

Analisis Sentimen Ulasan Aplikasi M-Paspor Menggunakan TF-IDF dan Support Vector Machine

Ni Luh Putu Happy Nirmala^{a1}, Ida Bagus Gede Dwidasmara^{a2}

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
¹nirmala.2208561015@student.unud.ac.id
²dwidasmara@unud.ac.id

Abstract

In the era of globalization, getting a passport has become an essential requirement as an official document for international travel. The Directorate General of Immigration introduced M-Paspor, a new application with more than 1 million downloads and 29 thousand reviews on the Google Play Store. This research aims to analyze the sentiments of Indonesian people using the M-Paspor application using the Support Vector Machine and TF-IDF methods for weighting, as well as evaluating the model with K-fold Cross Validation in Google Colab with the Python programming language. The SVM method was chosen because of its ability to achieve high classification accuracy, while feature extraction was carried out using the TF-IDF method to determine the weight of the words in the review. The dataset consists of 3,000 review data, with 1,500 negative sentiment review data and 1,500 positive sentiment review data, which underwent a series of preprocessing stages, namely noise removal, case folding, tokenization, normalization, stopwords removal, and stemming. The SVM model used to analyze and get the best combination of parameters C:1, gamma:scale, with the kernel:rbf. Evaluation of the model with K-Fold Cross Validation shows an average accuracy of 83.62%, precision of 84.7%, recall of 83.65%, and F1 score of 83.51%.

Keywords: Sentiment Analysis, M-Paspor, TF-IDF, Support Vector Machine, K-Fold Cross Validation

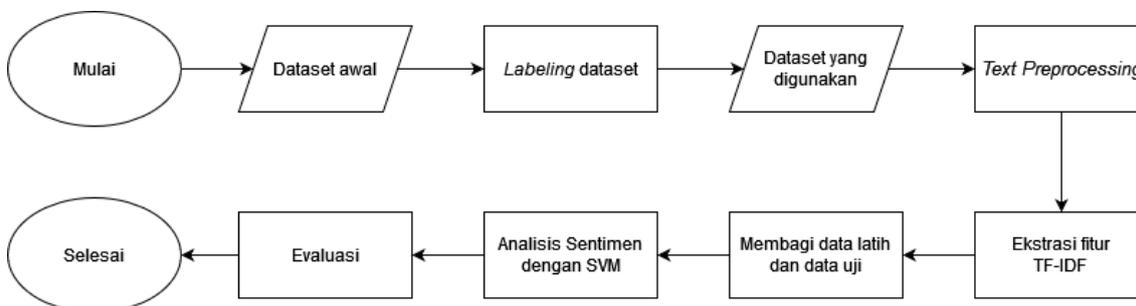
1. Pendahuluan

Pada era globalisasi ini, mobilitas manusia menjadi semakin penting. Warga Indonesia tidak terkecuali dari tren ini, semakin banyak yang tertarik untuk berkunjung ke luar negeri. Kebutuhan akan paspor sebagai dokumen resmi untuk melakukan perjalanan internasional menjadi suatu keharusan. Paspor dalam undang-undang diatur pada Undang-Undang Nomor 6 Tahun 2011 tentang Keimigrasian, hingga Peraturan Menteri Hukum dan HAM Nomor 8 Tahun 2014 [1] Menurut rencana Kementerian Hukum dan HAM pada tahun 2020-2024, salah satu fokus utama adalah meningkatkan layanan keimigrasian dengan menggunakan teknologi [2]. Oleh karena itu, Direktorat Jenderal Imigrasi memperkenalkan M-Paspor, sebuah aplikasi baru yang memudahkan proses pembuatan paspor karena dapat digunakan untuk mendaftar permohonan paspor Indonesia baru atau penggantian paspor. Aplikasi ini sudah diunduh oleh 1 juta lebih masyarakat Indonesia dengan total 29 ribu ulasan di Google Play Store. Apabila proses evaluasi dari umpan balik yang diberikan oleh pelanggan dilakukan secara manual maka tentu saja akan menjadi tidak efektif karena ulasan yang diberikan oleh pelanggan terhitung sangat banyak dengan variasi komentar yang berbeda-beda. Maka dari itu diperlukan suatu metode untuk menganalisis hal tersebut dengan lebih efektif dan akurat, salah satunya adalah dengan analisis sentimen. Analisis sentimen adalah sebuah penelitian komputasional yang berkaitan dengan emosi yang direpresentasikan ke dalam sebuah teks. [3] Sebelumnya, sebuah penelitian terkait analisis sentimen pernah dilakukan yaitu dengan menganalisis sebuah ulasan hotel berbahasa Indonesia, yang juga menggunakan metode support vector machine (SVM) serta TF-IDF [4]. Pada penelitian tersebut dilakukan tahap normalisasi dengan cara memperbanyak kosa kata yang ada pada kamus slang Bahasa Indonesia lalu memperoleh akurasi metode SVM dari eksperimen yang dilakukan yaitu sebesar 85%. Tujuan diadakannya penelitian ini tidak lain

