JNATIA Volume 3, Nomor 1, November 2024 p-ISSN: 2986-3929
Jurnal Nasional Teknologi Informasi dan Aplikasinya e-ISSN: 3032-1948

Deteksi Hate Speech pada Unggahan Media Sosial dengan Naive Bayes Menggunakan Seleksi Fitur Chi-Square

Putu Steven Belva Chan[a1], Ida Ayu Gde Suwiprabayanti Putra[a2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]stevenbelvachann@gmail.com
[2]iagsuwiprabayantiputra@unud.ac.id

Abstract

In the digital age, social media's pervasive use has revolutionized global communication but also introduced challenges like hate speech. This study proposes a Multinomial Naive Bayes model optimized with Chi-square feature selection to detect hate speech efficiently from large-scale social media data. Leveraging machine learning, this approach aims to combat harmful content by identifying relevant text features crucial for distinguishing hate speech from non-hate speech. The study utilizes TF-IDF for feature extraction and Chi-square for feature selection, showing significant performance improvements in hate speech detection. The Chi-square feature selection model yielded average precision, recall, F1-score, and accuracy values of 92%, 92%, 91%, and 92% respectively. In contrast, the model without feature selection achieved values of 89%, 89%, 88%, and 89% for the same metrics. Results demonstrate enhanced accuracy, precision, recall, and F1-score across various hate speech categories.

Keywords: *Hate Speech, Naive Bayes, TF-IDF, Chi-square*

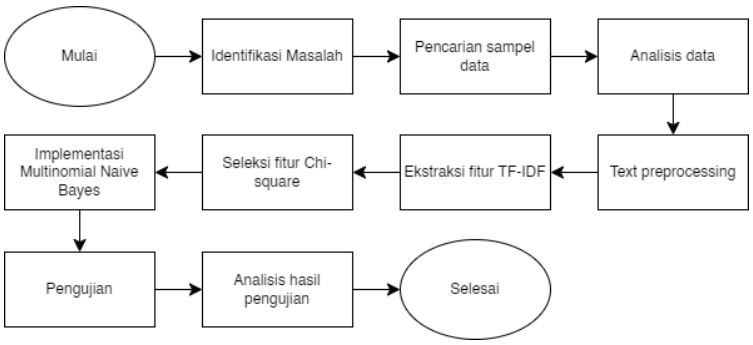
1. Pendahuluan

Di era digital saat ini, eksistensi media sosial telah membawa dampak signifikan dalam memfasilitasi komunikasi antar individu di seluruh dunia. Namun, bersamaan dengan kebebasan berekspresi yang ditawarkan oleh platform media sosial, kita juga menghadapi tantangan baru terkait penyalahgunaan informasi dan konten yang merugikan, seperti hate speech. Penggunaan media sosial secara massal menghasilkan volume data yang besar dan kompleks, membentuk apa yang kita kenal sebagai Big Data [1]. Data-data ini mencakup beragam informasi termasuk teks, gambar, video, dan lainnya yang diposting setiap detik. Di tengah kebebasan berbicara, kita juga melihat peningkatan dalam kasus hate speech, yang merupakan bentuk ujaran yang menghasut, menyerang, atau mengancam individu atau kelompok berdasarkan karakteristik seperti ras, agama, orientasi seksual, atau latar belakang budaya. Pentingnya penanggulangan hate speech di masyarakat semakin terasa, terutama mengingat dampaknya yang merusak kedamaian sosial dan kohesi komunitas. Hate speech dapat memicu konflik, memperburuk polarisasi sosial, dan menciptakan lingkungan online yang tidak aman dan tidak ramah [2]. Oleh karena itu, diperlukan pendekatan yang efektif untuk mendeteksi dan mengatasi hate speech secara proaktif di platform media sosial. Dalam menghadapi tantangan penanggulangan hate speech secara efektif, salah satu solusi yang ditawarkan adalah menggunakan metode klasifikasi teks berbasis machine learning, seperti Naive Bayes yang dioptimalkan dengan seleksi fitur Chi-square. Naive Bayes adalah algoritma klasifikasi yang populer dalam analisis teks karena sederhana dan efisien. Berdasarkan penelitian sebelumnya oleh Kurnia, dkk, didapatkan bahwa algoritma Naive Bayes mengungguli akurasi algoritma CNN [3]. Sementara itu, seleksi fitur digunakan untuk memilih subset optimal fitur untuk konstruksi model. Proses seleksi fitur menghitung skor dari setiap fitur yang mungkin berdasarkan teknik seleksi fitur tertentu, dan kemudian mengidentifikasi 'K' fitur terbaik[4]. Chi-square merupakan salah satu seleksi fitur yang

