

# Penerapan *Text Mining* Dengan Algoritma *Random Forest* Menganalisis Sentimen Ulasan SATUSEHAT *Mobile*

Bobby Rizki Nur Hidayat<sup>1</sup>, Rangga Sanjaya<sup>2</sup>

<sup>1</sup>Program Studi Teknik Informatika, Universitas Adhirajasa Reswara Sanjaya

<sup>2</sup>Program Studi Sistem Informasi, Universitas Adhirajasa Reswara Sanjaya

e-mail: <sup>1</sup>bobbyrizki15@gmail.com, <sup>2</sup>rangga@ars.ac.id

## Abstrak

Aplikasi SATUSEHAT Mobile merupakan hasil transformasi dari PeduliLindungi yang dikembangkan oleh Kementerian Kesehatan (Kemenkes) sebagai platform untuk menyebarkan informasi dan program kesehatan kepada masyarakat. Dengan lebih dari 1 juta ulasan dan nilai rating 3,7 bintang di Google Play Store, penting untuk menganalisis sentimen ulasan pengguna guna memahami pandangan pengguna terhadap aplikasi dan untuk meningkatkan kualitas aplikasi tersebut. Melalui pendekatan metode Teknik Text Mining, data ulasan pengguna dieksplorasi menggunakan algoritma pembelajaran mesin. Algoritma Multinomial Naive Bayes, K-Nearest Neighbors dan Decision Tree digunakan untuk membandingkan kinerja Random Forest dalam menganalisis sentimen ulasan pengguna tersebut. Hasilnya penelitian menunjukkan bahwa algoritma Random Forest memberikan analisis sentimen ulasan pengguna dengan tingkat akurasi tertinggi, sementara algoritma K-Nearest Neighbors (KNN) memiliki tingkat akurasi terendah. Penelitian ini memiliki relevansi penting dalam meningkatkan kualitas aplikasi SATUSEHAT Mobile berdasarkan pandangan dan umpan balik ulasan pengguna.

**Kata kunci**—*Algoritma, Aplikasi, Akurasi, Random Forest, Ulasan*

## Abstract

*The SATUSEHAT Mobile application is the result of a transformation from PeduliLindungi which was developed by the Ministry of Health (Kemenkes) as a platform for disseminating health information and programs to the public. With more than 1 million reviews and a 3.7 star rating value on the Google Play Store, it is important to analyze the sentiment of user reviews to understand how users view the app and to improve the quality of the app. Through the Text Mining Techniques approach, user data reviews are explored using machine learning algorithms. Multinomial Naive Bayes, K-Nearest Neighbors and Decision Tree Algorithms were used to compare the performance of Random Forest in analyzing the sentiments of these user reviews. The results show that the Random Forest algorithm provides sentiment analysis of user reviews with the highest accuracy, while the K-Nearest Neighbors (KNN) algorithm has the lowest accuracy. This research has important relevance in improving the quality of the SATUSEHAT Mobile application based on the views and feedback of user reviews*

**Keywords**—*Algorithms, Applications, Accuracy, Random Forests, Reviews.*

---

**Corresponding Author:**

**Rangga Sanjaya,**

Email: [rangga@ars.ac.id](mailto:rangga@ars.ac.id)

---

## 1. PENDAHULUAN

Di Indonesia, transformasi digital kesehatan sekali lagi mencapai tahap baru. Mulai 1 Maret 2023 PeduliLindungi telah dikembangkan menjadi aplikasi kesehatan masyarakat SATUSEHAT *Mobile* oleh Kementerian Kesehatan (Kemenkes) [1]. SATUSEHAT *Mobile* sebagai *platform* untuk menyebarluaskan *program* dan informasi seputar kesehatan. Aplikasi ini memiliki sejumlah fitur yang memudahkan pengguna diantaranya pencegahan *Covid-19*, notifikasi status vaksinasi (*screening*), pelacakan, notifikasi peringatan (*warning and fencing*), pelaksanaan layanan kesehatan dan berbagai informasi resmi layanan kesehatan dari Kementerian Kesehatan (Kemenkes). SATUSEHAT *Mobile* telah diunduh lebih dari 50 juta kali dan memiliki ulasan lebih dari 1 juta ulasan dengan nilai 3,7 bintang, menurut data dari situs *Google Play Store* per tanggal 04 Juni 2023 [2].

Sebagai bentuk pengetahuan dan evaluasi, ulasan pengguna dianggap sebagai sumber daya yang berguna bagi pengembang aplikasi untuk lebih memahami permintaan pengguna dan meningkatkan nilai bagi konsumen [3]. Ulasan seringkali terbuka dan mencantumkan peringkat bintang dalam skala 1 hingga 5 [4]. Namun, seperti yang terlihat dari nilai bintang yang diberikan oleh pengguna, tidak semua ulasan yang ditemukan di dalam aplikasi itu baik. Analisis ulasan aplikasi diperlukan karena sulit untuk membedakan antara perasaan positif dan negatif mengingat banyaknya data ulasan [5]. Ulasan antara perasaan positif dan negatif ini merupakan informasi tekstual yang dapat dievaluasi untuk mengetahui lebih jauh bagaimana pendapat masyarakat terhadap aplikasi dan hasilnya akan digunakan untuk menilai sentimen ulasan pengguna dari aplikasi SATUSEHAT *Mobile* [6]. Pendekatan metode *text mining* merupakan cara yang digunakan untuk menganalisis sentimen ulasan terhadap aplikasi tersebut [5].

*Text Mining* menjanjikan untuk menganalisis kumpulan teks masif secara efektif serta dengan cara yang transparan dan dapat diulang [7]. *Text mining* melibatkan berbagai alat penambangan teks untuk interpretasi dan analisis kalimat, kata, frasa, ucapan, klaim, iklan, prakiraan popularitas pemilu [8], Data tidak terstruktur ini secara efektif diubah menjadi representasi terstruktur yang dapat dibaca mesin yang cocok untuk analisis sentimen teks menggunakan berbagai algoritma pembelajaran mesin [9].