adalah untuk menganalisis sentimen masyarakat Indonesia pengguna aplikasi M-Paspor yang memberikan ulasan di Google Play Store menggunakan metode (SVM) serta dengan metode TF-IDF untuk pembobotannya. Dengan adanya analisis sentiment ini, ulasan akan dibagi menjadi ulasan dengan sentimen positif dan sentimen negatif. Beberapa penelitian sebelumnya yang menggunakan dan membandingkan metode SVM dengan metode lainnya menyatakan bahwasanya Support Vector Machine (SVM) seringkali dipilih karena kemampuannya dalam mencapai akurasi klasifikasi yang tinggi sehingga mengungguli metode klasifikasi lainnya. [5] Namun, sebelum itu harus dilakukan tahap ekstraksi fitur yang merupakan salah satu tahapan penting. Penelitian ini melakukan ekstraksi fitur dengan metode TF-IDF untuk mencari bobot dari setiap kata-kata yang ada. Lalu bobot yang ada akan diletakkan di dalam suatu ruang vektor. Setelah seluruh proses tersebut dilakukan, di akhir akan dilakukan proses evaluasi model dengan K-fold Cross Validation. Penelitian ini dilakukan di Google Colab menggunakan Python serta bantuan beberapa library yang digunakan.

2. Metode Penelitian

2.1 Alur Penelitian

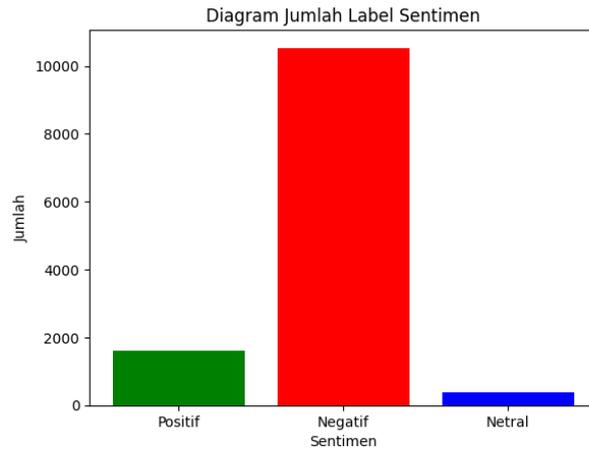


Gambar 1. Alur Penelitian

Alur pada penelitian ini tercakup pada Gambar 1. Langkah awal yang dilakukan adalah mencari dataset ulasan aplikasi M-Paspor yang ada di Google Play Store melalui beberapa situs yang ada. Setelah itu dilakukan labeling dataset untuk mengetahui label dataset yang akan digunakan untuk penelitian. Kemudian proses pengambilan sejumlah dataset yang akan digunakan agar mendapat dataset sebenarnya. Setelah proses tersebut dilakukan tahapan text preprocessing sebelum melalui ekstraksi fitur. Lalu akan dilakukan ekstraksi fitur dengan metode TF-IDF agar data tekstual yang ada selanjutnya dapat dipahami oleh metode analisis sentimen yang akan