dapat membantu mengidentifikasi fitur-fitur teks yang paling relevan dan informatif untuk membedakan antara konten hate speech dan non-hate speech. Pada penelitian ini, penguji akan mengembangkan model deteksi hate speech menggunakan algoritma Multinomial Naive Bayes dengan seleksi fitur Chi-square yang dapat mengolah data dalam skala besar (Big Data) dari media sosial dengan efisien dan akurat. Dengan memanfaatkan teknik-teknik machine learning dan analisis data, diharapkan penelitian ini dapat memberikan kontribusi dalam upaya melindungi lingkungan online dari konten yang merugikan dan mempromosikan komunikasi yang inklusif dan bermartabat di dunia digital saat ini.

2. Metode Penelitian

2.1. Desain Penelitian



Gambar 1. Bagan Alur Penelitian

Penelitian ini diawali dengan mengidentifikasi terkait masalah yang ingin diselesaikan kemudian dilanjutkan dengan pencarian sampel data terhadap teks postingan di media sosial di kaggle.com yang nantinya akan dilakukan proses klasifikasi. Setelah data terkumpul, tahapan selanjutnya adalah melakukan proses text preprocessing. Kemudian, akan dilakukan perhitungan ekstraksi fitur menggunakan Term Frequency-Inverse Document Frequency (TF-IDF). Setelah ekstraksi fitur, selanjutnya adalah melakukan pelatihan model Multinomial Naive Bayes dengan menggunakan fitur yang telah diekstraksi dan diseleksi pada tahap selanjutnya. Tahap terakhir adalah melakukan analisis terhadap hasil pengujian yang didapat dari pengujian dengan model yang telah dibuat.

2.2. Pengumpulan Data

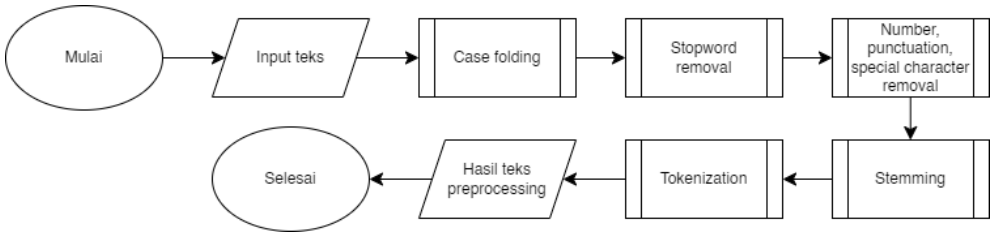
Data yang digunakan pada penelitian ini adalah data sekunder yang diambil dari Kaggle yaitu data hate speech dan abusive teks twitter. Data ini terdiri dari 13.169 data teks dan setiap record data dikelompokkan ke dalam 12 kategori ujaran kebencian. Berdasarkan target ujaran kebencian yaitu individu dan kelompok. Berdasarkan isi ujaran kebencian yaitu ras, agama, fisik, dan gender. Berdasarkan tingkat ujaran kebencian yaitu lemah, sedang, dan kuat.

Tabel 1. Gambaran Dataset

id	text	HS	Abusive	HS_Individual	HS_Race	HS_Moderate	…
1	Kalimat 1	1	0	1	1	0	…
2	Kalimat 2	0	1	0	0	1	…
3	Kalimat 3	1	0	0	1	1	…

2.3. Text Preprocessing

Text preprocessing adalah tahap awal yang biasa dilakukan saat melakukan klasifikasi teks. Text preprocessing adalah prosedur mengolah data tekstual tergantung pada beberapa kriteria untuk memperoleh teks murni[5]. Tahap ini akan menghasilkan teks yang siap digunakan ke dalam model machine learning. Tahap preprocessing pada penelitian ini ditunjukkan oleh gambar 2.