Teknik untuk menentukan bagaimana ekspresi sentimen teks dapat diklasifikasikan sebagai positif atau negatif merupakan teknik Analisis sentimen. Analisis sentimen digunakan untuk menginterpretasikan komentar pengguna dan menjelaskan tingkat penerimaan publik terhadap produk merek tertentu. Pengguna produk atau layanan dapat memposting ulasan, pemikiran, dan peringkat. Perasaan positif, netral, atau negatif dapat digambarkan secara tertulis dengan berbagai cara yang rumit [10]. Analisis sentimen merupakan bidang utama yang berfokus pada penelitian opini dan sudut pandang *public* yang memberikan informasi dari perspektif tersebut [11], seperti yang terdapat pada ulasan pengguna aplikasi SATUSEHAT *Mobile*. Oleh karena itu, analisis sentimen dapat menjadi proses klasifikasi yang pertama memproses teks menggunakan operasi seperti *tokenization*, *part-of-speech (PoS)*, dan standarisasi, kemudian menerapkan metode klasifikasi seperti pada pembelajaran mesin [11].

Salah satu pendekatan analisis sentimen menggunakan algoritma pembelajaran mesin adalah menggunakan algoritma *Multinomial Naïve Bayes*. Algoritma ini mendapatkan akurasi yang lebih baik daripada sekadar mencari kemunculan kata, *Multinomial Naïve Bayes* selalu merupakan metode yang lebih disukai untuk klasifikasi teks (termasuk deteksi spam, klasifikasi topik, dan analisis sentimen) [12]. Pengklasifikasi *Naïve Bayes* populer karena kesederhanaan dan keefektifannya yang besar karena didasarkan pada gagasan bahwa setiap karakteristik saling independen. Tetapi kinerja *Naïve Bayes* dalam tugas klasifikasi teks buruk karena fitur teks yang berlebihan dan estimasi parameter yang tidak tepat [13]. Sedangkan Algoritma *K-Nearest Neighbors (K-NN)* digunakan untuk menilai tingkat kesamaan antara dokumen dan K data pelatihan serta untuk menyimpan informasi klasifikasi dalam jumlah tertentu, sehingga mengidentifikasi kategori dokumen uji. Kapasitas untuk menemukan tetangga dari dokumen tertentu menggunakan ukuran kesamaan adalah komponen fundamental dari metode ini [14]. *Decision Tree*, pilihan terbaik jika kumpulan data dikategorikan dengan benar dan memiliki

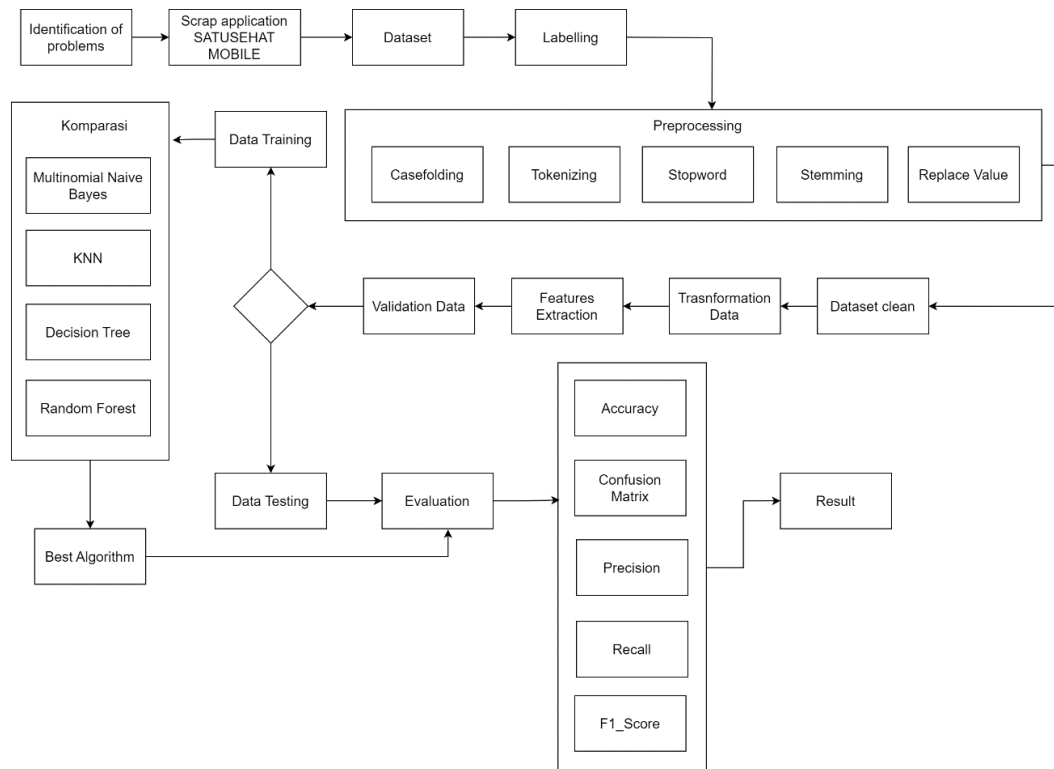
simpul sesedikit mungkin [15]. *Decision Tree* memberikan akurasi yang lebih baik dan lebih cepat daripada metode kategorisasi lainnya. Beberapa cabang *decision tree* mungkin mengandung *noise* atau *outlier* dalam data pelatihan selama proses klasifikasi data, yang menyebabkan *tree* yang kompleks dan menantang untuk dipahami. Oleh karena itu, prosedur pemangkasan digunakan untuk memotong cabang-cabang yang tersesat tersebut dalam upaya meningkatkan presisi. Selain itu, memotong area non-produktif *tree* akan menghasilkan *tree* yang lebih kecil dan tidak rumit [16]. Berbeda halnya dalam skenario *Random Forest*, beberapa *tree* klasifikasi dan regresi dibangun menggunakan *dataset* pelatihan yang dipilih secara acak dan *subset variabel* prediktor yang dipilih secara acak untuk memodelkan hasil. Hasil dari setiap *tree* digabungkan untuk menawarkan prediksi untuk setiap pengamatan. Karena itu, *Random Forest* sering memberikan akurasi yang lebih baik daripada model *Decision Tree* sambil mempertahankan beberapa sifat menguntungkan dari model *tree* tersebut [17].