digunakan. Kemudian dilakukan proses pada dataset tersebut melalui model pembelajaran mesin dengan Support Vector Machine (SVM) untuk mengklasifikasi apakah data tekstual tersebut merupakan analisis sentiment positif atau negatif. Kemudian proses selanjutnya yang akan dilakukan ialah proses evaluasi dengan K-Fold Cross-Validation terhadap analisis sentimen yang telah dilakukan.

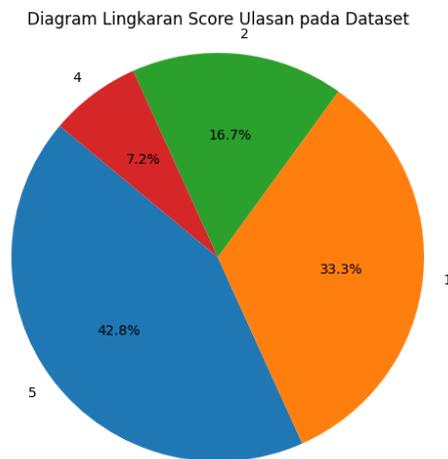
2.2 Pengumpulan Data

Proses pengumpulan dataset review aplikasi M-Paspor di Google Play Store ini menggunakan data primer yang diambil dari situs Kaggle tepatnya pada link berikut [ini](#). Dataset ulasan yang didapat menggunakan Bahasa Indonesia merupakan data ulasan dari tanggal 30 Desember 2021 hingga tanggal 05 Januari 2024. Data ini terdiri dari 12.499 ulasan yang terdiri dari nama pengguna yang memberikan ulasan, rating bintang (score), tanggal memberikan ulasan dan tentu saja isi ulasan yang diberikan. Kemudian dari dataset awal tersebut dilakukan proses labeling data, untuk ulasan dengan bintang 1 dan 2 diberi label negatif, ulasan dengan bintang 3 diberi label netral, serta ulasan dengan bintang 4 dan 5 diberi label positif.



Gambar 2. Diagram Batang Jumlah Label Sentimen pada Dataset Awal

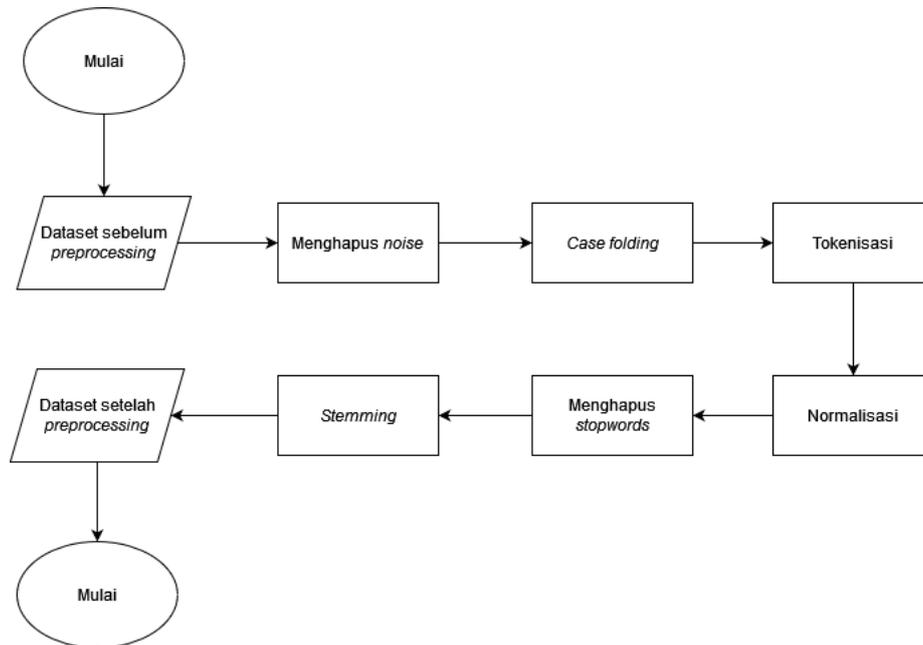
Namun karena setelah dianalisis bahwa ulasan dengan bintang 3 datanya cenderung rancu dengan adanya data yang terkadang netral, positif maupun negatif maka peneliti tidak mengikutsertakan ulasan dengan bintang 3 pada penelitian kali ini dan tidak menggunakan seluruh dataset yang ada, melainkan hanya memilih sejumlah data terbaru saja. Data yang akan dipakai yaitu sejumlah 1000 data ulasan dengan bintang 1, 500 data ulasan dengan bintang 2, 215 data dengan bintang 4 dan 1285 data dengan bintang 5. Persentase score ulasan yang digunakan pada dataset dapat dilihat di diagram lingkaran Gambar 3.



