

Analisis Perbandingan Algoritma Naive Bayes dan Support Vector Machine dengan Pendekatan TF-IDF Sebagai Klasifikasi Perintah Suara

Faisal Syarifuddin*, Dewi Kusumaningsih

Teknologi Informasi, Magister Ilmu Komputer, Universitas Budi Luhur, Jakarta Selatan, Indonesia

Email: ¹*2411601095@student.budiluhur.ac.id, ²dewi.kusumaningsih@budiluhur.ac.id

Email Penulis Korespondensi: 2411601095@student.budiluhur.ac.id

Submitted: 27/03/2025; Accepted: 31/05/2025; Published: 01/06/2025

Abstrak—Penelitian ini mengevaluasi kinerja dua algoritma klasifikasi, yaitu Naive Bayes dan Support Vector Machine (SVM), dalam mengidentifikasi perintah suara pada aplikasi keuangan bagi penyandang tunanetra. Data yang digunakan telah melalui proses preprocessing meliputi tokenisasi, stemming, dan penghapusan stopword, serta diekstraksi menggunakan metode TF-IDF. Model dilatih menggunakan skema pembagian data 80% untuk pelatihan dan 20% untuk pengujian, lalu dievaluasi berdasarkan akurasi, precision, recall, dan F1-score. Hasil pengujian menunjukkan bahwa kedua model mencapai tingkat akurasi yang sangat tinggi, dengan Naive Bayes memperoleh akurasi 98.6% dan SVM mencapai 98.4%. Keduanya menunjukkan precision, recall, dan F1-score yang tinggi di setiap kategori perintah suara, dengan nilai tertinggi pada kategori "Pembayaran QRIS" yang mencapai precision dan recall 1.00. Analisis confusion matrix menunjukkan bahwa kesalahan klasifikasi terjadi dalam jumlah minimal. Penelitian ini juga menunjukkan bahwa TF-IDF sebagai teknik ekstraksi fitur efektif dalam meningkatkan akurasi pengenalan suara dengan memberikan bobot lebih pada kata-kata yang relevan dan jarang muncul dalam dataset, yang membantu model untuk lebih fokus pada informasi yang paling penting. Dengan hasil tersebut, maka kedua algoritma terbukti efektif dalam mengenali perintah suara. Namun, Naive Bayes sedikit lebih unggul dalam akurasi, sehingga lebih direkomendasikan untuk aplikasi berbasis perintah suara dalam sistem keuangan digital. Temuan ini mendukung pengembangan teknologi yang lebih inklusif dan aksesibel bagi penyandang tunanetra.

Kata Kunci: Naive Bayes; Support Vector Machine; TF-IDF; Pengenalan Suara; Penyandang Tunanetra.

Abstract—This study evaluates the performance of two classification algorithms, namely Naive Bayes and Support Vector Machine (SVM), in identifying voice commands in financial applications for the blind. The data used has gone through a preprocessing process including tokenization, stemming, and stopword removal, and was extracted using the TF-IDF method. The models were trained using a data sharing scheme of 80% for training and 20% for testing, then evaluated based on accuracy, precision, recall, and F1-score. The test results show that both models achieve a very high level of accuracy, with Naive Bayes achieving an accuracy of 98.6% and SVM reaching 98.4%. Both show high precision, recall, and F1-score in each voice command category, with the highest value in the "QRIS Payment" category which achieved a precision and recall of 1.00. Confusion matrix analysis shows that classification errors occur in minimal amounts. This study also shows that TF-IDF as a feature extraction technique is effective in improving speech recognition accuracy by giving more weight to relevant and rarely appearing words in the dataset, which helps the model to focus more on the most important information. With these results, both algorithms are proven to be effective in recognizing voice commands. However, Naive Bayes is slightly superior in accuracy, so it is more recommended for voice-based applications in digital financial systems. These findings support the development of more inclusive and accessible technology for the visually impaired.

Keywords: Naive Bayes; Support Vector Machine; TF-IDF; Speech Recognition; Blind People.

1. PENDAHULUAN

Pengenalan perintah suara (speech recognition) telah berkembang menjadi salah satu teknologi yang paling signifikan dalam bidang Natural Language Processing (NLP) dalam beberapa tahun terakhir [1]. Teknologi ini memungkinkan pengguna untuk berinteraksi dengan perangkat atau aplikasi hanya menggunakan suara, yang secara efektif menggantikan antarmuka berbasis grafis yang lebih tradisional. Dalam konteks perangkat digital dan layanan berbasis internet, antarmuka grafis bergantung pada penglihatan sebagai medium utama untuk interaksi [2]. Hal ini menciptakan hambatan besar bagi kelompok penyandang disabilitas, khususnya penyandang tunanetra yang mana dalam penelitian ini sebagai studi kasus, terlebih tidak dapat memanfaatkan perangkat berbasis tampilan visual. Pengenalan suara, dengan kemampuannya untuk menginterpretasikan perintah suara pengguna, berpotensi membuka akses yang lebih luas bagi kelompok penyandang tunanetra untuk mengakses layanan digital secara lebih mandiri.

Salah satu aplikasi paling signifikan dari teknologi pengenalan suara adalah dalam aplikasi keuangan digital, dimana pengguna dapat melakukan transaksi atau memantau keuangan pribadi tanpa perlu bergantung pada antarmuka visual. Ini tidak hanya memberikan kemudahan bagi penyandang tunanetra, tetapi juga dapat memperkecil kesenjangan aksesibilitas yang selama ini ada di dunia digital. Namun, meskipun teknologi pengenalan suara memiliki potensi yang besar, penerapannya dalam sektor layanan digital, terutama dalam aplikasi keuangan yang membutuhkan akurasi dan keamanan tinggi, masih menghadapi sejumlah tantangan besar. Tantangan utama yang dihadapi adalah akurasi dan keandalan sistem dalam mengenali perintah suara, yang sangat penting untuk menjaga keamanan transaksi dan integritas data keuangan pengguna.

Akurasi dalam pengenalan perintah suara sendiri dapat dipengaruhi oleh berbagai faktor, seperti aksen, intonasi, gangguan suara latar belakang, serta variasi dalam cara berbicara dan pengucapan setiap individu [3]. Dalam konteks aplikasi keuangan, kesalahan dalam pengenalan suara dapat berakibat fatal, karena bisa menyebabkan kesalahan dalam melakukan transaksi atau pemrosesan data penting yang terkait dengan uang dan informasi pribadi.