Beberapa penelitian analisis sentimen, hanya menggunakan satu algoritma. seperti penelitian dengan judul, Analisis Sentimen Aplikasi Peduli Lindungi Menggunakan Metode *Naïve Bayes* [10], sama halnya dengan penelitian dengan judul Review Aplikasi *Text Mining* PeduliLindungi di *Google Play Store* [5]. Penelitian tersebut memiliki kekurangan karena hanya menggunakan satu algoritma yaitu Metode *Naïve Bayes* dan aplikasi yang diuji telah berganti menjadi *SATUSEHAT Mobile*. Kekurangan ini mengakibatkan tidak adanya perbandingan performa antara algoritma-algoritma lain yang juga dapat digunakan dalam klasifikasi teks. Untuk mendapatkan hasil penelitian yang lebih komprehensif dan objektif, perlu dilakukan perbandingan dengan menggunakan beberapa algoritma klasifikasi teks yang berbeda untuk memahami dan mengevaluasi kemampuan masing-masing algoritma dalam melakukan analisis sentimen ulasan aplikasi tersebut.

Dalam penelitian ini untuk menghadirkan algoritma klasifikasi, sentimen analisis terbaik dari data ulasan pengguna aplikasi *SATUSEHAT Mobile*, maka dibuatlah komparasi dari keempat algoritma tersebut yaitu *Multinomial Naïve Bayes*, *K-Nearest Neighbors*, *Decision Tree* dan *Random Forest*. Hasil komparasi digunakan sebagai perbandingan dan analisis sentimen ulasan pengguna terhadap aplikasi tersebut, dengan acuan melihat nilai *Accuracy* dari pemodelan keempat algoritma tersebut. Kemudian hanya algoritma terbaik yang akan menampilkan nilai yang lebih spesifik seperti *Confusion Matrix*, *Recall*, *Precision*, dan *F1\_Score*. Sehingga diharapkan hasil penelitian ini akan menjadi informasi yang bermanfaat bagi pengembang aplikasi *SATUSEHAT Mobile* dan membantu mereka menilai apa yang harus dilakukan untuk meningkatkan kualitas dan layanan dalam aplikasi tersebut.

## 2. METODE PENELITIAN

Komparasi algoritma digunakan dalam penelitian ini, untuk mendapatkan algoritma terbaik dalam menganalisis sentimen ulasan pengguna aplikasi *SATUSEHAT MOBILE*. Penelitian ini dilakukan dalam beberapa tahapan. Tahapan-tahapan tersebut ditunjukkan pada Gambar 1



Gambar 1. Tahapan Penelitian

### 2.1. Identification of Problem

Sebagaimana dinyatakan dalam pendahuluan penelitian ini, algoritma berikut akan digunakan dalam analisis sentimen untuk mengevaluasi ulasan pengguna aplikasi SATUSEHAT *Mobile*. Tujuan utama dari penelitian ini adalah untuk mengevaluasi kinerja dari algoritma *Multinomial Naive Bayes*, *K-Nearest Neighbors*, *Decision Tree*, dan *Random Forest* dalam mengklasifikasikan data dari *dataset* ulasan pengguna Aplikasi *Mobile* SATUSEHAT sehingga hasilnya bisa menentukan algoritma terbaik untuk analisis sentimen ulasan pengguna aplikasi tersebut.

### 2.2. Dataset

Tentu saja, sebuah *dataset* diperlukan dalam sebuah penelitian dan merupakan komponen utama baginya untuk memanfaatkan sebuah algoritma. *Dataset* yang digunakan dalam penelitian ini adalah hasil dari *scrape* menggunakan *Python* versi 3.7.2 pada aplikasi SATUSEHAT *Mobile* yang diambil dari *google playstore*. *Dataset* tersebut diambil dalam kurun waktu 2 bulan dari tanggal 03-02-2023 23-05-2023. *Dataset* ini berisi 20000 baris data

### 2.3. Labelling

*Dataset* tersebut diseleksi dengan hanya mengambil komentar ulasan pengguna dan diberi label dengan cara pelabelan dilakukan secara manual, kemudian ulasan pengguna tersebut dikelompokkan ke dalam 2 kategori ; ulasan yang memiliki nilai bintang 4 dan 5 dilabeli positif sedangkan ulasan yang memiliki nilai bintang 1 dan 2 dilabeli negatif. Ulasan yang memiliki nilai bintang 3 biasanya dikategorikan sebagai sentimen netral; namun, karena keluhan dan pujian mendominasi ulasan tersebut, peneliti menganggap sentimen netral kurang informatif maka tidak dilabeli dan tidak dimasukkan kedalam *dataset*[18]. Hasilnya *dataset* memiliki 19022 baris data, dengan 4971 data positif dan 14050 data negatif.

#### 2.4. Pre-Processing Data

*Preprocessing* adalah tahap pembersihan data. Biasanya, data tidak terstruktur dengan jumlah karakter yang besar diperoleh saat mengumpulkan data. Penghapusan *noise* adalah tujuan dari *preprocessing* [19]. Dalam *preprocessing* terdapat beberapa tahapan yang harus dilakukan diantaranya

1. *Case Folding*, adalah proses mengubah semua karakter kapital dalam dokumen menjadi huruf kecil. Karakter lain, seperti tanda baca dan spasi, yang tidak termasuk huruf dan angka, dianggap sebagai pembatas[5].
2. *Tokenize*, adalah proses yang digunakan dalam penelitian untuk memecah kalimat ulasan menjadi kata-kata individu. berfungsi sebagai entitas untuk menyusun matriks pada tahap selanjutnya, yang menentukan jumlah kata dalam suatu dokumen. Tahap ini memanfaatkan fungsi *Word Tokenize* dari pustaka *NLTK* [5].
3. *Stopword Removal*, Tahap *stop removal* digunakan untuk menghilangkan *term* yang kurang signifikan dan terlalu lazim karena frekuensinya dalam data review terlalu tinggi. Saat ini library *NLTK* digunakan dengan fungsi *stopwords* dan parameter Bahasa Inggris dan Bahasa Indonesia[5].
4. *Stemming* menggunakan *Sastrawi*. Pada tahapan ini, sebuah kata direduksi menjadi bentuknya yang paling mendasar. Prosedur *Stemming* dapat digunakan untuk menghilangkan imbuhan yang melekat pada kata dasar dalam bahasa Indonesia [5].
5. *Replace Value*, digunakan untuk mengganti nilai *string* kosong yang terdapat dalam dataset. Dalam penelitian ini nilai *string* kosong tersebut diisi dengan cara setiap baris dengan nilai *string* kosong diisi dengan nilai baris terdekat di atasnya.