Gambar 3. Diagram Lingkaran Score Ulasan pada Dataset yang Digunakan

Data tersebut berasal dari tanggal 30 Januari 2022 hingga tanggal 05 Januari 2024. Sehingga total data yang dipakai pada penelitian ini yaitu 3.000 data ulasan dengan jumlah ulasan sentimen yang negatif sejumlah 1.500 data dan jumlah ulasan sentimen yang positif sejumlah 1.500 data.

2.3 Text Preprocessing



Gambar 4. Alur Text Preprocessing

Adapun tahap-tahap dari text preprocessing penelitian kali ini dapat direpresentasikan pada Gambar 4. Proses awal yang dilakukan pada dataset adalah menghapus noise misalnya seperti menghilangkan tanda baca, karakter yang bukan teks, tautan, angka dan yang lainnya. Selanjutnya tahap case folding yaitu proses untuk menyamakan teks yang ada pada dataset agar tidak terpengaruh oleh adanya huruf besar maupun huruf kecil, dataset ini akan dibuat lower text seluruhnya. [6] Kemudian proses tokenisasi ialah proses pengolahan teks yang dapat membagi kalimat menjadi sebuah token yang pada kali ini dibagi menjadi kata per kata.[7] Setelah itu dataset melalui proses normalisasi yaitu pengubahan kata-kata yang menggunakan bahasa sehari-hari atau slang yang belum baku agar menjadi kata-kata yang baku dan sesuai aturan kepenulisannya menggunakan bantuan kamus. [8] Lalu akan dilakukan proses penghapusan stopwords pada dataset yang digunakan seperti kata-kata yang sebenarnya tidak penting atau kurang relevan di dalam dataset dan hanya akan menjadikan proses semakin lama sehingga harus dihapus. Setelah tahap-tahapan tersebut barulah dataset dilakukan proses stemming yaitu tahapan agar kata yang mengandung imbuhan awalan maupun akhiran dapat diubah menjadi kata dasar saja. Setelah itu akan dihasilkan sebuah dataset yang telah melalui tahap-tahap preprocessing menggunakan bahasa pemrograman Python.

