

ANALISA SENTIMEN APLIKASI PEDULILINDUNGI DENGAN METODE NBC DAN SVM

Farras Naufal Majid¹, Sulastri²

¹Sistem Informasi – Universitas Stikubank Semarang, fnmajid@gmail.com

²Sistem Informasi – Universitas Stikubank Semarang, sulastri@edu.unisbank.ac.id

Co-author sulastri@edu.unisbank.ac.id

ARTICLE INFO

Article history:

Received Desember 28 2022

Received in revised form Januari 8 2023

Accepted Januari 14 2023

Available online Juli 16 2023

ABSTRACT

PeduliLindungi is an application from the Government of Indonesia that was made in response to the COVID-19 pandemic. Since its initial release in 2020, this application has received many updates with the goal of improving its overall performance. One of the basics of updating applications is to process the reviews given by users at the Google Play Store using sentiment analysis. The methods used this time are Naive Bayes Classifier (NBC) and Support Vector Machine (SVM). The sample data used were 300 reviews with positive feedback and 300 reviews with negative feedback, for a total of 600 user reviews. The results of the NBC algorithm calculations produce an accuracy of 76%, a precision of 76%, a recall of 82%, and an f1-score of 79%. As for the SVM algorithm, it produces an accuracy rate of 80%, a precision of 83%, a recall of 80%, and an f1-score of 81%.

Keywords: PeduliLindungi, Sentiment, NBC, SVM

1. PENDAHULUAN

Pandemi Covid-19 yang melanda dunia sejak tahun 2020 telah memberikan dampak yang luar biasa bagi Indonesia. Pemerintah melakukan beberapa peraturan serta kebijakan baru untuk menanggulangi dan mencegah peredaran Covid-19 di Indonesia, salah satunya adalah dengan meluncurkan aplikasi PeduliLindungi. Aplikasi PeduliLindungi ini dibuat dengan tujuan untuk mencegah dan mengurangi penyebaran virus Covid-19. Sejak pertama kali diluncurkan pada tahun 2020 silam, aplikasi ini telah mengalami beberapa kali pembaharuan guna meningkatkan performa dan kinerja pada aplikasi tersebut [1].

Dalam melakukan pembaharuan suatu sistem aplikasi, perlu dilakukan analisa dan penelitian untuk mengetahui kekurangan atau *error code* yang ada agar pembaharuan dinilai positif dan memiliki peningkatan secara umum. Salah satu acuan mengetahui kebutuhan maupun kekurangan untuk meningkatkan performa aplikasi *mobile apps* adalah dari *feedback user* yang meliputi *rating score* dan *sentiment review* pada Google Play Store. Berbagai pendapat yang ditulis oleh pengguna pada kolom *review* Google Play Store ini merupakan *big data* yang bisa berupa tanggapan positif, negatif, maupun netral.

Banyaknya data yang dihasilkan dari *sentiment review* tersebut, dapat dilakukan *text mining* untuk klasifikasi sentimen positif dan negatif. Analisa sentimen merupakan suatu proses pengelolaan data yang berupa teks untuk memahami informasi tentang respon atau opini. *Text mining* merupakan suatu metode penambangan data turunan dari *data mining* yang berfokus pada pengelolaan teks untuk menemukan suatu pola dan informasi sesuai kebutuhan [2].

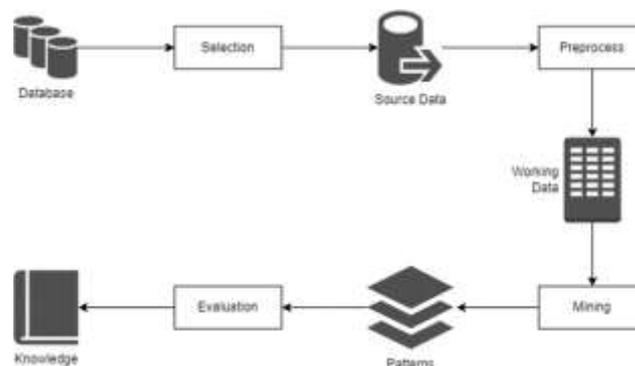
Seperti pada penelitian Irwansyah Saputra, dkk [3] sebelumnya telah melakukan analisa sentimen aplikasi PeduliLindungi dengan metode *Naïve Bayes Classifier* (NBC). Penelitian tersebut menggunakan dataset dari Google Play Store dengan jumlah 496 data yang meliputi 248 data positif dan 248 data negative. Hasil analisa tersebut menghasilkan *Accuracy* 85%, *Precision* 77,7%, *Recall* 98% dan *F1-Score* 86,7%. Dari hasil penelitian tersebut, perlu dilakukan analisa kembali menggunakan algoritma klasifikasi lain.