#### 2.5. Dataset Clean

Setelah tahapan pembersihan data, *dataset* telah dibersihkan secara efektif dari segala jenis gangguan melalui tahapan pembersihan menyeluruh, termasuk simbol, emoji, tanda baca, dan item lain yang tidak relevan atau mengganggu. Hasilnya, dataset tersebut telah lebih bersih, lebih teratur, dan bebas dari hambatan apa pun yang dapat mengganggu analisis data. Setelah melewati tahap ini, data disiapkan untuk diuji terhadap algoritma atau model terkait untuk menghasilkan temuan atau analisis yang lebih tepat dan akurat pada penelitian tersebut

#### 2.6. Transformation Data

Selanjutnya pada tahap ini *dataset* pada kolom berlabel "*value*" sekarang diubah menjadi tipe data kategori. Hal ini dilakukan untuk memudahkan pengelompokan dan analisis data menurut kategori yang telah ditetapkan sebelumnya. Kemudian kolom dengan label "*comment*", yang berfungsi untuk mewakili data tekstual dalam kumpulan data, diubah menggunakan tipe data *string*.

#### 2.7. Features Extraction

*CountVectorizer* dan *TF-IDF* adalah dua prosedur yang digunakan dalam tahap ekstraksi fitur. *CountVectorizer* digunakan untuk menghitung kata dalam dokumen serta mengubah teks menjadi format *vektor*. Sementara itu, *TF-IDF* digunakan untuk menimbang jumlah kata dengan ukuran seberapa sering kata tersebut muncul dalam sebuah dokumen. Pustaka *sklearn* digunakan dalam implementasi *TF-IDF*, fungsi yang disebut *TfidfTransformer* digunakan [20].

#### 2.8. Validation Data

Dalam penelitian ini, metode validasi yang digunakan adalah validasi pemisahan (*Split Validation*). *Split Validation* adalah metode validasi yang membagi data sampel set menjadi *training* dan *testing* set secara acak [20]. Model prediktif dilatih menggunakan data *training*, dan

keefektifan model dinilai menggunakan data *testing*. Keefektifan kemampuan model untuk meramalkan secara akurat data yang belum pernah dilihat sebelumnya dapat dievaluasi dengan menggunakan teknik ini [21]. Untuk menilai kinerja setiap model dan mengidentifikasi model yang memberikan hasil klasifikasi terbaik, *dataset* akan dibagi menjadi set data *training* dan data *testing*. *Dataset* terlebih dahulu akan dipecah menjadi data *training* dan *testing* kemudian dilakukan *split validation* dengan rasio 0,5 - 0,9 seperti yang ditunjukkan tabel 1. Data *testing* akan digunakan untuk mengevaluasi performa model pada data yang belum pernah dilihat sebelumnya, sedangkan data *training* akan digunakan untuk melatih model.

Tabel 1. Persebaran Data

No.	Split Validation	Data Training	Data Testing
1	0.9	17.119	1.902
2	0.8	15.217	3.804
3	0.7	13.315	5.706
4	0.6	11.413	7.608
5	0.5	9.511	9.511

## 2.9. Comparison Algorithm

Tahap komparasi algoritma merupakan tahap yang penting, penelitian ini menggunakan empat model algoritma yang telah disebutkan sebelumnya yaitu *Multinomial Naive Bayes*, *K-Nearest Neighbors (KNN)*, *Decision Tree*, dan *Random Forest*. Tahapan *Validation Data* dimulai ketika data *training* dan data *testing* telah dipisahkan. Model algoritma tersebut semuanya akan dilatih menggunakan data *training*. Untuk mendapatkan hasil klasifikasi analisis sentimen terbaik, prosedur pelatihan melibatkan *preprocessing* koleksi dokumen seperti kategorisasi teks, ekstraksi informasi, ekstraksi kata, dan lain-lain. Selanjutnya akan dilakukan uji terhadap data *testing* untuk melihat model dan performa dari semua algoritma. *Accuracy* adalah hasil yang nantinya akan digunakan untuk menilai hasil analisis sentiment pengguna aplikasi tersebut. Untuk menganalisis sentimen pengguna terhadap aplikasi, hasil ini akan memberikan pemahaman yang lebih baik tentang kinerja dan kemampuan klasifikasi dari masing-masing model. Model yang menghasilkan nilai *Accuracy* analisis sentimen terbaik untuk *dataset* ulasan pengguna aplikasi SATUSEHAT *Mobile* yang digunakan dalam penelitian ini dapat diidentifikasi dengan komparasi hasil *Accuracy* algoritma tersebut. Pemilihan model terbaik akan memainkan peran penting dalam kesimpulan

### 1. *Multinomial Naive Bayes*,

Teknik pembelajaran mesin yang memanfaatkan perhitungan probabilitas adalah *Multinomial Naive Bayes*. Teorema *Bayes* secara efektif diterapkan pada kategorisasinya menggunakan set pelatihan dari beberapa data. *Naive Bayes* membuat asumsi bahwa nilai atribut di satu kelas tidak terkait dengan atribut di kelas lain. *Thomas Bayes*, seorang fisikawan Inggris, mempopulerkan teorema *Bayes*, yang membuat prediksi tentang peluang berdasarkan data historis[10]. Berikut formula *Bayes* ditunjukkan persamaan (1), Secara umum, teorema *Bayes* dinyatakan sebagai berikut;

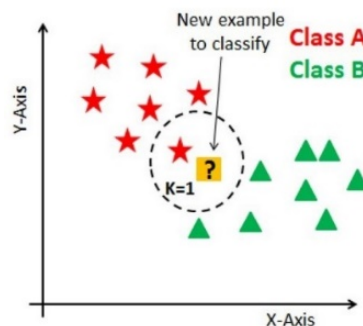
$$P(A|B) = \frac{P(A|B) * P(A)}{P(B)} \quad (1)$$

Keterangan:

Pada notasi ini,  $P(A|B)$  menggambarkan probabilitas peristiwa A terjadi jika peristiwa B telah terjadi, sedangkan  $P(B|A)$  menggambarkan probabilitas peristiwa B terjadi jika peristiwa A telah terjadi.