Gambar 2. Tahap Preprocessing Teks

Tahapan pertama dalam text preprocessing adalah melakukan case folding, yaitu mengubah semua huruf menjadi huruf kecil atau lowercase. Setelah dilakukan case folding, tahapan selanjutnya adalah menghilangkan karakter selain alfabet seperti tanda baca, angka, dan karakter spesial. Kemudian, akan dilakukan proses stemming pada data teks. Stemming dalam preprocessing teks adalah proses untuk memperoleh kata dasar (stem) dari sebuah kata dengan menghilangkan prefiks dan afiks kata tersebut[5]. Tahapan terakhir adalah melakukan tokenisasi seluruh kata pada setiap kalimat pada data teks.

2.4. Ekstraksi Fitur

Metode ekstraksi fitur yang digunakan dalam penelitian ini adalah Term Frequency-Inverse Document Frequency (TF-IDF). Pada klasifikasi teks, kata-kata yang terdapat pada dokumen merupakan fitur dari dokumen tersebut. Salah satu algoritma yang digunakan untuk mengekstraksi fitur ini adalah TF-IDF. Algoritma ini mencatat nilai yang menunjukkan seberapa penting sebuah kata pada dokumen. Prinsip kerja algoritma ini adalah jika sebuah kata sering muncul dalam sebuah dokumen tapi tidak pada dokumen lainnya, maka kata tersebut dapat dikatakan sebagai pembeda yang baik untuk klasifikasi. Nilai TF-IDF didapatkan dengan menghitung TF (term frequency) yang dikalikan dengan IDF (inverse document frequency).

Term frequency (TF) adalah proses untuk menghitung jumlah kemunculan sebuah kata dalam dokumen. Persamaan matematis dari TF dapat dilihat pada persamaan 1[6].

$$\mathrm{TF(t,d)} = \frac{\text{jumlah term dalam dokumen d}}{\text{jumlah kata dalam dokumen d}} \quad (1)$$

Inverse Document Frequency (IDF) adalah proses untuk menghitung nilai yang menunjukkan seberapa penting suatu term secara global dalam suatu koleksi dokumen. IDF memberi bobot lebih pada term yang sering muncul pada satu dokumen tertentu tapi jarang muncul pada dokumen secara global. Persamaan matematis dari IDF dapat dilihat pada persamaan 2[6].

$$\mathrm{IDF(t)} = \log\left(\frac{\mathrm{N}}{\mathrm{df(t)}}\right) \quad (2)$$

Dimana N adalah jumlah total dokumen dalam koleksi dan $\mathrm{df(t)}$ adalah jumlah dokumen yang mengandung term t dalam koleksi.

Rumus TF-IDF menggabungkan Term Frequency (TF) dan Inverse Document Frequency (IDF) untuk memberikan bobot pada setiap term dalam suatu dokumen dalam konteks seluruh koleksi dokumen[6].

$$\mathrm{TFIDF(t,d)} = \mathrm{TF(t,d)} \times \mathrm{IDF(t)} \quad (3)$$

2.5. Multinomial Naive Bayes

Multinomial Naive Bayes merupakan model pembelajaran probabilitas yang berasal dari teori Bayes dan tergolong dalam model supervised learning. Model ini banyak digunakan dalam Pengolahan Bahasa Alami (NLP) dan beroperasi berdasarkan prinsip frekuensi kata, yang mengukur kemunculan kata-kata dalam sebuah dokumen. Model ini menjelaskan dua aspek: kehadiran atau ketiadaan sebuah kata dalam teks, dan frekuensi kemunculan kata tersebut dalam dokumen. Persamaan Multinomial Naive Bayes dapat dilihat pada persamaan 4.

$$P(C|D) = P(C) \prod_{i-1}^{n} \quad P(W_i|C) \tag{4}$$

Probabilitas kelas C muncul dalam dokumen D, dilambangkan sebagai P(C│D), diturunkan dari persamaan 5, yang menghitung total jumlah kata dalam dokumen, yang dilambangkan sebagai Nc.

$$P(C) = \frac{Nc}{N} \tag{5}$$

Persamaan 6 menghitung probabilitas kelas C, P(C), dengan membagi total jumlah dokumen oleh jumlah dokumen kelas C, Nc.

$$P(w_i|C) = \frac{count(w_i|C)+1}{count(C)+|V|} \tag{6}$$

P(w_i│c) adalah probabilitas bersyarat kata ke-i dalam kelas C. count(w_i│C) adalah frekuensi kata ke-i dalam kelas C. count(C) adalah total jumlah kata dalam kelas C, sedangkan |V| adalah jumlah total kata unik di semua kelas.