Oleh karena itu, untuk memastikan tingkat keandalan yang tinggi, diperlukan algoritma klasifikasi yang dapat mengidentifikasi perintah suara dengan tingkat akurasi yang tinggi dan toleransi terhadap variasi bahasa yang luas.

Untuk meningkatkan sebuah akurasi dalam pengenalan suara, pemilihan algoritma yang tepat menjadi faktor kunci [4]. Beberapa algoritma telah terbukti efektif dalam mengklasifikasikan data teks dan suara, dua di antaranya yang paling sering digunakan dalam pemrosesan bahasa alami adalah Naive Bayes (NB) dan Support Vector Machine (SVM) [5].

Naive Bayes, yang berbasis pada prinsip probabilitas, adalah algoritma yang sangat populer dalam aplikasi klasifikasi teks [6].Keunggulan utama dari Naive Bayes terletak pada kesederhanaannya, kemampuannya dalam mengolah data dalam jumlah besar, serta kecepatannya dalam proses komputasi. Meskipun lebih sederhana dibandingkan dengan algoritma lainnya, Naive Bayes tetap dapat memberikan hasil yang sangat efektif dalam pengenalan suara, terutama ketika data yang diproses relatif terstruktur dan dapat dipisahkan dengan jelas antara kelas-kelas tertentu [7].

Di sisi lain, Support Vector Machine (SVM) adalah algoritma yang lebih kompleks dan sangat efektif dalam menangani data yang lebih kompleks dan tidak selalu terpisah dengan jelas [8]. SVM bekerja dengan cara mencari hyperplane terbaik yang dapat memisahkan dua kelas data dengan margin terbesar, yang membuatnya sangat cocok untuk masalah klasifikasi dengan data yang memiliki perbedaan yang tidak selalu linier. Dalam pengenalan suara, SVM dapat memberikan hasil yang sangat baik, terutama ketika perintah suara yang diberikan memiliki variasi yang luas dan data membutuhkan pemisahan yang lebih rumit.

Pada studi pengenalan suara, pendekatan TF-IDF (Term Frequency-Inverse Document Frequency) banyak diaplikasikan di berbagai bidang dan studi kasus. TF-IDF sendiri merupakan metode statistik yang mengukur pentingnya kata atau frasa dalam suatu dokumen dalam konteks koleksi dokumen lainnya. Dalam konteks pengenalan suara, TF-IDF dapat digunakan untuk mengkonversi perintah suara yang diucapkan oleh pengguna menjadi representasi numerik yang lebih dapat dipahami oleh algoritma klasifikasi. Dengan pendekatan ini, algoritma seperti Naive Bayes dan SVM dapat fokus pada kata-kata kunci yang paling relevan dalam perintah suara, meningkatkan akurasi dan efektivitas pengenalan suara [9].

Beberapa penelitian terdahulu menunjukkan bahwa teknologi berbasis suara memiliki potensi besar dalam meningkatkan pengalaman pengguna, terutama untuk penyandang tunanetra. mengungkapkan bahwa sistem berbasis suara mempermudah penyandang tunanetra dalam mengakses informasi digital, yang sebelumnya sulit diakses dengan antarmuka berbasis grafis. Penelitian lain oleh Hafidzah dkk. (2024) juga menemukan bahwa pengenalan perintah suara memiliki potensi besar dalam meningkatkan interaksi dengan sistem berbasis teknologi, memberikan mereka lebih banyak kontrol dan akses terhadap aplikasi digital [10]. Namun, meskipun manfaat pengenalan suara untuk penyandang tunanetra telah terbukti, sebagian besar penelitian ini lebih fokus pada penerapan teknologi ini secara umum dan belum mengeksplorasi penerapannya dalam konteks yang lebih spesifik, seperti aplikasi keuangan, yang memerlukan tingkat akurasi dan keamanan yang sangat tinggi.

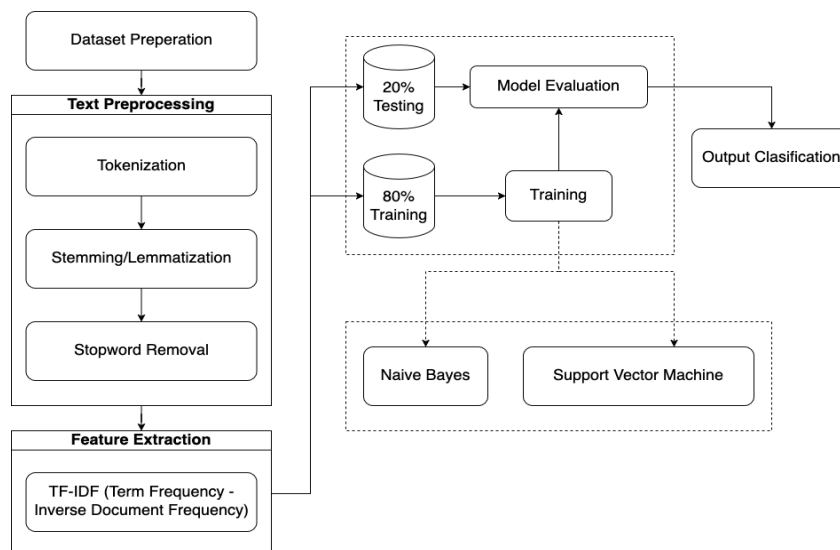
Selain itu, beberapa penelitian juga telah menguji penggunaan Naive Bayes dan SVM dalam pengolahan bahasa alami dan pengenalan suara. Hafidzah dkk. (2024), menemukan bahwa SVM sangat efektif dalam pengenalan suara dalam bahasa alami, mengungguli algoritma lainnya dalam hal akurasi. Di sisi lain, hasil penelitian dari Saputra & Parjito pada tahun 2025 menunjukkan bahwa Naive Bayes memiliki efisiensi tinggi dalam klasifikasi teks, meskipun lebih sederhana dibandingkan dengan SVM [10] [11]. Meskipun banyak penelitian telah menguji kedua algoritma ini dalam pengenalan suara, sebagian besar dari mereka tidak secara langsung membandingkan keduanya dalam aplikasi yang lebih kompleks seperti aplikasi keuangan berbasis suara.