## 2. *K-Nearest Neighbors*

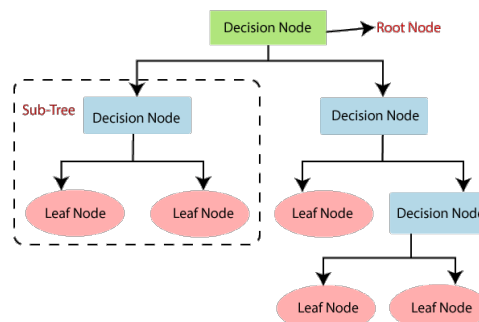
Algoritma *K-Nearest Neighbors* adalah untuk menilai tingkat kesamaan antara dokumen dan  $k$  data pelatihan serta untuk menyimpan data klasifikasi dalam jumlah tertentu. Teknik ini merupakan algoritma pembelajaran berbasis instan yang mengklasifikasikan objek berdasarkan ruang fitur terdekat dari set pelatihan. Ruang fitur multidimensi dipetakan ke set pelatihan. Berdasarkan klasifikasi training set, ruang fitur dibagi menjadi area geografis. Jika sebuah kategori adalah yang paling lazim di antara  $k$  data pelatihan terdekat, sebuah titik dalam ruang fitur dialokasikan ke kategori tersebut. Jarak antara vektor biasanya dihitung dengan menggunakan *Euclidean Distance*[14]. Dalam Gambar 2 bisa dilihat struktur algoritma *K-Nearest Neighbors* (KNN).



Gambar 2. Struktur *K-Nearest Neighbors* (KNN)

## 3. *Decision Tree*

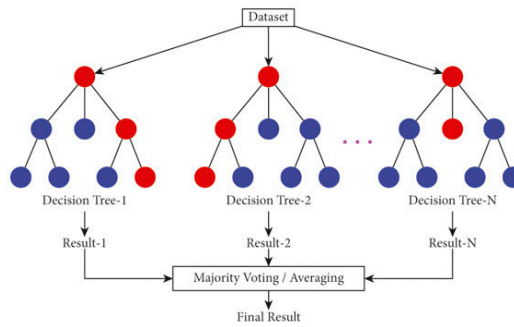
Salah satu teknik yang sering digunakan dalam berbagai domain, termasuk pembelajaran mesin, pemrosesan gambar, dan pengenalan pola adalah *Decision Tree*., *Decision Tree* adalah model klasifikasi yang sering digunakan dalam *data mining*. Setiap *Tree* terdiri dari simpul dan cabang. Setiap *node* mewakili fitur dalam kategori klasifikasi, dan setiap *subset* menentukan nilai yang dapat diterima oleh *node*. *Decision Tree* telah menemukan beberapa domain implementasi karena analisis langsung dan akurasi pada banyak tipe data [15]. Dalam Gambar 3 dapat dilihat struktur algoritma *Decision Tree*.



Gambar 3. Struktur *Decision Tree*

## 4. *Random Forest*

Salah satu teknik *Random Forest* merupakan teknik pembelajaran *ensemble* berbasis *decision tree*. *Random Forest* bertujuan untuk membuat masing-masing estimator lebih andal atau kokoh. Klasifikasi dari sejumlah estimator mendasar yang dibuat melalui prosedur *decision tree* digabungkan dalam *Random Forest*. [17]. Dalam Gambar 4 dapat dilihat struktur algoritma *Random Forest*.



Gambar 4. Struktur Random Forest

2.10. Best Algorithm

Pada tahap ini, dilakukan pemilihan algoritma terbaik yang mampu memberikan klasifikasi analisis sentimen ulasan pengguna aplikasi SATUSEHAT *Mobile*. Algoritma terbaik dipilih berdasarkan nilai *Accuracy* yang paling tinggi dari komparasi hasil algoritma lain yang digunakan dalam penelitian ini. Selain itu, hasil *Accuracy* yang didapatkan juga sejalan dengan nilai dari *confusion matrix*, *precision*, *recall*, dan *f1-score* yang memperoleh nilai tinggi dari algoritma tersebut.

2.11. Evaluation

Kemudian pada tahap ini, evaluasi kinerja model algoritma, *Multinomial Naive Bayes*, *K-Nearest Neighbor*, *Decision Tree*, dan *Random Forest* dalam analisis sentimen ulasan pengguna aplikasi tersebut dapat dilihat dan dinilai dari hasil *Accuracy* yang umum dikeluarkan dalam *text mining*, Nantinya, dalam penelitian ini, hanya algoritma terbaik yang akan ditampilkan nilai yang lebih spesifik seperti *Confusion Matrix*, *Recall*, *Precision*, dan *f1\_Score*.

1. *Accuracy*, adalah rasio prediksi yang benar (baik positif maupun negatif) terhadap semua data adalah ukuran akurasi. Dengan kata lain, akurasi adalah sejauh mana nilai prediksi menyerupai nilai nyata (aktual).

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

2. *Confusion Matrix*, adalah tabel yang pada dasarnya membandingkan hasil klasifikasi yang dihasilkan oleh sistem (model) dengan hasil klasifikasi aktual. Performa model klasifikasi pada sejumlah set data uji dengan nilai nyata dijelaskan oleh *confusion matrix* yang berbentuk tabel pada Gambar 5 dibawah ini.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<p><b>TP</b> (True Positive)</p>	<p><b>FP</b> (False Positive) <i>Type I Error</i></p>
	0 (Negative)	<p><b>FN</b> (False Negative) <i>Type II Error</i></p>	<p><b>TN</b> (True Negative)</p>

Gambar 5. Tabel *Confusion Matrix*



TP adalah singkatan dari *True Positive*, yang berarti hasil prediksi positif sesuai dengan realitas positif. *False Positive* (FP) mengacu pada situasi di mana hasil positif tidak sesuai dengan situasi sebenarnya. FN adalah singkatan dari *False Negative*, yang berarti temuan ramalan negatif tidak sesuai dengan kenyataan positif. Terakhir, TN adalah *True Negative*, yang berarti perkiraan hasil negatif cocok dengan hasil actual [7].