2.6. Chi-square

Chi-square adalah salah satu algoritma seleksi fitur berbasis statistik yang paling efisien[7]. Formula Chi-Square terkait dengan fungsi seleksi fitur berbasis informasi yang mencoba untuk memahami bahwa fitur terbaik t_k untuk kategori c_i adalah yang paling berbeda dalam himpunan contoh positif dan negatif dari kelas c_i[8]. Persamaan matematis Chi-square dapat dilihat pada persamaan 7.

$$Chi - square(t_{k,}c_i) = \frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)} \tag{7}$$

Dimana N adalah jumlah total dokumen dalam dataset, A adalah jumlah dokumen dalam kategori c_i yang mengandung fitur t_k, B adalah jumlah dokumen yang mengandung fitur t_k dalam kategori lain, C adalah jumlah dokumen dalam kategori c_i yang tidak mengandung fitur t_k, D adalah jumlah dokumen yang tidak mengandung fitur t_k dalam kategori lain.

2.7. Evaluasi

Evaluasi dilakukan dengan melakukan testing pada model yang telah dibangun. Pada penelitian ini, dataset yang telah melalui proses *preprocessing* akan dibagi menjadi data training dan data testing masing-masing sebanyak 80% dan 20%. Kemudian dilakukan ekstraksi fitur dengan Term Frequency-Inverse Document Frequency (TF-IDF) dan seleksi fitur dengan algoritma Chi-square. Dua jenis model dibangun, satu model menggunakan seleksi fitur dan satu model lainnya tanpa seleksi fitur. Dari skenario tersebut, akan dilakukan testing pada masing-masing model untuk melakukan prediksi pada teks di setiap kategori. Dari hasil testing akan dilihat perbandingan performa model yang menggunakan seleksi fitur dan tanpa seleksi fitur. Performa yang ditinjau dalam penelitian ini adalah tingkat accuracy, recall, precision, dan F1-score. Perhitungan accuracy, recall, precision, dan F1-score dapat dihitung menggunakan rumus berikut[9]:

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

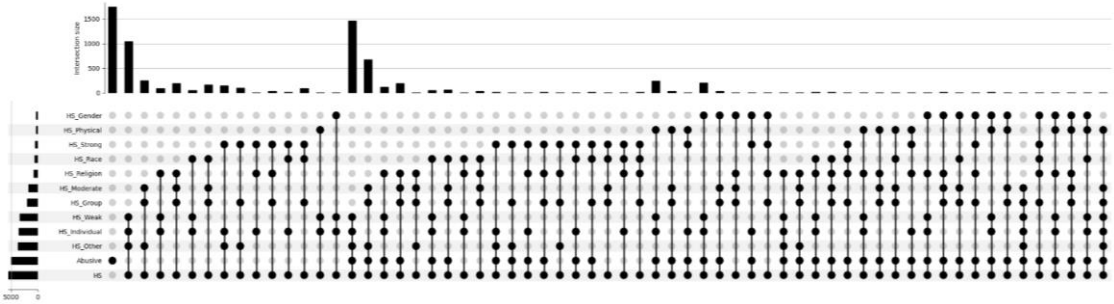
$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (11)$$

3. Hasil dan Diskusi

3.1. Analisis dataset



Gambar 3. Persebaran kategori dataset

Sebelum melakukan pemrosesan pada data, dilakukan analisis terhadap dataset terlebih dahulu. Analisis dilakukan dengan melihat bagaimana persebaran kategori dari data yang dimiliki. Dapat dilihat bahwa kategori dengan data terbanyak adalah kategori HS, diikuti kategori Abusive dan seterusnya. Dapat dilihat pula bahwa hampir seluruh data masuk ke dalam lebih dari 3 kategori. Jika teks adalah ujaran kebencian, maka data akan masuk ke dalam 3 kategori yang mewakili target, isi, dan tingkat ujaran kebencian serta dapat juga masuk ke dalam kategori *abusive*. Dapat dilihat pada ujaran kebencian yang paling banyak ditemukan adalah ujaran kebencian yang masuk ke kategori dengan target individu, isinya adalah kategori selain yang dikategorikan pada dataset, dan tingkat kebencian lemah.