Ada bermacam-macam algoritma klasifikasi *text mining* selain NBC, salah satunya adalah *Support Vector Machine* (SVM). Baik NBC maupun SVM merupakan algoritma klasifikasi *supervised learning*, artinya pada implementasinya perlu dilakukan pelatihan dan pengenalan suatu objek analisa kemudian dapat dilanjutkan dengan pengujian data [4]. NBC memiliki kelebihan seperti mudah digunakan serta cepat dalam melakukan pengujian data jamak. Sedangkan SVM merupakan algoritma klasifikasi dengan Teknik *kernel* sehingga mampu menyelesaikan permasalahan yang lebih kompleks. *Kernel* mampu merubah *sample input* data menjadi data yang berdimensi tinggi sehingga asumsi *sample* pada data dipisahkan secara linier bisa dihilangkan [5]. Tujuan dilakukannya penelitian ini adalah untuk mengetahui perbandingan perhitungan klasifikasi antara algoritma NBC dan SVM.

2. METODE PENELITIAN

2.1. Model Penelitian

Knowledge Discovery on Database (KDD) adalah suatu metode yang dinilai tepat digunakan dalam penelitian ini. KDD merupakan suatu proses penambangan data untuk mendapatkan informasi baru dari database yang besar. Metode ini terdiri dari 4 tahap utama yang meliputi Pengumpulan Data, *Preprocessing Data*, Penambangan Data, serta evaluasi [6]. Alur Proses *KDD Modelling* dapat dilihat pada gambar 1.



Gambar 1. KDD Modelling

2.2. Pengumpulan Data

Pengumpulan data atau *scraping data* dilakukan dengan menggunakan Google Play Scraper dari bahasa pemrograman *Phyton*. Pengambilan data dilakukan dengan cara mengambil data ulasan paling relevan dan secara spesifik dipilih berdasarkan *rating score* 1 dan 2 untuk ulasan negatif, serta *rating score* 4 dan 5 untuk ulasan positif. Label ulasan positif berjumlah 300 data yang terdiri dari 150 data dengan *rating score* 4 dan 150 data dengan *rating score* 5. Hal

yang sama juga dilakukan untuk pengambilan data dengan label ulasan negatif yang terdiri dari 300 data dengan *rating score* 1 dan 2 yang masing-masing berjumlah 150 data. Dari hasil *crawling* data tersebut, dilakukan penggabungan data yang menghasilkan total data yang diperoleh adalah 600 data.

2.3. Preprocessing Data

Tahap *preprocessing data* merupakan tahap pengolahan data dari data mentah menjadi data yang lebih bersih agar bisa digunakan untuk proses *mining data*. Tahap *preprocessing data* yang dilakukan yaitu *case folding*, *tokenizing*, *filtering*, *stemming* menggunakan Sastrawi, serta pembobotan term menggunakan TF-IDF.

a. Case Folding

Case Folding dengan maksud untuk menyamakan semua huruf dari a hingga z. Proses yang dilakukan adalah penghapusan tanda baca, serta merubah huruf menjadi *lowercase*.