3. *Recall*, adalah keberhasilan model dalam pencarian informasi kembali. *Recall* merupakan rasio perkiraan positif yang benar terhadap semua data positif yang benar.

$$Recall = \frac{(TP)}{(TP + FN)} \times 100\% \quad (3)$$

4. *Precision*, adalah tingkat akurasi antara data yang diinginkan dan hasil yang disampaikan oleh model disebut sebagai *precision*. Rasio hasil yang diprediksi dengan benar yang positif terhadap seluruh hasil yang diprediksi disebut *precision*. Berapa banyak kelas yang benar-benar positif yang telah diprediksi dengan benar dari semua kelas yang positif.

$$Precision = \frac{(TP)}{(FP + TP)} \times 100\% \quad (4)$$

5. *f1\_score*, adalah ukuran rata-rata dari *precision* dan *recall* yang dibobotkan.

$$f1\_score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \times 100\% \quad (5)$$

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Scrapping Dataset

Tahap pengumpulan data ini dilakukan menggunakan aplikasi *Jupyter Notebook* dengan teknik *scraping* dan program bahasa *Python*. Dengan bantuan library *google-play-scrapers* untuk mengambil data ulasan pengguna aplikasi SATUSEHAT *Mobile* di *Google Play Store*. Dataset hasil *scraping* merupakan data yang diambil dalam kurun waktu 2 bulan dari tanggal 03-02-2023 sampai tanggal 23-05-2023. Hasilnya didapatkan data yang terdiri dari 20.000 baris data. Dalam tabel 2, *Dataset* tersebut telah dilakukan pemilahan dengan hanya mengambil beberapa atribut yang penting dalam penelitian ini diantaranya *username* merupakan nama pengguna, *score* adalah nilai yang di berikan pengguna, *at* adalah waktu dan *content* adalah komentar atau ulasan pengguna. Kemudian dataset tersebut disimpan dengan nama *dataset\_saturehat* dalam bentuk format *.csv*.

Tabel 2. *Dataset* Aplikasi SATUSEHAT *Mobile*

	userName	score	at	content
0	W mel	1	2023-05-23 12:22:54	Saya sudah booster 1 (vaksin 3) .. Aplikasi ...
1	M kudsi Rahmi	1	2023-05-23 12:10:58	Apk ga jelas , kotel
2	Muhammad Dores223	5	2023-05-23 11:50:37	Oke
...				
...				
19999	Little Pony	4	2023-03-02 14:30:50	Ini kenapa ga bisa dibuka setelah diperbaruuii ...

### 3.2. Labelling

Pelabelan dilakukan secara manual dengan mengelompokkan ulasan pengguna ke dalam dua kategori: positif dan negatif. Ulasan dengan nilai bintang 4 dan 5 dilabeli sebagai positif, sementara ulasan dengan nilai bintang 1 dan 2 dilabeli sebagai negatif. Ulasan dengan nilai bintang 3 tidak diberi label karena dianggap kurang informatif. Sebagai hasilnya, dataset terdiri dari 19.022 baris data, dengan 4.971 data positif dan 14.050 data negatif. Kemudian Nama atribut pada tabel 3 diganti dari atribut *score* dan *content* menjadi atribut *value* dan *comment*. *value* merupakan *label* sedangkan *comment* adalah ulasan .

Tabel 3. *Labelling Dataset Aplikasi SATUSEHAT Mobile*

value	Comment
0	Negatif
1	Negatif
2	Negatif
...	...
19019	Postitif
19020	Postitif
19021	Postitif

### 3.3. Preprocessing Data

Setelah pengumpulan dan pelabelan *dataset*, data tersebut kemudian masuk ke tahap *preprocessing* untuk tahap pembersihan data dan penghapusan *noise*. Proses *preprocessing* melibatkan beberapa tahapan yang harus dilakukan, berikut tahapan-tahapannya; *Case Folding, Tokenize, Stopword Removal, Stemming* dan *Replace Values*.

### 3.4. Dataset Baru

Setelah tahapan *preprocessing* data selesai, kemudian *dataset* disimpan sebagai data baru yang telah bersih dari segala macam gangguan . Dapat dilihat pada tabel 4, data telah bersih dan sangat berbeda jika dibandingkan dengan *dataset* awal. Lalu *dataset* tersebut disimpan dengan nama *data\_clean* dalam bentuk format *.csv*.

Tabel 4. *Dataset Baru*

value	comment
0	Negatif
1	Negatif
2	Negatif
...	...
19019	Postitif
19020	Postitif
19021	Postitif

### 3.5. Transformation data

Kemudian selanjutnya dilakukan *Transformation data*, yaitu *dataset* pada label *value* akan diubah menjadi tipe data kategori. Kemudian atribut "*comment*", diubah menggunakan tipe data *string* Hal ini bertujuan untuk meningkatkan efektivitas dan keakuratan analisis data serta mempermudah pengambilan keputusan berdasarkan data dalam *dataset*.

### 3.6. Feature Extraction

Setelah itu tahapan *Feature Extraction* dilakukan. *CountVectorizer* dan *TF-IDF* adalah dua prosedur yang digunakan dalam tahap ekstraksi fitur. *CountVectorizer* digunakan untuk menghitung kata dalam dokumen serta mengubah teks menjadi format Vector. Sementara itu, *TF-IDF* adalah metode yang menggabungkan frekuensi kata dalam dokumen (TF) dengan kebalikannya, yaitu frekuensi kata dalam seluruh *korpus* (IDF).

### 3.7. Hasil Pengujian Dataset

Tahapan selanjutnya model akan diuji, untuk menentukan algoritma *text mining* mana yang memberikan hasil klasifikasi terbaik untuk analisis sentimen ulasan pengguna aplikasi SATUSEHAT *Mobile*. Peneliti menguji *dataset* menggunakan 4 algoritma *text mining* dalam penelitian ini yaitu; *Multinomial Naïve Bayes*, *K-Nearest Neighbor*, *Decision Tree*, dan *Random Forest*. Kemudian hasil *Accuracy* akan ditampilkan dan digunakan sebagai penilaian untuk menilai hasil klasifikasi analisis sentimen pada penelitian ini. Untuk mencari hasil *Accuracy* terbaik dari algoritma, maka pengujian dilakukan 5 kali dengan rasio perbandingan data yang berbeda-beda, hasil pengujiannya ditunjukkan tabel 5.