3.2. Preprocessing data

Semua data teks pada dataset melalui tahap *preprocessing* untuk mendapatkan data teks yang lebih baik untuk digunakan dalam model. Tahapan dan hasil preprocessing dapat dilihat pada Tabel 2.

Tabel 2. Tahap dan Hasil Preprocessing

No.	Proses	Hasil
1	Initial Data	Film ini sangat bagus! Saya sangat terkesan dengan jalan ceritanya.
2	Case Folding	film ini sangat bagus! saya sangat terkesan dengan jalan ceritanya.
3	Stopword removal	film bagus! terkesan jalan ceritanya
4	Number, punctuation, special character removal	film bagus terkesan jalan ceritanya
5	Stemming	film bagus kesan jalan cerita

p-ISSN: 2986-3929
e-ISSN: 3032-1948

No.	Proses	Hasil
5	Tokenization	["film", "bagus", "kesan", "jalan", "cerita"]

3.3. Ekstraksi fitur

Setelah melalui tahap preprocessing, selanjutnya akan dilakukan ekstraksi fitur dengan algoritma Term Frequency-Inverse Frequency Document (TF-IDF). Untuk melakukan ekstraksi fitur TF-IDF, penulis menggunakan bantuan modul TfidfVectorizer dari library skcit-learn dan menggunakan bahasa pemrograman Python.

```
tfidf = TfidfVectorizer()
tfidf.fit(train_data['text'])

train_X_vec = tfidf.transform(train_data['text'])
val_X_vec = tfidf.transform(val_data['text'])
```

Gambar 4. Implementasi TF-IDF

3.4. Chi-square

Setelah dilakukan ekstraksi fitur TF-IDF, dapat dilakukan seleksi fitur dengan algoritma Chi-square, menggunakan bantuan modul SelectKBest dan chi2 dari library skcit-learn dengan jumlah *top feature* sebanyak 5000 fitur.

```
chi2_features = SelectKBest(chi2, k=5000)
X_kbest_features = chi2_features.fit_transform(X_tfidf_array, y)
```

Gambar 5. Implementasi Chi-square

3.5. Pemodelan Multinomial Naive Bayes

Model Multinomial Naive bayes dibangun dengan bantuan modul MultinomialNB dari library skcit-learn menggunakan bahasa pemrograman Python. Model akan dilatih dengan data training dari ekstraksi fitur TF-IDF tanpa seleksi fitur Chi-square dan dengan seleksi fitur Chi-square.

```
mnb = MultinomialNB()
mnb.fit(train_X_vec, train_y)

mnb_chi = MultinomialNB()
mnb_chi.fit(train_X_vec_chi, train_y)
```

Gambar 6. Implementasi Model MNB

3.6. Performa Model

Tabel 3. Perbandingan Performa Model

	Tanpa seleksi fitur Chi-square				**Dengan seleksi fitur Chi-square**			
	Precision	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Accuracy**
HS	81%	81%	81%	81%	89%	89%	89%	89%
Abusive	87%	87%	87%	87%	92%	92%	92%	92%

	Tanpa seleksi fitur Chi-square				**Dengan seleksi fitur Chi-square**			
	Precision	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Accuracy**
HS_Individual	79%	80%	78%	80%	86%	84%	82%	84%
HS_Group	88%	89%	86%	89%	90%	90%	88%	90%
HS_Religion	95%	95%	94%	95%	96%	96%	96%	96%
HS_Race	96%	96%	94%	96%	97%	97%	96%	97%
HS_Physical	98%	98%	97%	98%	98%	98%	97%	98%
HS_Gender	95%	98%	96%	98%	98%	98%	97%	98%
HS_Other	82%	82%	81%	82%	87%	87%	85%	87%
HS_Weak	79%	80%	78%	80%	85%	84%	81%	84%
HS_Moderate	87%	88%	85%	88%	90%	90%	87%	90%
HS_Strong	96%	97%	95%	97%	98%	98%	97%	98%
Rata-rata	89%	89%	88%	89%	92%	92%	91%	92%