2.4 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF merupakan singkatan dari Term Frequency-Inverse Document Frequency adalah salah satu metode untuk melakukan ekstraksi fitur dan pembobotan dalam sebuah teks berdasarkan frekuensi munculnya sebuah kata untuk mendeteksi pentingnya kata tersebut di sebuah dataset. Pembobotan kata dengan TF-IDF didapat dari hasil kali Term Frequency dengan hasil Inverse Document Frequency berdasarkan query.[9] Nilai TF sendiri berarti frekuensi dari kemunculan suatu kata di dalam dokumen sehingga jika nilai TFnya tinggi berarti kata tersebut sangat penting di dalam dokumen. Sedangkan nilai IDF adalah invers dari frekuensi dari kemunculan suatu kata dari seluruh dokumen yang ada sehingga jika nilai IDFnya tinggi maka berarti jumlah kemunculan katanya sedikit dan berarti sangat penting. [10] Setelah dataset melalui proses preprocessing selanjutnya akan dilakukan ekstraksi fitur dan pembobotan kata dengan metode TF-IDF. Untuk sebuah kata "t" dalam dokumen "d", maka TF-IDFnya didapat menggunakan rumus seperti di bawah ini:

$$TF(t, d) = \frac{\text{Frekuensi munculnya kata } t \text{ di dokumen } d}{\text{Frekuensi kata seluruhnya di dokumen } d} \quad (1)$$

$$IDF(t) = \log \frac{\text{Frekuensi dokumen}}{\text{Frekuensi dokumen yang terdapat kata } t} \quad (2)$$

$$TF-IDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

2.5 Support Vector Machine (SVM)

SVM adalah singkatan dari Support Vector Machine adalah salah satu algoritma machine learning yang dipakai untuk analisis sentimen. Konsep utama dari SVM adalah mencari pemisah atau hyperplane terbaik yang dapat sebagai pemisah antara dua jenis sentimen data dengan margin maksimum, margin adalah jarak terdekat antara hyperplane dan titik data terdekat dari kedua kelas. SVM berusaha untuk menemukan hyperplane yang paling baik membedakan antara kelas dengan memaksimalkan margin ini, sehingga membuatnya sangat efektif dalam mengklasifikasikan data yang kompleks. Selain itu, SVM juga dapat memproyeksikan data dengan memanfaatkan fungsi kernel sehingga dapat diproyeksikan ke dimensi yang lebih tinggi, yang memungkinkannya untuk menangani kasus-kasus untuk kelas tidak dapat dipisahkan secara linear dalam dimensi yang rendah. Support vector merupakan penyebutan untuk pola yang posisinya terdekat dengan hyperplane. [11]

2.6 Evaluasi Model Dengan K-Fold Cross-Validation

Evaluasi model dengan k-fold cross-validation adalah sebuah metode yang dipakai untuk mengukur seberapa baik model machine learning dalam melakukan prediksi. Dalam k-fold cross-validation, data dipisahkan menjadi k subset yang sama besar. Model dilatih menggunakan k-1 subset dan diuji pada subset yang tersisa. Proses ini diulangi k kali, di mana setiap subset digunakan sebagai data uji secara bergantian. Hasil akhir dari evaluasi ini adalah rata-rata dari nilai-nilai metrik evaluasi (seperti akurasi) dari setiap iterasi k-fold cross-validation. [12]

3. Hasil dan Pembahasan

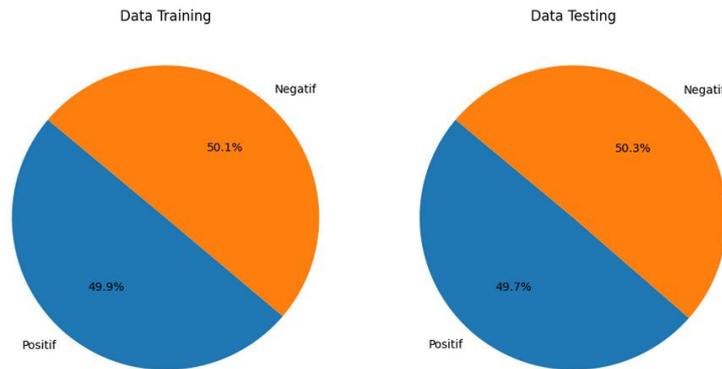
3.1 Text Preprocessing

Langkah awal yang dilakukan terhadap dataset yaitu tahap preprocessing kepada 3.000 data yang ada di dataset. Berikut adalah tahapan preprocessing yang dilakukan terhadap dataset:

Tahap Preprocessing	Data
Data sebelum preprocessing	Hari libur, server libur? aplikasi tidak dpat diakses.. Kedepannya harap diperbaiki
Menghapus noise	Hari libur server libur aplikasi tidak dpat diakses Kedepannya harap diperbaiki
Case folding	hari libur server libur aplikasi tidak dpat diakses kedepannya harap diperbaiki
Tokenisasi	['hari', 'libur', 'server', 'aplikasi', 'tidak', 'dpat', 'diakses', 'kedepannya', 'harap', 'diperbaiki']
Normalisasi	['hari', 'libur', 'server', 'aplikasi', 'tidak', 'dapat', 'diakses', 'kedepannya', 'harap', 'diperbaiki']
Menghapus stopwords	['hari', 'libur', 'server', 'aplikasi', 'diakses', 'kedepannya', 'diperbaiki']
Stemming	['hari', 'libur', 'server', 'aplikasi', 'akses', 'depan', 'perbaik']

Gambar 5. Hasil Tahap Preprocessing

Tahapan text preprocessing di atas dilakukan di Google Colab menggunakan bahasa pemrograman python. Dalam prosesnya penelitian ini dibantu oleh library pandas, nltk dan juga sastrawi. Serta untuk proses normalisasi menggunakan bantuan kamus kamus.txt. Kemudian dataset yang telah melalui tahap preprocessing dibagi menjadi data training serta data testing. Adapun perbandingan data training dan data testing-nya yaitu 8:2 sehingga jumlah data latihnya menjadi 2.400 dan data ujinya menjadi 600. Untuk persentase jumlah label dataset pada data uji dan data latih, terlihat pada Gambar 6.



Gambar 6. Diagram Lingkaran Persentase Sentimen pada Data Training dan Data Testing

Berikut merupakan code pada tahap text processing dari awal hingga akhir:

```
def casefolding(teks):  
    #menghapus noise  
    dataset = re.sub(r'@[A-Za-a0-9]+', ' ', teks)  
    dataset = re.sub(r'#[A-Za-z0-9]+', ' ', teks)  
    dataset = re.sub(r"http\S+", ' ', teks)  
    dataset = re.sub(r'[0-9]+', ' ', teks)  
    dataset = re.sub(r"[-()\"#/@;:<>{}'+~|.!?,_]", " ", teks)  
  
    #case folding  
    teks = teks.lower()  
  
    return teks
```

Gambar 7. Code Tahap Preprocessing Bagian Menghapus Noise dan Case Folding

```
def tokenisasi(teks):  
    teks = word_tokenize(teks)  
  
    return teks
```

Gambar 8. Code Tahap Preprocessing Bagian Tokenisasi

```
def normalisasi(teks):  
    kamusSlang = eval(open("kamus.txt").read())  
    pattern = re.compile(r'\b( ' + '|'.join (kamusSlang.keys())+r')\b')  
    content = []  
    for kata in teks:  
        filterSlang = pattern.sub(lambda x: kamusSlang[x.group()],kata)  
        content.append(filterSlang.lower())  
    teks = content  
    return teks
```

Gambar 9. Code Tahap Preprocessing Bagian Normalisasi

```
stopwords_indonesia = set(stopwords.words('indonesian'))

def removedStopwords(teks):
    tokens_with_no_stopwords = list()

    for word in teks:
        if word not in stopwords_indonesia:
            tokens_with_no_stopwords.append(word)

    return tokens_with_no_stopwords
```