Tabel 5. Hasil Pengujian Dataset

Algoritma	Accuracy				
	rasio 0.9	rasio 0.8	rasio 0.7	rasio 0.6	rasio 0.5
<i>Multinomial Naïve Bayes</i>	0,87	0,88	0,88	0,88	0,87
<i>K-Nearest Neighbors</i>	0,57	0,52	0,52	0,53	0,86
<i>Decision Tree</i>	0,87	0,88	0,87	0,87	0,87
<i>Random Forest</i>	0,89	0,90	0,90	0,90	0,90

Kemudian dilakukan komparasi hasil evaluasi, masing-masing algoritma di uji dengan lima perbandingan rasio data yang berbeda untuk menghasilkan hasil yang maksimal dan optimal. Setelah melalui proses pengujian didapatkan hasil terbaik dari masing-masing algoritma, dapat dilihat bahwa komposisi rasio data yang optimal adalah 0.8 tetapi dalam algoritma *K-NearestNeighbors* rasio yang terbaik adalah 0.5. Sehingga dalam rangka menemukan algoritma terbaik dalam analisis sentimen ulasan pengguna aplikasi SATUSEHAT *Mobile*, dilakukan komparasi antara performa masing-masing algoritma. Algoritma klasifikasi yang memiliki hasil *Accuracy* terbaik dapat diidentifikasi sebagai algoritma terbaik untuk diterapkan guna klasifikasi analisis sentimen ulasan pengguna aplikasi SATUSEHAT *Mobile* berdasarkan perbandingan hasil prediksi dan tingkat stabilitas kinerja algoritma tersebut. Berdasarkan hasil pengujian, seperti yang terdapat di tabel 6 menunjukkan *Random Forest* adalah algoritma terbaik untuk analisis sentimen ulasan pengguna aplikasi SATUSEHAT *Mobile*

Tabel 6. Hasil Komparasi Algoritma

Algoritma	Rasio	Accuracy
<i>Multinomial Naivebayes</i>	0,8	0,88
<i>K-NearestNeighbor</i>	0,5	0,86
<i>Decision Tree</i>	0,8	0,88
<i>Random Forest</i>	0,8	0,90

Selanjutnya, dalam evaluasi algoritma *Random Forest* sebagai algoritma terbaik, tidak hanya akan dilakukan pengukuran tingkat *Accuracy*, tetapi juga akan dievaluasi menggunakan *Confusion Matrix*, *Precision*, *Recall*, dan *F1-Score*. Berikut ini hasil evaluasi seperti yang ditunjukkan gambar 6 dengan menggunakan komposisi rasio data yang optimal 0.8. Dalam gambar tersebut terdapat 2741 data positif yang benar diprediksi (*True Positive*), 113 Data yang seharusnya negatif, justru diprediksi sebagai data positif (*False Positive*), 257 data yang pada

kenyataannya adalah positif, tetapi diprediksi sebagai data negatif (*False Negative*), dan 694 data prediksi yang benar untuk kelas negatif (*True Negatif*). Selain itu dari data tersebut dapat dilakukan perhitungan manual untuk menghitung tingkat *Accuracy*, *Confusion Matrix*, *Precision*, *Recall*, dan *F1-Score* seperti berikut ini.

```

confusion matrix:
[[2741 113]
 [ 257 694]]
=====

```

	precision	recall	f1-score	support
Negatif	0.91	0.96	0.94	2854
Positif	0.86	0.73	0.79	951
accuracy			0.90	3805
macro avg	0.89	0.85	0.86	3805
weighted avg	0.90	0.90	0.90	3805

Gambar 6. Hasil Evaluasi Algoritma *Random Forest*

### 1. Accuracy

Pengujian mengungkapkan bahwa algoritma *Random Forest* memiliki tingkat akurasi 90%. Gambar 6 digunakan untuk melakukan perhitungan manual untuk menganalisis lebih lanjut. Hasil perhitungan manual adalah sebagai berikut:

$$Accuracy = \frac{(694+2741)}{(694+2741+113+257)} \times 100\% = 90\% \quad (6)$$

### 2. Precision

Tingkat akurasi antara data yang diinginkan dan hasil yang disampaikan oleh *Precision* penelitian ini, mengungkapkan bahwa algoritma *Random Forest* menghasilkan *Precision* Negatif 91%. Dan *Precision* Positif 86%. Gambar 6 digunakan untuk melakukan perhitungan manual untuk menganalisis lebih lanjut. Hasil perhitungan manual adalah sebagai berikut:

$$Precision\ Negatif = \frac{(2741)}{(2741+257)} \times 100\% = 91\% \quad (7)$$

$$Precision\ Positif = \frac{(694)}{(694+113)} \times 100\% = 86\% \quad (8)$$

### 3. Recall

Keberhasilan model dalam pencarian informasi kembali dalam pengujian mengungkapkan bahwa algoritma *Random Forest* memiliki tingkat *Recall* Negatif 96%. Dan *Recall* Positif 73%. Gambar 6 digunakan untuk melakukan perhitungan manual untuk menganalisis lebih lanjut. Hasil perhitungan manual adalah sebagai berikut

$$Recall\ Negatif = \frac{(2741)}{(2741+113)} \times 100\% = 96\% \quad (9)$$

$$Recall\ Positif = \frac{(694)}{(694+257)} \times 100\% = 73\% \quad (10)$$

### 4. F1\_Score

Rata-rata dari *precision* dan *recall* yang dibobotkan dalam pengujian mengungkapkan bahwa algoritma *Random Forest* diperoleh nilai *f1\_score* sebesar 94% untuk negatif dan 79% untuk

positif. Gambar 6 digunakan untuk melakukan perhitungan manual untuk menganalisis lebih lanjut. Hasil perhitungan manual adalah sebagai berikut diperoleh sebagai berikut.

$$f1\_score\ negatif = 2x \frac{(0.91 \times 0.96)}{(0.91 + 0.96)} \times 100\% = 94\% \quad (11)$$

$$f1\_score\ positif = 2x \frac{(0.86 \times 0.73)}{(0.86 + 0.73)} \times 100\% = 79\% \quad (12)$$

#### 4. KESIMPULAN

Penelitian ini dilakukan untuk menganalisis sentimen ulasan pengguna aplikasi SATUSEHAT *Mobile* berbasis Teknik *Text Mining* dengan menggunakan algoritma ; *Multinomial Naïve Bayes*, *K-Nearest Neighbors (KNN)*, *Decision Tree* dan *Random Forest*. Hasilnya bahwa ulasan negatif mendominasi dalam ulasan pengguna aplikasi SATUSEHAT *Mobile*. Data menunjukkan bahwa dari 19.022 data ulasan, sebanyak 14.050 ulasan atau sekitar 73.8% merupakan ulasan negatif. Hal ini mengindikasikan bahwa banyak pengguna yang memiliki pandangan negatif terhadap aplikasi tersebut. Dalam menghadapi dominasi ulasan negatif, rekomendasi model menggunakan algoritma *Random Forest* dapat membantu dalam mengidentifikasi dan memahami sentimen ulasan pengguna. Algoritma *Random Forest* memberikan nilai *Accuracy* tertinggi dibandingkan dengan algoritma lainnya. Dengan nilai *Accuracy* sebesar 90%. Sedangkan untuk nilai *Accuracy* terendah dihasilkan oleh algoritma *K-Nearest Neighbors (KNN)* dengan nilai sebesar 52%. Dengan hasil berikut *Algoritma Random Forest* memberikan pemahaman yang baik tentang ulasan negatif, pengembang aplikasi dapat mengambil tindakan yang diperlukan untuk meningkatkan kualitas dan pengalaman pengguna aplikasi SATUSEHAT *Mobile*.