Berdasarkan Tabel 3, dapat dilihat bahwa nilai precision, recall, F1-score, dan accuracy dari model Naive Bayes dengan mengimplementasikan seleksi fitur Chi-square di setiap kategori yang ada lebih besar dibandingkan model Naive Bayes tanpa seleksi fitur Chi-square. Nilai akurasi tanpa seleksi fitur di bawah 90% mengalami kenaikan yang cukup signifikan setelah dilakukan seleksi fitur. Data ini membuktikan bahwa performa model Naive Bayes dengan ekstraksi fitur TF-IDF dan seleksi fitur Chi-square lebih baik dari model tanpa seleksi fitur Chi-square.

4. Kesimpulan

Penelitian ini menunjukkan bahwa penggunaan metode Multinomial Naive Bayes yang dioptimalkan dengan seleksi fitur Chi-square mampu meningkatkan performa dalam mendeteksi hate speech pada media sosial. Model dengan seleksi fitur Chi-square memberikan nilai rata-rata precision, recall, F1-score, dan accuracy masing-masing sebesar 89%, 89%, 88%, dan 89%. Sementara itu, model tanpa seleksi fitur memberikan nilai masing-masing sebesar 92%, 92%, 91%, dan 92%. Hasil evaluasi menunjukkan peningkatan signifikan dalam metrik evaluasi seperti precision, recall, F1-score, dan accuracy pada berbagai kategori ujaran kebencian setelah penerapan seleksi fitur Chi-square.

Daftar Pustaka

[1] N. A. Ghani, S. Hamid, I. A. T. Hashem, E. Ahmed, “Social media big data analytics: A survey”, Computers in Human Behavior, vol. 101, p. 417, 2019.
[2] D. J. Ningrum, Suryadi, D. E. C. Wardhana, “Kajian Ujaran Kebencian Di Media Sosial”, Jurnal Ilmiah KORPUS, vol. 2, no. 3, p. 243, 2018.
[3] Y. Kurnia, E. D. Kusuma, L. W. Kusuma, Suwitno, W. Apridius, “Perbandingan Naïve Bayes dan CNN yang Dioptimasi PSO pada Identifikasi Berita Hoax Politik Indonesia”, Binary Digital - Technology, vol. 6, no. 3, p. 346, 2024.
[4] S. T. Ikram, A. K. Cherukuri, “Intrusion detection model using fusion of chi-square feature selection and multi class SVM”, Journal of King Saud University - Computer and Information Sciences. vol. 29, p. 463, 2017.
[5] M. F. Karaca, “Effects of Preprocessing on Text Classification in Balanced and Imbalanced Datasets”, Ksii Transactions on Internet and Information Systems, vol. 18, no. 3, p. 595, 2024.
[6] K. P. Harmandini, K. Muslim L, “Analysis of TF-IDF and TF-RF FeatureExtraction on Product Review Sentiment”, Sinkron: Jurnal dan Penelitian Teknik Informatika, vol. 8, no. 2, p. 933, 2024.
[7] C. Jin, T. Ma, R. Hou, M. Tang, Y. Tian, A. Al-Dhelaan, & M. Al-Rodhaan, “Chi-Square StatisticsFeature Selection Based on Term Frequency and Distribution for Text Categorization”, IETE Journal of Research, vol. 61, no. 4, pp. 351-362, 2015.
[8] S. Bahassine, A. Madani, M. Al-Sarem, M. Kissi, “Feature selection using an improved Chi-

square for Arabic text classification", Journal of King Saud University - Computer and Information Sciences. vol. 32, no. 2, pp. 225–231, 2020.

[9] Mas'udah E, Wahyuni ED, and Anjani A, "Analisis sentimen: Pemindahan ibu kota Indonesia pada twitter", Jurnal Informatika dan Sistem Informasi, vol. 1, no. 2, pp. 397-401, 2020.