Tabel 2. Proses Case Folding

<i>Sebelum</i>
Sesaat setelah berhasil login selalu muncul pesan "pemberitahuan sesi anda telah habis mohon login ulang". Jadi ini aplikasi belum difungsikan sudah tidak berfungsi karena sesi selalu habis bahkan sesaat setelah login.... Piye to min dev?? mohon dibantu... [2]
<i>Setelah</i>
sesaat setelah berhasil login selalu muncul pesan pemberitahuan sesi anda telah habis mohon login ulang jadi ini aplikasi belum difungsikan sudah tidak berfungsi karena sesi selalu habis bahkan sesaat setelah login piye to min dev mohon dibantu [2]

b. Tokenizing

Tokenizing merupakan tahapan memberi pembatas kata pada kalimat dengan tanda baca koma (,). Tahap ini dilakukan setelah proses *case folding* untuk menghapus tanda baca yang tidak diperlukan.

Tabel 2. Proses Tokenizing

<i>Sebelum</i>
sesaat setelah berhasil login selalu muncul pesan pemberitahuan sesi anda telah habis mohon login ulang jadi ini aplikasi belum difungsikan sudah tidak berfungsi karena sesi selalu habis bahkan sesaat setelah login piye to min dev mohon dibantu [2]
<i>Setelah</i>
sesaat,setelah,berhasil,login,selalu,muncul,pesan,pemberitahuan,sesi,anda,telah,habis,mohon,login,ulang,jadi,ini,aplikasi,belum,difungsikan,sudah,tidak,berfungsi,karena,sesi,selalu,habis,bahkan,sesaat,setelah,login,piye,to,min,dev,mohon,dibantu

c. Filtering

Pada tahap *filtering*, yang dilakukan adalah proses *stopword removal* yang bertujuan untuk menghilangkan kata yang tidak relevan agar memudahkan dalam analisa modeling serta efisien.

Tabel 3. Proses Filtering

<i>Sebelum</i>
sesaat,setelah,berhasil,login,selalu,muncul,pesan,pemberitahuan,sesi,anda,telah,habis,mohon,login,ulang,jadi,ini,aplikasi,belum,difungsikan,sudah,tidak,berfungsi,karena,sesi,selalu,habis,bahkan,sesaat,setelah,login,piye,to,min,dev,mohon,dibantu

ak, berfungsi, karena, sesi, selalu, habis, bahkan, sesaat, setelah, login, piye, to, min, dev, mohon, dibantu
Setelah
berhasil, login, muncul, pesan, pemberitahuan, sesi, habis, mohon, login, ulang, aplikasi, difungsikan, berfungsi, sesi, habis, login, piye, to, min, dev, mohon, dibantu

d. Stemming

Proses *Stemming* ini menggunakan database kata dari Sastrawi untuk dilakukan perubahan setiap kata menjadi akta dasar dengan tujuan mengelompokkan setiap kata yang memiliki makna yang sama.

Tabel 4. Proses Stemming

Sebelum
berhasil, login, muncul, pesan, pemberitahuan, sesi, habis, mohon, login, ulang, aplikasi, difungsikan, berfungsi, sesi, habis, login, piye, to, min, dev, mohon, dibantu
Setelah
hasil login muncul pesan pemberitahuan sesi habis mohon login ulang aplikasi fungsi fungsi sesi habis login piye to min dev mohon bantu

e. Term Frequency-Inverse Document Frequency (TF-IDF)

Setelah dilakukan tahapan *preprocessing*, maka perlu dilakukan pembobotan *term*. Salah satu metode pembobotan *term* adalah dengan *Term Frequency-Inverse Document Frequency* (TF-IDF). Metode ini menggabungkan dua konsep yaitu kemunculan kata pada setiap dokumen (TF) dan invers frekuensi dokumen yang terdapat kata tersebut (IDF) [7]. Definisi dari TF sendiri adalah jumlah frekuensi munculnya suatu *term* dalam sebuah dokumen. Sedangkan IDF adalah perhitungan terhadap hubungan ketersediaan *term* dalam keseluruhan dokumen.

Formula yang digunakan untuk menghitung bobot (W) masing-masing dokumen terhadap kata kunci adalah:

$$W_{t,d} = TF_{t,d} * IDF_t \quad (01)$$

Keterangan =

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

2.4. Pemodelan Data

a. Naïve Bayes Classifier

Naïve Bayes merupakan algoritma klasifikasi kemungkinan yang menggunakan metode probabilitas dan statistic. Metode ini memprediksi kemungkinan yang akan terjadi di masa yang akan datang dengan berdasarkan kejadian yang sudah terjadi, atau yang bisa disebut dengan *Teorama Bayes* [8]. Teorama ini dikombinasikan dengan *Naïve* yang memperkirakan seluruh atribut yang saling bebas dan lepas.