Berdasarkan latar belakang yang sudah di uraikan, penelitian ini bertujuan untuk menganalisis dan membandingkan efektivitas Naive Bayes dan Support Vector Machine (SVM) dalam mengklasifikasikan perintah suara dengan menggunakan pendekatan TF-IDF, khususnya dalam konteks aplikasi keuangan digital yang memerlukan tingkat akurasi dan keamanan yang sangat tinggi. Penelitian ini diharapkan dapat memberikan wawasan baru dalam pemilihan algoritma yang tepat untuk pengenalan suara berbasis teknologi, serta memberikan kontribusi yang signifikan bagi penyandang tunanetra untuk mengakses layanan keuangan digital secara lebih mandiri dan efisien.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini berfokus pada sebuah Analisis Perbandingan Algoritma Naive Bayes dan Support Vector Machine (SVM) dengan Pendekatan TF-IDF Sebagai Klasifikasi Perintah Suara. Melalui beberapa tahap yang saling berkaitan, yang dimulai dengan tahap persiapan dataset, preprocessing teks, hingga evaluasi model. Setiap tahap memiliki tujuan khusus yang akan mendukung pencapaian hasil penelitian yang maksimal. Secara keseluruhan, metodologi ini mengadopsi pendekatan eksperimental dengan studi kasus yang penyandang tunanetra sebagai sampel utama. Gambar 1 berikut merupakan alur dari tahapan penelitian yang dilakukan.



Gambar 1. Alur Tahapan Penelitian

2.2 Persiapan Dataset

Dataset Preparation Data yang digunakan dalam penelitian ini adalah kumpulan perintah suara yang berhubungan dengan transaksi keuangan. Dataset ini dapat mencakup berbagai perintah seperti "Cek Saldo & Mutasi", "Transfer", "Pembayaran QRIS", dan lain-lain. Data ini diperoleh dari rekaman suara yang diucapkan oleh penyandang tunanetra dan pengguna lainnya yang terlibat dalam eksperimen.

2.3 Preprocessing Teks

Preprocessing teks adalah tahap yang sangat penting untuk mempersiapkan data agar dapat diproses oleh model machine learning [12]. Langkah-langkahnya meliputi:

- Tokenization*: Tokenisasi memecah kalimat atau perintah suara menjadi token atau kata-kata individu. Hal ini memungkinkan model untuk bekerja dengan unit data yang lebih kecil dan lebih mudah diproses.
- Stemming/Lemmatization*: Proses ini mengubah kata-kata menjadi bentuk dasarnya (root form). Stemming menghapus akhiran kata untuk mendapatkan bentuk dasar, sementara lemmatization mempertimbangkan konteks untuk menghasilkan kata dasar yang lebih bermakna.
- Stopword Removal*: Pada langkah ini, kata-kata yang tidak memberikan informasi penting (seperti "dan", "atau", "untuk") akan dihapus dari teks, karena mereka dapat mengganggu akurasi model. Menggunakan daftar kata-kata stopwords yang telah dikenal atau menggunakan filter berdasarkan frekuensi kata.

2.4 Ekstraksi Fitur

Setelah preprocessing, fitur-fitur yang relevan dari data teks diekstraksi untuk digunakan oleh model machine learning. Salah satu teknik yang digunakan adalah *TF-IDF* (*Term Frequency - Inverse Document Frequency*) yang mana metode ini digunakan untuk mengukur seberapa penting suatu kata dalam dokumen relatif terhadap keseluruhan kumpulan dokumen [19]. Ini membantu untuk menilai kata-kata mana yang paling relevan dan berpengaruh dalam perintah suara yang diberikan oleh pengguna.

Formula TF-IDF:

$$TF - IDF(t, d) = TF(t, d) \times \log \left(\frac{N}{DF(t)} \right) \quad (1)$$

Dalam metode TF-IDF, $TF(t, d)$ menunjukkan frekuensi kemunculan kata t dalam dokumen d , $DF(t)$ adalah jumlah dokumen yang mengandung kata t , dan N merupakan total jumlah dokumen dalam kumpulan. Ketiga komponen ini digunakan untuk menghitung bobot pentingnya suatu kata dalam konteks dokumen tertentu.

2.5 Pembagian Dataset

Dataset dibagi menjadi dua bagian untuk memastikan bahwa model dapat diuji secara adil, 80% data digunakan untuk melatih model, dimana data ini akan diproses untuk belajar mengenali pola-pola dalam perintah suara. Sedangkan 20% data yang tersisa digunakan untuk menguji akurasi model setelah pelatihan [13].

2.6 Pelatihan Model

Dalam tahap ini, dua algoritma utama digunakan untuk membangun model pengenalan suara, yaitu *Naive Bayes* dan *Support Vector Machine* (*SVM*). Kedua algoritma ini dipilih karena telah terbukti efektif dalam tugas klasifikasi teks. *Naive Bayes* sendiri merupakan Algoritma yang menggunakan teori probabilitas untuk mengklasifikasikan data

berdasarkan kemungkinan kata-kata yang muncul dalam perintah suara [14]. Ini mengasumsikan independensi antar fitur, sehingga sangat cepat dan efisien dalam klasifikasi teks. Rumus yang digunakan untuk *Multinomial Naïve Bayes* dalam menghitung probabilitas suatu kategori C berdasarkan fitur X ditunjukkan pada Persamaan 2 yang merupakan *Formula Teorema Bayes*, serta dijabarkan lebih lanjut pada Persamaan 3 sebagai Rumus *Probabilitas Naïve Bayes*.

$$P(C|X) = \frac{P(C|X) \cdot P(C)}{P(X)} \quad (2)$$

Dalam rumus *Formula Bayes' Theorem*, $P(C|X)$ disebut sebagai probabilitas posterior, yaitu probabilitas bahwa data termasuk ke dalam kelas C setelah fitur X diamati. $P(X|C)$ merupakan probabilitas likelihood, yakni peluang kemunculan fitur X jika diketahui data berasal dari kelas C . Sementara itu, $P(C)$ adalah probabilitas prior, yaitu probabilitas awal terjadinya kelas C sebelum fitur X diamati. Adapun $P(X)$ disebut evidence, yaitu probabilitas kemunculan fitur X secara keseluruhan tanpa memperhatikan kelas tertentu.