Gambar 10. Code Tahap Preprocessing Bagian Menghapus Stopwords

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stemming(teks):
    return [stemmer.stem(word) for word in teks]
```

Gambar 11. Code Tahap Preprocessing Bagian Stemming

3.2 Ekstraksi Fitur

Tahapan ekstraksi fitur kepada dataset yang sudah menjalani tahap preprocessing adalah dengan TF-IDF. Hasil ekstraksi fitur ini yang akan menjadi input dalam model analisis sentimen dengan metode SVM. Dalam proses ekstraksi fitur ada dua parameter utama yang digunakan yaitu $max_df=0.5$ yang berarti kata yang muncul di lebih dari 50% dari semua dokumen akan diabaikan. Kata-kata ini kemungkinan besar adalah kata umum yang tidak relevan. Selain itu terdapat juga $min_df=2$ yang berarti kata yang muncul di kurang dari 2 dokumen akan diabaikan. Ini membantu menghilangkan kata-kata yang terlalu spesifik yang mungkin tidak memberikan banyak informasi. Kemudian dataset diubah menjadi representasi TF-IDF yang mana hasilnya adalah Setiap kolom akan mewakili fitur yang dihasilkan oleh TF-IDF, dan setiap barisnya akan mewakili satu dokumen.

3.3 Analisis Sentimen

Setelah dataset melalui beberapa tahap proses preprocessing dan ekstraksi fitur pembobotan dengan TF-IDF, akan dilakukan proses analisis sentimen dengan metode Support Vector Machine. Adapun parameter C yang digunakan yaitu 0.1, 1, 10, 100, parameter kernel yang digunakan yaitu kernel linear, rbf, dan poly, sedangkan parameter gamma menggunakan scale dan auto. Pada implementasinya gridsearch digunakan untuk mencari kombinasi hyperparameter yang terbaik. Ternyata kombinasi parameter terbaik yang ditemukan yaitu C yaitu 1, gamma yaitu scale, dengan kernel yaitu rbf. Untuk data latihnya didapat hasil SVM terbaik yaitu mean accuracy 95,29%, precision 95,35%, recall 95,29% dan F1 score 95,29%. Untuk data ujinya didapat hasil SVM terbaik yaitu mean accuracy 84,5%, precision 85,13%, recall 84,5% dan F1 score 84,37%.

3.4 Evaluasi Model dengan K-Fold Cross-Validation

Setelah tahap analisis sentimen berhasil dilakukan pada dataset. Maka evaluasi model dengan K-Fold Cross-Validation telah dapat dilakukan untuk mengevaluasi model dengan lebih baik. Metrik yang dipakai adalah accuracy, precision, recall, serta F1 score. Pada penelitian kali ini didapat hasil mean accuracy 83,62%, precision 84,7%, recall 83,65% dan F1 score 83,51%.

4. Kesimpulan

Penelitian ini memiliki tujuan untuk melakukan analisis sentimen dari ulasan pengguna aplikasi M-Paspor menggunakan metode TF-IDF untuk ekstraksi fitur dan pembobotan, serta Support Vector Machine (SVM) untuk analisis sentimen. Dataset yang terdiri dari 3.000 data ulasan, dengan 1.500 data ulasan sentimen negatif dan 1.500 data ulasan sentimen positif, menjalani serangkaian tahapan preprocessing termasuk penghapusan noise, case folding, tokenisasi, normalisasi, penghapusan stopwords, dan stemming. Fitur-fitur dari ulasan diekstraksi dan diberi bobot dengan metode TF-IDF, kemudian model SVM digunakan untuk menganalisis sentiment dan didapat kombinasi parameter terbaik C yaitu 1, gamma yaitu scale, dengan kernel yaitu rbf. Evaluasi model dilakukan dengan K-Fold Cross Validation, menyatakan akurasi rata-rata sebesar 83,62%, precision sebesar 84,7%, recall sebesar 83,65%, dan F1 score yaitu sebesar 83,51%. Hasil penelitian ini menunjukkan potensi metode TF-IDF dan SVM dalam analisis sentimen ulasan khususnya ulasan aplikasi M-Paspor.