Berikut merupakan persamaan dari teorama bayes :

$$P(H|X) = \frac{P(H|X).P(H)}{P(X)} \quad (02)$$

Keterangan =

X = Data dengan class yang belum diketahui

H = Hipotesis data X merupakan suatu class spesifik

$P(H|X)$ = Probabilitas hipotesis H berdasar kondisi X (posteriori probability)

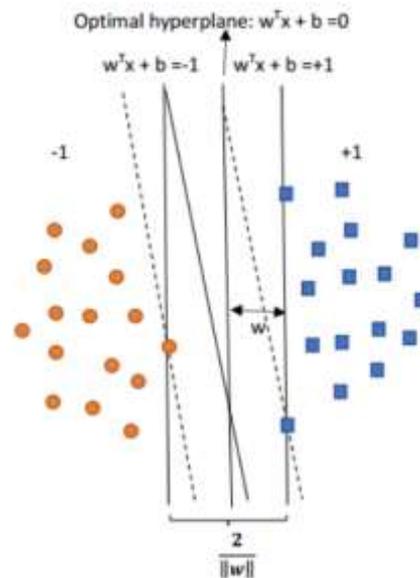
$P(H)$ = Probabilitas hipotesis H (prior probability)

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

$P(X)$ = Probabilitas X

b. Support Vector Machine

Support Vector Machine (SVM) merupakan suatu *machine learning* yang lebih muktahir dari *learning Neural Network* (NN). Prinsip dasar dari SVM adalah dengan mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua kelas data. Ilustrasi ini dapat dilihat pada gambar 2



Gambar 2. Ilustrasi Hyperplane

Pada gambar tersebut terlihat bahwa SVM mencari *hyperplane* yang optimal dengan memaksimalkan margin data yang berbeda kelas [9].

Berdasarkan hasil perhitungan pembobotan TF-IDF pencarian *hyperplane* pada implementasi SVM dapat menggunakan formula sebagai berikut [10]:

$$y_i(W_0 + W_1 X_1 + W_2 X_2 + b) \geq 0 \quad (03)$$

Keterangan:

W_1 = vektor label positif

W_2 = vektor label negatif

$X(x_1, x_2)$ = nilai atribut W

b = pembobotan dari W_0

2.5. Evaluasi Pemodelan

Pada tahap evaluasi pemodelan dapat menggunakan perhitungan performance matrix yang disebut dengan *Confusion Matrix*. *Confusion Matrix* sendiri merupakan suatu perhitungan *performance matrix* dari algoritma *supervised learning* dimana *supervised learning* merupakan metode *machine learning* yang dalam implementasinya membutuhkan *labelling* atau *class* pada datasetnya. *Class* atau label pada dataset ini yang dinamakan dengan *actual values* yaitu nilai

yang sesuai dengan data yang ada. Sedangkan *predicted values* merupakan prediksi hasil pemodelan dari suatu algoritma *machine learning* yang digunakan. Hasil *confusion matrix* pada perhitungan suatu algoritma *supervised learning* dapat dilihat pada tabel berikut.

		Actual Value	
		Positive	Negative
Predictive Value	Positive	TP	FP
	Negative	FN	TN

Keterangan =

TP merupakan *True Positive*

FP merupakan *False Positive*

FN merupakan *False Negative*

TN merupakan *True Negative*

Setelah diketahui hasil dari *confusion matrix*, maka untuk mengukur *Performance Metrics* pada suatu pemodelan algoritma dapat menggunakan *accuracy*, *precision*, *recall*, dan *F1-Score*

Accuracy = mengukur tingkat akurasi kebenaran dari hasil perhitungan suatu model klasifikasi data.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} * 100\% \quad (04)$$

Precision (Positive Predictive Value) = tingkat akurasi antara data prediksi yang diminta dengan hasil data prediksi yang diberikan oleh perhitungan model.