$$P(C|X) \propto P(C) \prod_{i=1}^n P(x_i|C) \quad (3)$$

Dalam Rumus Probabilitas Naïve Bayes, $P(C|X)$ adalah probabilitas posterior, yaitu peluang bahwa data termasuk ke dalam kelas C setelah fitur X diamati. Simbol \propto menyatakan bahwa nilai $P(C|X)$ bersifat proporsional terhadap hasil perkalian di sisi kanan persamaan. $P(C)$ merupakan probabilitas awal (prior) bahwa data termasuk ke dalam kelas C sebelum fitur diamati. Operator perkalian $\prod_{i=1}^n$ menunjukkan bahwa probabilitas dihitung untuk seluruh fitur dari $i=1$ hingga n . Adapun $P(x_i|C)$ adalah probabilitas bahwa fitur ke- i muncul pada kelas C , dan n menyatakan jumlah total fitur yang diamati dalam data tersebut.

Sementara *Support Vector Machine (SVM)* adalah algoritma supervised learning yang digunakan untuk mengklasifikasikan data dengan mencari hyperplane terbaik yang memisahkan kelas-kelas dalam suatu dataset. Fungsi keputusan dalam SVM dapat dinyatakan dengan rumus seperti yang dapat dilihat pada Persamaan 4:

$$f(x) = w \cdot x + b \quad (4)$$

Dalam persamaan fungsi keputusan pada *Hyperplane Support Vector Machine (SVM)*, $f(x)$ merupakan fungsi yang digunakan untuk memisahkan kelas. Komponen w adalah vektor bobot yang menentukan arah hyperplane, sementara x adalah vektor fitur atau input data yang dianalisis. Nilai b berfungsi sebagai bias, yaitu intersepsi terhadap sumbu yang mengatur posisi hyperplane dalam ruang fitur.

Untuk menentukan nilai w dan b , SVM memecahkan masalah optimasi dengan rumus berikut: $\min_{w,b} \frac{1}{2} \|w\|^2$, sebagaimana ditunjukkan pada Persamaan 5:

$$y_i(w \cdot x_i + b) \geq 1, \forall i \quad (5)$$

Dalam konteks klasifikasi biner pada *Support Vector Machine (SVM)*, y_i merupakan label kelas untuk data ke- i , yang bernilai $+1$ atau -1 . Simbol \propto menyatakan hubungan proporsional, yang berarti suatu nilai sebanding dengan hasil perhitungan di sisi lainnya dalam persamaan.

SVM sangat efektif dalam menangani data yang tidak dapat dipisahkan secara linear dengan menerapkan *Kernel Trick*, sebuah teknik yang memungkinkan pemetaan data ke dalam dimensi yang lebih tinggi sehingga data menjadi lebih mudah dipisahkan. Beberapa jenis kernel yang umum digunakan dalam pemrosesan teks meliputi *Linear Kernel*, *Radial Basis Function (RBF) Kernel*, dan *Polynomial Kernel* [15].

Untuk menentukan tingkat akurasi dari setiap model yang diuji, digunakan metode *cross-validation* dan *confusion matrix*. Akurasi dihitung menggunakan rumus seperti yang ditunjukkan pada Persamaan 6 hingga 8, sehingga diperoleh nilai akurasi dalam bentuk persentase.

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Total\ Prediksi} \times 100\% \quad (6)$$

$$Mean\ Accuracy = \frac{\sum Akurasi\ dari\ tiap\ Fold}{Jumlah\ Fold} \quad (7)$$

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

Dalam *confusion matrix*, TP (*True Positive*) menunjukkan jumlah prediksi benar untuk kelas positif, sedangkan TN (*True Negative*) menunjukkan prediksi benar untuk kelas negatif. FP (*False Positive*) adalah prediksi salah yang seharusnya termasuk kelas negatif, sementara FN (*False Negative*) adalah prediksi salah yang seharusnya termasuk kelas positif.

2.7 Evaluasi Model

Setelah model dilatih, tahap evaluasi dilakukan menggunakan data uji untuk menilai seberapa baik model dapat mengklasifikasikan perintah suara yang baru. Metrik evaluasi yang digunakan adalah:

- Akurasi: Persentase perintah suara yang berhasil diklasifikasikan dengan benar.
- Presisi: Mengukur berapa banyak prediksi yang benar dibandingkan dengan total prediksi positif.
- Recall: Mengukur seberapa banyak perintah yang relevan berhasil diklasifikasikan dengan benar.

d. F1-Score: Rata-rata atau keseimbangan antara presisi dan recall.

2.8 Klasifikasi Output

Setelah model dilatih dan dievaluasi, model akan mengklasifikasikan perintah suara yang diberikan oleh pengguna ke dalam kategori yang sesuai, seperti “Cek Saldo & Cek Mutasi”, “Transfer”, dan “Pembayaran QRIS”. Klasifikasi output ini memungkinkan penyandang tunanetra untuk mengoperasikan aplikasi keuangan dengan menggunakan perintah suara secara mandiri.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Preprocessing Teks

Untuk memahami bagaimana sistem memproses masukan teks, tahap awal yang dilakukan adalah tokenisasi, yaitu pemecahan kalimat menjadi unit-unit kata [20]. Proses ini merupakan bagian dari tahapan awal dalam penerapan algoritma pemrosesan bahasa alami (Natural Language Processing), yang digunakan dalam sistem ini.

```

tokenize data:
      hasil_text
0      [mutasu]
1      [mutasi]
2      [cek, mutasi]
3      [cek, mutasu]
4      [Cek, transaksi, terakhir]
...
1245   [KEFOS, aj, buat, transaksi, ibo]
1246   [Pakqi, KROS, biar, cepqt]
1247   [Gunakab, KEFIS, untuk, transaksi, saya]
1248   [Saya, maii, sczn, QITID, buat, bayar]
1249   [Tolibg, sczn, KOFIS, ini]

[1250 rows x 1 columns]

steam data & stopword data:
      hasil_steam
0      [mutasu]
1      [mutasi]
2      [cek, mutasi]
3      [cek, mutasu]
4      [cek, transaksi]
...
1245   [kefos, aj, transaksi, ibo]
1246   [pakqi, kros, biar, cepqt]
1247   [gunakab, kefis, transaksi]
1248   [mai, sczn, qitid, bayar]
1249   [tolibg, sczn, kofis]

[1250 rows x 1 columns]

```

Gambar 2. Hasil Preprocessing Text

Hasil tokenisasi menunjukkan bahwa model mampu memisahkan setiap kata dengan baik, memungkinkan analisis yang lebih terstruktur. Misalnya, perintah "cek mutasi" dipisahkan menjadi ["cek", "mutasi"], sementara perintah yang lebih panjang seperti "cek transaksi terakhir" menjadi ["cek", "transaksi", "terakhir"]. Proses ini memastikan bahwa setiap kata dapat dianalisis secara independen sebelum masuk ke tahap pemrosesan berikutnya.