Daftar Pustaka

- [1] A. Irawan, M. Z Jamlean, dan H. Ibrahim, "Manajemen Pelayanan Pembuatan Paspor Pada Kantor Imigrasi Kelas Ii Tpi Merauke," MADANI Jurnal Politik dan Sosial Kemasyarakatan, vol. 14, hlm. 399–415, Agu 2022.
- [2] A. Prameswari, S. Zakaria, dan S. Centia, "Pelayanan Publik Berbasis Electronic Government Melalui Penerapan Aplikasi Mobile Paspor (M-Paspor) Di Kantor Imigrasi Kelas I Tpi Bandung Tahun 2022," Jurnal Administrasi Pemerintahan (JANITRA), vol. 3, hlm. 256–262, Nov 2023.
- [3] F. Gunawan, M. Ali Fauzi, dan P. Pandu Adikara, "Analisis Sentimen Pada Ulasan Aplikasi Mobile Menggunakan Naive Bayes Dan Normalisasi Kata Berbasis Levenshtein Distance (Studi Kasus Aplikasi Bca Mobile)," Malang, Agu 2017.
- [4] V. W. D. Thomas dan F. Rumaisa, "Analisis Sentimen Ulasan Hotel Bahasa Indonesia Menggunakan Support Vector Machine dan TF-IDF," Jurnal Media Informatika Budidarma, vol. 6, no. 3, hlm. 1767–1774, Jul 2022, doi: 10.30865/mib.v6i3.4218.
- [5] P. A. Octaviani, Y. Wilandari, dan D. Ispriyanti, "Penerapan Metode Klasifikasi Support Vector Machine (Svm) Pada Data Akreditasi Sekolah Dasar (Sd) Di Kabupaten Magelang," Jurnal Gaussian, vol. 3, no. 4, hlm. 811–820, 2014, [Daring]. Tersedia pada: <http://ejournal-s1.undip.ac.id/index.php/gaussian>
- [6] D. Alita dan A. Rahman, "Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier," Labuhanratu, 2020.
- [7] E. K. Putri dan T. Setiadi, "Penerapan Text Mining Pada Sistem Klasifikasi Email Spam Menggunakan Naive Bayes," Jurnal Sarjana Teknik Informatika, vol. 2, hlm. 73–83, Okt 2014.
- [8] Y. Arum Sari dan P. Pandu Adikara, "Klasifikasi Hate Speech Berbahasa Indonesia di Twitter Menggunakan Naive Bayes dan Seleksi Fitur Information Gain dengan Normalisasi Kata," Malang, Mei 2019. [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [9] M. A. Rofiqi, Abd. C. Fauzan, A. P. Agustin, dan A. A. Saputra, "Implementasi Term-Frequency Inverse Document Frequency (TF-IDF) Untuk Mencari Relevansi Dokumen Berdasarkan Query," ILKOMNIKA: Journal of Computer Science and Applied Informatics, vol. 1, no. 2, hlm. 58–64, Des 2019, doi: 10.28926/ilkomnika.v1i2.18.
- [10] A. Ikegami, I. Dewa, M. Bayu, dan A. Darmawan, "Analisis Sentimen dan Pemodelan Topik Ulasan Aplikasi Noice Menggunakan XGBoost dan LDA," Badung, Nov 2022.
- [11] P. A. N. Aryanti dan I. B. M. Mahendra, "Analisis Sentimen Opini Berbahasa Indonesia Pada Sosial Media Menggunakan TF-IDF dan Support Vector Machine," Jurnal Elektronik Ilmu Komputer Udayana (JNATIA), vol. 12, no. 1, hlm. 2654–5101, Agu 2023.
- [12] R. Tuntun, K. Kusriani, dan K. Kusnawi, "Analisis Perbandingan Kinerja Algoritma Klasifikasi dengan Menggunakan Metode K-Fold Cross Validation," Jurnal Media Informatika Budidarma, vol. 6, no. 4, hlm. 2111, Okt 2022, doi: 10.30865/mib.v6i4.4681.