$$Precision = \frac{TP}{FP+TP} * 100\% \quad (05)$$

Recall atau *Sensitivity (True Positive Rate)* merupakan tingkat keberhasilan model dalam menemukan kembali informasi yang ada.

$$Recall = \frac{TP}{TP+FN} * 100\% \quad (06)$$

F1 Score = Perbandingan rata-rata presisi dan *recall* yang dibobotkan

$$F1 - Score = 2 * \frac{precision*recall}{precision+recall} \quad (07)$$

3. HASIL DAN PEMBAHASAN

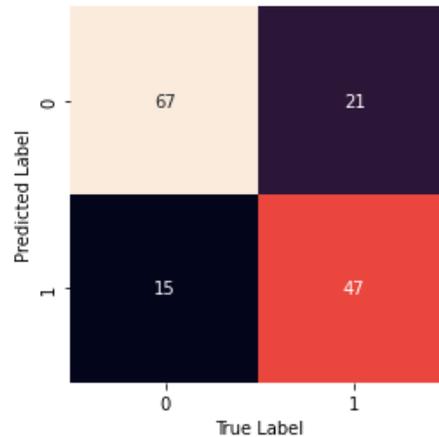
3.1. Pembagian Data (*Split Data*)

Data bersih dari hasil *preprocessing* data kemudian dilakukan split data yang terdiri dari 2 data yaitu data latih dan data uji. Untuk mengetahui hasil optimal dari perhitungan yang dilakukan, proporsi pembagian data yang dilakukan ada 3, yaitu 80% data training 20% data testing, 75%

data training 25 data testing, serta 70% data training 30% data testing. Dari perhitungan tersebut didapati hasil paling optimal adalah perhitungan klasifikasi dengan perbandingan 75% data training 25% data testing.

3.2. Klasifikasi *Naïve Bayes Classifier*

Setelah dilakukan pembagian data, maka hasil perhitungan *Naïve Bayes Classifier* (NBC) menghasilkan confusion matrix seperti pada gambar 3.



Gambar 3. Confusion Matrix NBC

Dari hasil confusion matrix tersebut, di peroleh data lain yang dapat dijadikan acuan yaitu *accuracy* sebesar 76%, *precision* sebesar 76%, *recall* 82%, dan *f1-score* 79% seperti gambar 4.

```

MultinomialNB Accuracy: 0.76
MultinomialNB Precision: 0.7613636363636364
MultinomialNB Recall: 0.8170731707317073
MultinomialNB f1_score: 0.788235294117647
confusion matrix:
[[67 15]
 [21 47]]
=====

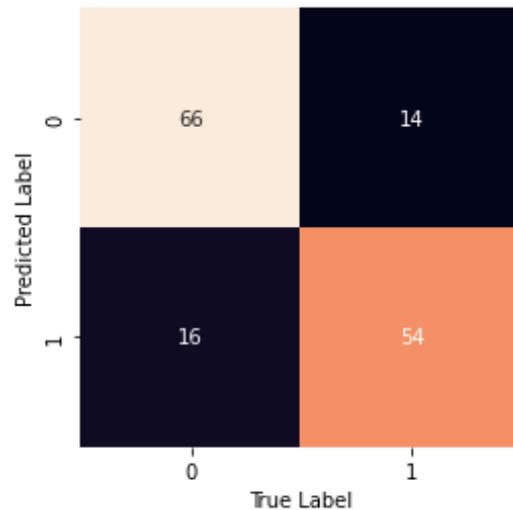
```

	precision	recall	f1-score	support
NEGATIF	0.76	0.82	0.79	82
POSITIF	0.76	0.69	0.72	68
accuracy			0.76	150
macro avg	0.76	0.75	0.76	150
weighted avg	0.76	0.76	0.76	150

Gambar 4. Hasil Klasifikasi NBC

3.3. Klasifikasi *Support Vector Machine*

Penerapan algoritma klasifikasi kedua SVM menghasilkan nilai confusion matrix seperti pada gambar 5.