Setelah tokenisasi, dilakukan tahap stemming dan penghapusan stopword untuk menyederhanakan data. Stemming berfungsi untuk mengubah kata ke bentuk dasar, sedangkan penghapusan stopword bertujuan menghilangkan kata-kata yang tidak memiliki makna signifikan dalam analisis, seperti "untuk" dan "dan". Hasil dari proses ini menunjukkan bahwa kata-kata utama tetap dipertahankan, sementara elemen yang tidak relevan berhasil dihilangkan. Sebagai contoh, perintah "cek transaksi terakhir" setelah stemming dan penghapusan stopword menjadi ["cek", "transaksi"], menunjukkan bahwa hanya kata-kata bermakna yang dipertahankan untuk analisis lebih lanjut.

Data yang telah melalui proses tokenisasi, stemming, dan penghapusan stopword kemudian digunakan dalam tahap ekstraksi fitur, yang berperan dalam mengonversi teks ke dalam representasi numerik agar dapat diproses oleh model machine learning. Dengan pendekatan ini, model dapat mengklasifikasikan perintah suara dengan lebih akurat dan efisien. Hasil penelitian ini menunjukkan bahwa metode Natural Language Processing (NLP) yang diterapkan dalam aplikasi keuangan untuk penyandang tunanetra dapat bekerja secara efektif, meningkatkan pemahaman mesin terhadap perintah suara pengguna, serta memberikan respons yang lebih akurat dalam eksekusi perintah keuangan.

```

↳
      hasil_text      kategori  hasil_steam
0      [mutasu]  Cek Saldo & Cek Mutasi  [mutasu]
1      [mutasi]  Cek Saldo & Cek Mutasi  [mutasi]
2      [cek, mutasi]  Cek Saldo & Cek Mutasi  [cek, mutasi]
3      [cek, mutasu]  Cek Saldo & Cek Mutasi  [cek, mutasu]
4  [Cek, transaksi, terakhir]  Cek Saldo & Cek Mutasi  [cek, transaksi]
    
```

Gambar 3. Lima Baris Pertama DataFrame

Fungsi `df.head()` digunakan untuk menampilkan lima baris pertama dari sebuah DataFrame secara default. hal ini ditunjukkan pada Gambar 3, dan sangat berguna untuk memeriksa struktur data dalam DataFrame sebelum melakukan analisis lebih lanjut.

3.2 Ekstraksi Fitur: TF-IDF

Hasil ekstraksi fitur, metode TF-IDF (Term Frequency - Inverse Document Frequency) digunakan untuk mengukur sejauh mana pentingnya sebuah kata dalam konteks dokumen yang bersangkutan. Metode ini terdiri dari dua komponen utama, yaitu Term Frequency (TF) dan Inverse Document Frequency (IDF). Term Frequency (TF) mengukur seberapa sering suatu kata muncul dalam dokumen tertentu, bisa dilihat pada Gambar 4. Sebagai contoh, kata "mutasu" memiliki frekuensi TF sebesar 1.0, yang berarti kata ini hanya muncul sekali dalam dokumen tersebut, sedangkan kata "cek" memiliki frekuensi 0.5, menunjukkan bahwa kata ini muncul setengah kali dibandingkan dengan kata lainnya dalam dokumen yang sama.

```

↳
      TF (Term Frequency):
      TF
0      {'mutasu': 1.0}
1      {'mutasi': 1.0}
2      {'cek': 0.5, 'mutasi': 0.5}
3      {'cek': 0.5, 'mutasu': 0.5}
4      {'cek': 0.5, 'transaksi': 0.5}
...
1245 {'kefos': 0.25, 'aj': 0.25, 'transaksi': 0.25, ...
1246 {'pakqi': 0.25, 'kros': 0.25, 'biar': 0.25, 'c...
1247 {'gunakab': 0.3333333333333333, 'kefis': 0.333...
1248 {'mai': 0.25, 'sczn': 0.25, 'qitid': 0.25, 'ba...
1249 {'tolibg': 0.3333333333333333, 'sczn': 0.33333...

[1250 rows x 1 columns]
    
```

Gambar 4. Hasil Perhitungan Term Frequency

Setelah itu, dilakukan perhitungan Inverse Document Frequency (IDF), seperti ditunjukkan pada Gambar 5, untuk menilai sejauh mana pentingnya suatu kata dalam keseluruhan koleksi dokumen. Semakin jarang kata tersebut muncul di seluruh dataset, semakin besar bobot IDF yang diberikan pada kata tersebut. Misalnya, kata "transfer" memiliki IDF sebesar 1.9324, yang menunjukkan bahwa kata tersebut relatif jarang muncul dalam dataset, sehingga memberikan kontribusi yang lebih besar pada analisis. Sementara itu, kata "cek" memiliki IDF sebesar 1.8892, yang lebih kecil karena kata ini lebih sering muncul dalam dokumen dibandingkan dengan "transfer".

```

↳
      IDF (Inverse Document Frequency):
      Kata      IDF
264      tmn      7.130899
144      tranzasky      7.130899
146      ttransaksi      7.130899
147      trannsaksi      7.130899
148      transakssi      7.130899
..      ...
5      transfer      1.932402
2      cek      1.889152
16      saldo      1.883875
20      rekening      1.837594
22      bank      1.307853

[528 rows x 2 columns]
    
```