#### DAFTAR PUSTAKA

- [1] Rokom, “Besok PeduliLindungi Resmi Bertransformasi Menjadi SATUSEHAT Mobile,” sehatNegeriku. Accessed: Jun. 04, 2023. [Online]. Available: <https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20230228/2042474/besok-pedulilindungi-resmi-bertransformasi-menjadi-satusehat-mobile/>
- [2] Kementerian Kesehatan Republik Indonesia, “SATUSEHAT Mobile.” Accessed: Jun. 04, 2023. [Online]. Available: [https://play.google.com/store/apps/details?id=com.telkom.tracencare&hl=en\\_US](https://play.google.com/store/apps/details?id=com.telkom.tracencare&hl=en_US)
- [3] H. Yang and P. Liang, “Identification and Classification of Requirements from App User Reviews.” [Online]. Available: <http://www.perculasoft.com/appreviews/>
- [4] D. T. Alamanda, A. Ramdhani, I. Kania, W. Susilawati, and E. S. Hadi, “Sentiment Analysis Using Text Mining of Indonesia Tourism Reviews via Social Media,” *International Journal of Humanities, Arts and Social Sciences*, vol. 5, no. 2, pp. 72–82, Apr. 2019, doi: 10.20469/ijhss.5.10004-2.
- [5] I. Saputra *et al.*, “Text Mining of PeduliLindungi Application Reviews on Google Play Store,” vol. 15, no. 2, pp. 1979–276, 2022, doi: 10.30998/faktorexacta.v15i2.10629.
- [6] I. Yunanto and S. Yulianto, “TWITTER SENTIMENT ANALYSIS PEDULILINDUNGI APPLICATION USING NAÏVE BAYES AND SUPPORT VECTOR MACHINE,” *Jurnal Teknik Informatika (Jutif)*, vol. 3, no. 4, pp. 807–814, Aug. 2022, doi: 10.20884/1.jutif.2022.3.4.292.
- [7] D. Antons, E. Grünwald, P. Cichy, and T. O. Salge, “The application of text mining methods in innovation research: current state, evolution patterns, and development priorities,” in *R and D Management*, Blackwell Publishing Ltd, Jun. 2020, pp. 329–351. doi: 10.1111/radm.12408.

- [8] “View of A COMPARISON STUDY ON TEXT MINING AND SENTIMENT ANALYSIS FEATURES AND FUNCTIONS USING SAS ENTERPRISE MINER, PYTHON AND R”.
- [9] O. J. Achilonu, V. Olago, E. Singh, R. M. J. C. Eijkemans, G. Nimako, and E. Musenge, “A text mining approach in the classification of free-text cancer pathology reports from the south african national health laboratory services,” *Information (Switzerland)*, vol. 12, no. 11, Nov. 2021, doi: 10.3390/info12110451.
- [10] Z. Rais, F. T. T. Hakiki, and R. Aprianti, “Sentiment Analysis of Peduli Lindungi Application Using the Naive Bayes Method,” *SAINSMAT: Journal of Applied Sciences, Mathematics, and Its Education*, vol. 11, no. 1, pp. 23–29, Jun. 2022, doi: 10.35877/sainsmat794.
- [11] S. Q. Ong, M. B. M. Pauzi, and K. H. Gan, “Text mining in mosquito-borne disease: A systematic review,” *Acta Tropica*, vol. 231. Elsevier B.V., Jul. 01, 2022. doi: 10.1016/j.actatropica.2022.106447.
- [12] M. Abbas, K. Ali, A. Jamali, K. Ali Memon, and A. Aleem Jamali, “Multinomial Naive Bayes Classification Model for Sentiment Analysis,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 19, no. 3, p. 62, 2019, doi: 10.13140/RG.2.2.30021.40169.
- [13] H. Chen and D. Fu, “An Improved Naïve Bayes Classifier for Large Scale Text,” 2018.
- [14] R. Manikandan and D. R. Sivakumar, “Machine learning algorithms for text-documents classification: A review,” 2018. [Online]. Available: [www.academiciansjournal.com](http://www.academiciansjournal.com)
- [15] B. Charbuty and A. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning,” *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, Mar. 2021, doi: 10.38094/jastt20165.
- [16] M. A. Ahmed, A. Rizaner, and H. U. Ali, “A novel decision tree classification based on post-pruning with Bayes minimum risk,” *PLoS One*, vol. 13, no. 4, Apr. 2018, doi: 10.1371/journal.pone.0194168.
- [17] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, “A comparison of random forest variable selection methods for classification prediction modeling,” *Expert Syst Appl*, vol. 134, pp. 93–101, Nov. 2019, doi: 10.1016/j.eswa.2019.05.028.
- [18] S. Fransiska and A. Irham Gufroni, “Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method,” *Scientific Journal of Informatics*, vol. 7, no. 2, pp. 2407–7658, 2020, [Online]. Available: <http://journal.unnes.ac.id/nju/index.php/sji>
- [19] M. Khoirul, U. Hayati, and O. Nurdiawan, “ANALISIS SENTIMEN APLIKASI BRIMO PADA ULASAN PENGGUNA DI GOOGLE PLAY MENGGUNAKAN ALGORITMA NAIVE BAYES,” 2023.
- [20] M. Farid, S. Wibowo, N. F. Puspitasari, and B. Satya, “PENERAPAN DATA MINING DAN ALGORITMA NAÏVE BAYES UNTUK PEMILIHAN KONSENTRASI MAHASISWA MENGGUNAKAN METODE KLASIFIKASI,” 2022.
- [21] R. Budi Widodo, *Machine Learning Metode k-Nearest Neighbors Klasifikasi Angka Bahasa Isyarat*, I., vol. I. 2022.