Gambar 5. Confussion Matrix SVM

Dari hasil *confussion matrix* tersebut, di peroleh data lain yang dapat dijadikan acuan yaitu *accuracy* sebesar 80%, *precision* sebesar 83%, *recall* 80%, dan *f1-score* 81% seperti gambar 6.

```
Support Vector Machine Accuracy: 0.8
Support Vector Machine Precision: 0.825
Support Vector Machine Recall: 0.8048780487804879
Support Vector Machine f1_score: 0.8148148148148149
confusion matrix:
[[66 16]
 [14 54]]
=====
              precision    recall  f1-score   support

   NEGATIF      0.82      0.80      0.81         82
   POSITIF      0.77      0.79      0.78         68

 accuracy              0.80              150
 macro avg             0.80      0.80      0.80         150
 weighted avg          0.80      0.80      0.80         150
```

Gambar 6. Hasil Klasifikasi SVM

4. KESIMPULAN

Berdasarkan hasil pengujian analisa sentimen pada aplikasi PeduliLindungi dapat disimpulkan bahwa dari 600 dataset dan dengan berbagai proporsi data training dan data testing dapat disimpulkan bahwa :

1. Hasil paling optimal dari analisa menggunakan metode *Naïve Bayes* memiliki tingkat 76%, *precision* sebesar 76%, *recall* 82%, dan *f1-score* 79% dengan perbandingan proporsi 75% data training dan 25% data testing.

2. Sedangkan dengan perhitungan dengan metode *Support Vector Machine* yang mendapat hasil optimal dengan hasil *Accuracy* 80%, *Precision* 83%, *Recall* 80%, dan *F1-Score* 81% proporsi 75% data training dan 25% data testing.
3. Dengan menerapkan hasil analisa sentimen aplikasi, *developer* akan lebih mudah dalam memperbaiki kekurangan dari aplikasi tersebut.

DAFTAR PUSTAKA

- [1] Ministry of Communication and Information of the Republic of Indonesia, "Aplikasi PeduliLindungi Jadi Syarat Utama Bepergian," <https://www.kominfo.go.id/content/detail/36596/Aplikasi-Pedulilindungi-Jadi-Syarat-Utama-Bepergian/0/Artike>, 2021.
- [2] B. K. Widodo, N. H. Matondang, and D. S. Prasvita, "Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Penggunaan Aplikasi Jobstreet," 2022. doi: 10.33633/tc.v21i3.6361.
- [3] I. Saputra, T. Djatna, R. R. A. Siregar, D. A. Kristiyanti, H. R. Yani, and A. A. Riyadi, "Text Mining of PeduliLindungi Application Reviews on Google Play Store," *Fakt. Exacta*, vol. 15, no. 2, pp. 101–108, 2022, doi: 10.30998/faktorexacta.v15i2.10629.
- [4] T. F. T. Hidayat, G. Garno, and A. A. Ridha, "Analisis Sentimen Opini Pemindahan Ibu Kota Pada Twitter Dengan Metode Support Vector Machine," *J. Ilmu Komput.*, vol. 14, no. 1, p. 49, 2021, doi: 10.24843/jik.2021.v14.i01.p06.
- [5] 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.
- [6] G. J. Williams and Z. Huang, "Modelling the KDD Process," *Proc Cent. Softw. Reliab. Conf. Meas. Softw. Control Assur.*, no. June, pp. 1–8, 1996.
- [7] B. Herwijayanti, D. E. Ratnawati, and L. Muflikhah, "Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 306–312, 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [8] J. J. Aripin, "Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi pada BPR Pantura," 2019, [Online]. Available: <https://repository.nusamandiri.ac.id/index.php/repo/viewitem/13890>
- [9] A. S. Ritonga and E. S. Purwaningsih, "Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding)," *Ilm. Educat*, vol. 5, no. 1, pp. 17–25, 2018.
- [10] S. Y. Pangestu, Y. Astuti, and L. D. Farida, "Algoritma Support Vector Machine Untuk Klasifikasi Sikap Politik Terhadap Partai Politik Indonesia," *J. Mantik Penusa*, vol. 3, no. 1, pp. 236–241, 2019, [Online]. Available: <https://t.co/eF>