Gambar 5. Hasil Perhitungan Inverse Document Frequency

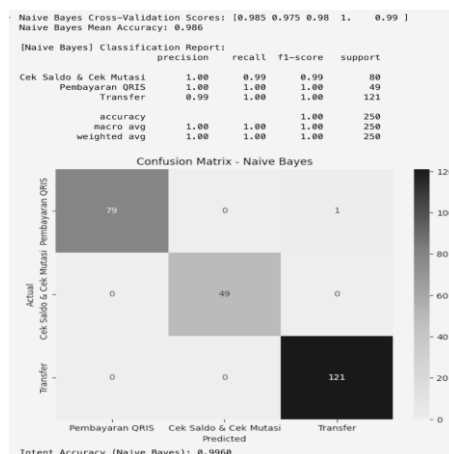
Dengan menggunakan metode TF-IDF, fitur ini secara efektif memberikan bobot lebih pada kata-kata yang dianggap lebih informatif dan jarang digunakan, sehingga memungkinkan model untuk fokus pada kata-kata yang paling relevan dalam menganalisis perintah suara.

3.3 Pembagian Data

Pembagian data dilakukan dengan membagi dataset menjadi dua bagian utama, yaitu 80% untuk pelatihan dan 20% untuk pengujian. Data yang digunakan untuk pelatihan ini memungkinkan model untuk belajar dan mengidentifikasi pola dalam data, yang kemudian digunakan untuk membuat prediksi. Sementara itu, 20% data yang tersisa digunakan untuk menguji kinerja model setelah pelatihan, dengan tujuan memastikan bahwa model tidak hanya menghafal data pelatihan (overfitting), tetapi juga dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Pendekatan ini mengikuti praktik yang umum digunakan dalam berbagai penelitian machine learning, seperti yang dilakukan oleh [16], yang menyarankan penggunaan pembagian data yang seimbang untuk melatih dan menguji model guna menghindari bias dan memberikan evaluasi yang lebih akurat [16]. Dengan pembagian ini, penelitian dapat memastikan bahwa model yang dikembangkan memiliki performa yang baik pada data baru dan dapat diandalkan dalam aplikasi dunia nyata dalam konteks ini adalah Aplikasi Keuangan Bagi Penyandang Tunanetra.

3.4 Pelatihan dan Evaluasi Model

Hasil pelatihan, dua algoritma yang diuji dalam penelitian ini adalah Naive Bayes dan Support Vector Machine (SVM). Kedua model ini dilatih menggunakan data yang telah diproses dengan metode TF-IDF, yang sebelumnya telah menjalani tahapan tokenisasi, stemming, dan penghapusan stopwords. Setelah pelatihan selesai, hasil dari kedua model dievaluasi menggunakan beberapa metrik kinerja, seperti akurasi, precision, recall, dan F1-score untuk mengukur sejauh mana kemampuan model dalam mengklasifikasikan perintah suara dengan benar. Hasil kinerja model untuk kedua algoritma menunjukkan hasil yang sangat baik. Pada Gambar 6 dapat dilihat hasil klasifikasi menggunakan algoritma Naive Bayes.

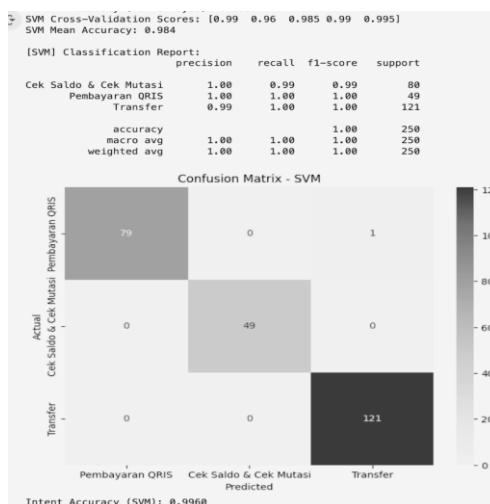


Gambar 6. Laporan Klasifikasi Naive Bayes

Model Naive Bayes memperoleh akurasi 98.6%, dengan nilai precision, recall, dan F1-score yang sangat tinggi untuk masing-masing kelas:

- Cek Saldo & Cek Mutasi: Precision 1.00, Recall 0.99, F1-Score 0.99;
- Pembayaran QRIS: Precision 1.00, Recall 1.00, F1-Score 1.00;
- Transfer: Precision 0.99, Recall 1.00, F1-Score 1.00.

Pada Gambar 7 ditampilkan hasil klasifikasi menggunakan algoritma Support Vector Machine (SVM).



Gambar 7. Laporan Klasifikasi Support Vector Machine

Sementara itu, model Support Vector Machine (SVM) juga menunjukkan kinerja yang sangat baik dengan akurasi 98.4%, dan nilai precision, recall, serta F1-score yang hampir identik dengan Naive Bayes:

- Cek Saldo & Cek Mutasi: Precision 1.00, Recall 0.99, F1-Score 0.99
- Pembayaran QRIS: Precision 1.00, Recall 1.00, F1-Score 1.00
- Transfer: Precision 0.99, Recall 1.00, F1-Score 1.00

Evaluasi model ini juga dilengkapi dengan analisis *confusion matrix*, yang menunjukkan bahwa sebagian besar prediksi yang dilakukan oleh kedua model berjalan dengan baik. Kedua model ini berhasil mengklasifikasikan perintah suara dengan akurat, meskipun terdapat beberapa kesalahan minor dalam mengklasifikasikan kategori Pembayaran QRIS dan Transfer.

