfo, "PeduliLindungi," 2020. <https://play.google.com/store/apps/details?id=com.telkom.tracencare&hl=en>.
- [3] Y. Jiao and Q. X. Qu, "A proposal for Kansei knowledge extraction method based on natural language processing technology and online product reviews," *Comput. Ind.*, vol. 108, pp. 1–11, 2019, doi: 10.1016/j.compind.2019.02.011.
- [4] D. Nguyen, A. S. Dogruoz, C. P. Rose, and F. de Jong, "Computational Sociolinguistics: A Survey," *Comput. Linguist.*, vol. 42, no. 3, pp. 537–593, 2016, doi: 10.1162/COLI_a_00258.
- [5] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali, "Mean birds: Detecting aggression and bullying on Twitter," in *WebSci 2017 - Proceedings of the 2017 ACM Web Science Conference*, 2017, pp. 13–22, doi: 10.1145/3091478.3091487.
- [6] E. Aramaki, S. Maskawa, and M. Morita, "Twitter catches the flu: Detecting influenza epidemics using Twitter," in *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2011, vol. 2011, no. January 2011, pp. 1568–1576, [Online]. Available: <http://www.aclweb.org/anthology/D11-1145>.
- [7] M. Imran, P. Mitra, and C. Castillo, "Twitter as a lifeline: Human-annotated Twitter corpora for NLP of crisis-related messages," in *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2016, no. May, pp. 1638–1643.
- [8] P. Karisani and E. Agichtein, "Did you really just have a heart attack?: Towards robust detection of personal health mentions in social media," in *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018*, 2018, pp. 137–146, doi: 10.1145/3178876.3186055.
- [9] M. Salathé and S. Khandelwal, "Assessing vaccination sentiments with online social media: Implications for infectious disease dynamics and control," *PLoS Comput. Biol.*, vol. 7, no. 10, 2011, doi: 10.1371/journal.pcbi.1002199.
- [10] M. De Choudhury, E. Kiciman, M. Dredze, G. Coppersmith, and M. Kumar, "Discovering shifts to suicidal ideation from mental health content in social media," in *Conference on Human Factors in Computing Systems - Proceedings*, 2016, pp. 2098–2110, doi: 10.1145/2858036.2858207.
- [11] WHO, "Pertanyaan dan jawaban terkait Coronavirus." <https://www.who.int/indonesia/news/novel-coronavirus/qa-for-public>.
- [12] WHO, "WHO Coronavirus Disease (COVID-19) Dashboard," 2020. <https://covid19.who.int/>.
- [13] WHO, "Dashboard COVID-19 Indonesia," 2020. <https://covid19.who.int/region/searo/country/id>.
- [14] CDC, "Social Distancing," 2020. <https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html>.
- [15] I. Saputra, J. A. Halomoan, A. B. Raharjo, and C. R. A. Syavira, "Sentiment Analysis on Twitter of Psbb Effect Using Machine Learning," *J. Techno Nusa Mandiri*, vol. 17, no. 2, pp. 143–150, 2020, doi: 10.33480/techno.v17i2.1635.
- [16] A. A. Permana, M. F. Fahrezi, D. Y. Priyanggodo, A. Kristiyanti, and M. Sihotang, "Sentimen Analisis Opini Masyarakat Pada Media Sosial Twitter Terhadap Vaksin Berbayar Menggunakan Metode Naïve Bayes Classifier (NBC)," *JTS J. Tek.*, vol. 10, no. 2, pp. 84–92, 2021.
- [17] A. A. Permana, arsanah arsanah, D. A. Kristiyanti, and M. Sihotang, "Analisis Sentimen Pendapat Masyarakat Terhadap Ppkm Darurat Pada Media Sosial Twitter Menggunakan Metode Naïve Bayes," *JIKA (Jurnal Inform.*, vol. 5, no. 3, pp. 400–404, 2021, [Online]. Available: <http://jurnal.umt.ac.id/index.php/jika/article/view/5190>.
- [18] E. Indrayuni, A. Nurhadi, and D. A. Kristiyanti, "Implementasi Algoritma Naive Bayes, Support Vector Machine, dan K-Nearest Neighbors untuk Analisa Sentimen Aplikasi Halodoc," *Fakt. Exacta*, vol. 14, no. 2, p. 64, 2021, doi: 10.30998/faktorexacta.v14i2.9697.

- [19] A. Muzakir and R. A. Wulandari, "Model Data Mining sebagai Prediksi Penyakit Hipertensi Kehamilan dengan Teknik Decision Tree," *Sci. J. Informatics*, vol. 3, no. 1, pp. 19–26, 2016, doi: 10.15294/sji.v3i1.4610.
- [20] X. Wu *et al.*, *Top 10 algorithms in data mining*, vol. 14, no. 1. 2008.
- [21] X. F. Zhang, H. Y. Huang, and K. L. Zhang, "KNN text categorization algorithm based on semantic centre," in *Proceedings - 2009 International Conference on Information Technology and Computer Science, ITCS 2009*, 2009, vol. 1, pp. 249–252, doi: 10.1109/ITCS.2009.57.
- [22] G. William and Z. Huang, *Modelling the KDD Process*, no. June. CSIRO Div. Inf. Technol, 1996.
- [23] A. Librian, "Sastrawi," 2017. .
- [24] R. Rasenda, H. Lubis, and R. Ridwan, "Implementasi K-NN Dalam Analisa Sentimen Riba Pada Bunga Bank Berdasarkan Data Twitter," *J. Media Inform. Budidarma*, vol. 4, no. 2, p. 369, 2020, doi: 10.30865/mib.v4i2.2051.
- [25] R. Feldman and J. Sanger, *The Text Mining HandBook*. 2007.