Hasil kedua model, Naive Bayes dan SVM, menunjukkan performa yang sangat baik dengan akurasi lebih dari 98%, membuktikan efektivitas kedua algoritma dalam melakukan pengenalan perintah suara dalam aplikasi keuangan bagi penyandang tunanetra. Dengan hasil yang sangat baik ini, baik Naive Bayes maupun SVM terbukti dapat diandalkan dalam mengklasifikasikan perintah suara ke dalam kategori yang tepat, mendukung pengembangan aplikasi yang lebih inklusif dan aksesibel

3.5 Pengujian Model

Pada Gambar 8 ditampilkan hasil pengujian model pada kategori Cek Rekening, yang menunjukkan performa klasifikasi terhadap perintah-perintah yang berkaitan dengan pengecekan informasi rekening.

```

🔊 Cek Audio: True
Path Audio: audio_data/clean_noise/cekrekening.wav
Hasil Konversi Suara Ke Teks: ['cek rekening']

Teks Versi NB: cek rekening
Kategori: Cek Saldo & Cek Mutasi | Skor: 0.9538
Kategori: Pembayaran QRIS | Skor: 0.0058
Kategori: Transfer | Skor: 0.0404

Teks Versi SVM: cek rekening
Kategori: Cek Saldo & Cek Mutasi | Skor: 2.2665
Kategori: Pembayaran QRIS | Skor: -0.2424
Kategori: Transfer | Skor: 0.8103

```

Gambar 8. Pengujian Model Kategori Cek Rekening

Selanjutnya, Gambar 9 menyajikan hasil pengujian model pada kategori Transfer Uang, yang menunjukkan kemampuan model dalam mengklasifikasikan perintah terkait transaksi pengiriman dana.

```

🔊 Cek Audio: True
Path Audio: audio_data/clean_noise/transferuang.wav
Hasil Konversi Suara Ke Teks: ['transfer uang']

Teks Versi NB: transfer uang
Kategori: Cek Saldo & Cek Mutasi | Skor: 0.0345
Kategori: Pembayaran QRIS | Skor: 0.0058
Kategori: Transfer | Skor: 0.9597

Teks Versi SVM: transfer uang
Kategori: Cek Saldo & Cek Mutasi | Skor: 0.8908
Kategori: Pembayaran QRIS | Skor: -0.2321
Kategori: Transfer | Skor: 2.2452

```

Gambar 9. Pengujian Model Kategori Transfer Uang

Pengujian dilakukan dengan menggunakan beberapa contoh audio perintah suara yang berbeda untuk menguji kemampuan kedua model dalam mengklasifikasikan perintah suara. Salah satu contoh perintah suara yang diuji adalah “cek rekening” dan “transfer uang”. Kedua model, yaitu Naive Bayes dan Support Vector Machine (SVM), memberikan hasil yang konsisten dalam mengklasifikasikan perintah tersebut. Model Naive Bayes mengklasifikasikan perintah suara “cek rekening” ke dalam kategori “Cek Saldo & Mutasi” dengan skor 0.9538 dan untuk perintah suara “transfer uang” diklasifikasikan ke dalam kategori “Transfer” dengan skor 0.9597. Sementara SVM memberikan hasil yang sama, mengklasifikasikan perintah suara “cek rekening” ke dalam kategori “Cek Saldo & Mutasi” dengan skor 2.2665 dan untuk perintah suara “transfer uang” diklasifikasikan ke dalam kategori “Transfer” dengan skor 2.2452. Hasil ini menunjukkan bahwa kedua model mampu mengklasifikasikan perintah suara dengan sangat baik, memberikan bukti lebih lanjut mengenai efektivitas kedua model dalam mengenali dan mengkategorikan perintah suara ke dalam kategori yang tepat.

3.6 Pembahasan

Pada tahap preprocessing teks, serangkaian langkah dilakukan untuk mempersiapkan data agar siap diproses lebih lanjut. Proses pertama adalah tokenisasi, yang bertujuan memisahkan setiap perintah suara menjadi kata-kata terpisah. Contohnya, perintah “cek mutasi” dipisahkan menjadi [“cek”, “mutasi”], dan perintah panjang seperti “cek transaksi terakhir” menjadi [“cek”, “transaksi”, “terakhir”]. Setelah itu, dilakukan stemming untuk mengubah kata-kata menjadi bentuk dasarnya, misalnya “mentransfer” menjadi “transfer”, dan penghapusan stopword untuk menghilangkan kata-

kata tidak signifikan seperti "untuk" dan "dan". Dengan langkah-langkah ini, model dapat fokus pada kata-kata kunci yang lebih bermakna, yang meningkatkan efektivitas analisis teks.

Selanjutnya, proses ekstraksi fitur menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency) diterapkan untuk mengukur pentingnya kata dalam konteks dokumen [17]. Sementara Inverse Document Frequency (IDF) menilai seberapa jarang kata tersebut muncul di seluruh dataset [18]. Kata-kata yang jarang muncul dalam dataset ini, seperti "transfer", akan diberikan bobot yang lebih besar. Metode TF-IDF memungkinkan model untuk lebih fokus pada kata-kata yang lebih informatif dan relevan, yang sangat membantu dalam meningkatkan akurasi model dalam mengklasifikasikan perintah suara [19].

Setelah tahap preprocessing dan ekstraksi fitur, data dibagi menjadi dua bagian utama: 80% untuk pelatihan dan 20% untuk pengujian. Pembagian ini memastikan bahwa model tidak hanya menghafal data pelatihan (menghindari overfitting), tetapi juga mampu menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Pelatihan dilakukan dengan menggunakan dua algoritma, yaitu Naive Bayes dan Support Vector Machine (SVM). Kedua model ini diuji dan menghasilkan performa yang sangat baik, dengan akurasi lebih dari 98%, precision, recall, dan F1-score yang tinggi di setiap kategori perintah suara. Hal ini menunjukkan bahwa kedua algoritma efektif dalam mengenali dan mengklasifikasikan perintah suara dalam aplikasi keuangan untuk penyandang tunanetra.

4. KESIMPULAN

Berdasarkan hasil evaluasi yang dilakukan, kedua algoritma Naive Bayes dan Support Vector Machine (SVM) terbukti sangat efektif dalam mengenali dan mengklasifikasikan perintah suara dalam hal klasifikasi perintah suara. Pada dasarnya kinerja kedua model ini menunjukkan hasil yang sangat baik dengan akurasi lebih dari 98%. Model Naive Bayes mencapai akurasi 98.6%, sementara model SVM mendapatkan akurasi 98.4%. Perbedaan ini menunjukkan bahwa Naive Bayes lebih efektif dalam mengenali dan mengklasifikasikan perintah suara dengan tingkat akurasi yang lebih tinggi, yang mana sangat penting dalam konteks aplikasi keuangan berbasis perintah suara yang membutuhkan hasil yang akurat. Keduanya juga menunjukkan nilai precision, recall, dan F1-score yang sangat tinggi untuk setiap kategori, dengan Cek Saldo & Cek Mutasi mencapai precision dan recall 1.00, Pembayaran QRIS mendapatkan precision dan recall 1.00, dan Transfer dengan precision 0.99 dan recall 1.00, menunjukkan bahwa keduanya memiliki kemampuan klasifikasi yang sangat baik dengan minim kesalahan. Dalam hal preprocessing data, penggunaan TF-IDF untuk ekstraksi fitur, bersama dengan stemming dan penghapusan stopword, terbukti efektif dalam menghasilkan data yang lebih bersih dan terstruktur. Dengan preprocessing yang baik, model dapat bekerja lebih efisien, yang memungkinkan untuk mengenali perintah suara yang diberikan oleh pengguna dengan lebih cepat dan tepat. Secara keseluruhan, Naive Bayes lebih direkomendasikan untuk aplikasi yang mengutamakan sebuah akurasi dalam mengenali perintah suara. Hal ini menjadikan Naive Bayes pilihan yang cukup ideal untuk memastikan pengolahan perintah suara yang optimal dan dapat diandalkan dalam sebuah aplikasi berbasis perintah suara, terutama dalam konteks aplikasi yang memiliki tingkat kompleksitas tinggi seperti sistem keuangan digital.

REFERENCES

- [1] R. C. Tarumingkeng, *Natural Language Processing (NLP)*. 2024.
- [2] P. Kollamudi, B. Koduru, B. Poranki, and M. P. Golla, 'Smart virtual assistant', *Mater Today Proc*, 2021, doi: <https://doi.org/10.1016/j.matpr.2021.07.303>.
- [3] F. E. B. Setyawan, *Pendekatan pelayanan kesehatan dokter keluarga (pendekatan holistik komprehensif)*. Zifatama Jawa, 2019.
- [4] I. Gavati, A. Griparis, and S. Segarceanu, 'Natural language processing in assistive technologies', *Romanian Journal of Technical Sciences - Applied Mechanics*, vol. 68, no. 2, pp. 129–140, 2023.
- [5] A. A. Ariffin, A. F. Ibrahim, S. Hasan, and R. Latip, 'An Efficient Virtual Machine Scheduling Algorithm To Minimize Makespan And Maximize Profit Using Hyper Heuristic Approach', *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 1, pp. 206–216, 2019.
- [6] R. L. Simanjuntak, T. R. Siagian, V. Anggriani, and A. Arnita, 'Analisis Sentimen Ulasan Pada Aplikasi E-Commerce Shopee Dengan Menggunakan Algoritma Naive Bayes', *Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 3, no. 3, pp. 23–39, 2023.
- [7] R. Ade and P. R. Deshmukh, 'Efficient Knowledge Transformation System Using Pair of Classifiers for Prediction of Students Career Choice', *Procedia Comput Sci*, vol. 46, pp. 176–183, 2015.
- [8] P. A. Pravesy, 'Studi Perbandingan Metode Support Vector Machine, Random Forest, Dan Convolutional Neural Network Untuk Klasifikasi Penyakit Kulit', *Jurnal Kecerdasan Buatan dan Teknologi Informasi*, vol. 4, no. 1, pp. 70–76, 2025.
- [9] S. Deshmukh, P. S. Rede, and S. Iyer, 'Voice-Enabled Vision For The Visually Disabled', in *International Conference on Advances in Computing, Communication, and Control (ICAC3)*, 2021.
- [10] P. Hafidzah, S. Maryani, B. Y. Ihsani, N. Nurmiwati, E. Erwin, and A. K. Niswariyana, 'Penerapan Deep Learning dalam Menganalisis Sentimen di Media Sosial', in *Seminar Nasional Paedagogia*, 2024, pp. 328–339.
- [11] M. R. Saputra and P. Parjito, 'Analisis Sentimen Twitter Terhadap Konflik Di Papua Menggunakan Perbandingan Naive Bayes Dan Svm', *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 2, pp. 1197–1208, 2025.
- [12] A. Alfando and R. Hayami, 'Klasifikasi Teks Berita Berbahasa Indonesia Menggunakan Machine Learning Dan Deep Learning: Studi Literatur', *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 1, pp. 681–686, 2023.



- [13] N. Nurwanda, N. Suarna, and W. Prihartono, 'Penerapan Nlp (Natural Language Processing) Dalam Analisis Sentimen Pengguna Telegram Di Playstore', *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 2, pp. 1841–1846, 2024.
- [14] S. Wijaya and S. Hariyanto, 'Perancangan Chatbot Dengan Metode Natural Language Processing (Nlp) Dalam Proses Booking Order Di Carwash Park Tangcity', *Akselerator: Jurnal Sains Terapan dan Teknologi*, vol. 5, no. 1, pp. 33–45, 2024.
- [15] S. Rabbani, D. Safitri, N. Rahmadhani, A. A. F. Sani, and M. K. Anam, 'Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM: Comparative Evaluation of SVM Kernels for Sentiment Classification in Fuel Price Increase Analysis', *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 153–160, 2023.
- [16] S. B. Kotsiantis, 'Supervised Machine Learning: A Review of Classification Techniques', *Informatica University of Peloponnese*, pp. 260–265, 2007.
- [17] L. Suryani and K. Edy, 'Pengembangan Aplikasi "Lost & Found" Berbasis Android Dengan Menggunakan Metode Term Frequency–Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity', *Electro Luceat*, vol. 6, no. 2, pp. 190–204, 2020.
- [18] K. Huda, S. D. Pohan, and Y. Herlina, 'Penerapan Pembobotan Term Frequency-Inverse Document Frequency Dan Algoritma K-Nearest Neighbor Untuk Analisis Ulasan Hotel Di Situs Tripadvisor', *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, 2024.
- [19] R. R. Putra, N. A. Putri, and A. D. Putra, 'Teknik Cosine Similarity dan TF-IDF dalam Analisis Data', *Serasi Media Teknologi*, 2024.
- [20] E. M. W. Runturamby, V. P. Rantung, and K. Santa, 'Peringkasan Teks Otomatis Berita Online Komisi Pemilihan Umum Menggunakan Algoritma K-Means Clustering', *Prosiding SISFOTEK*, vol. 8, no. 1, pp. 299–312, 2024.