

Analisis Sentimen Aplikasi MyTelkomsel berbasis *Support Vector Machine*: Perbandingan Performa Kernel

Sentiment Analysis of the MyTelkomsel App based on Support Vector Machines: A Kernel Performance Comparison

¹Fitri Novianti Hidayah, ²Endah Setyowati*

^{1,2}Sistem Telekomunikasi, Kampus Daerah Purwakarta, Universitas Pendidikan Indonesia

^{1,2}Cibening, Kabupaten Purwakarta, Jawa Barat, Indonesia

*e-mail: endahsetyowati@upi.edu

(received: 6 April 2026, revised: 8 May 2026, accepted: 9 May 2026)

Abstrak

MyTelkomsel merupakan aplikasi layanan pelanggan milik salah satu operator seluler terbesar, dengan lebih dari 100 juta jumlah pengguna. Dikarenakan volume pengguna aplikasi yang tinggi, analisis sentimen diperlukan untuk mengkaji opini pengguna dalam optimalisasi layanan. Namun, dalam klasifikasi analisis sentimen sering kali terkendala karena distribusi kelas sentimen yang tidak seimbang, sehingga memengaruhi performa model. Penelitian ini menganalisis sentimen aplikasi MyTelkomsel berbasis algoritma Support Vector Machine (SVM), berfokus pada evaluasi performa kernel Linear, RBF, dan Polinomial. Dataset berjumlah 1000 ulasan pengguna secara acak pada Google Play Store, dengan pelabelan kelas positif dan negatif berdasarkan Indonesia Sentiment Lexicon (InSet), pemisahan data *training* dan *testing* menggunakan rasio 80:20. Proses perancangan model dilakukan di RapidMiner, dengan hasil performa optimal diperoleh kernel Linear dengan menerapkan SMOTE dan *K-Fold Cross Validation* yang mencapai tingkat *accuracy* 100%, *precision* sebesar 100%, *recall* sebesar 100%, dan *F1 score* sebesar 100%, yang menunjukkan bahwa data dapat dipisahkan dengan garis linear. Synthetic Minority Over-sampling Technique (SMOTE) digunakan untuk menangani ketidakseimbangan kelas pada data, dan *K-Fold Cross Validation* ($k=10$) diterapkan untuk memastikan tidak adanya *overfitting*, dengan pengujian pada keseluruhan data yang terbagi ke dalam 10 bagian. Hasil penelitian ini dapat dijadikan dasar pengoptimalan layanan aplikasi, sehingga strategi peningkatan aplikasi dapat terealisasi sesuai masukan dari berbagai ulasan pengguna.

Kata kunci: MyTelkomsel, analisis sentimen, support vector machine, rapidminer

Abstract

MyTelkomsel is a customer service application developed by one of the largest cellular operators, with more than 100 million users. Due to the high volume of application users, sentiment analysis is essential for examining user opinions to optimize service quality. However, sentiment classification often faces challenges caused by imbalanced sentiment class distributions, which can affect model performance. This study analyzes sentiment toward the MyTelkomsel application using the Support Vector Machine (SVM) algorithm, focusing on evaluating the performance of Linear, RBF, and Polynomial kernels. The dataset consisted of 1,000 user reviews randomly collected from the Google Play Store, with positive and negative labels assigned based on the Indonesia Sentiment Lexicon (InSet). The dataset was divided into training and testing sets using an 80:20 ratio. The model development process was carried out using RapidMiner. The optimal performance was achieved by the Linear kernel through the implementation of the Synthetic Minority Over-sampling Technique (SMOTE) and K-Fold Cross Validation, resulting in an accuracy of 100%, precision of 100%, recall of 100%, and F1-score of 100%. These results indicate that the data can be effectively separated using a linear boundary. SMOTE was applied to address class imbalance in the dataset, while K-Fold Cross Validation ($k = 10$) was used to ensure the absence of overfitting by testing the entire dataset divided into 10 folds. The findings of this study can serve as a foundation for optimizing application services, enabling improvement strategies to be implemented in accordance with feedback derived from user reviews.

Keywords: MyTelkomsel, sentiment analysis, support vector machine, rapidminer

<http://sistemasi.ftik.unisi.ac.id>

1 Pendahuluan

Teknologi informasi dan komunikasi telah berkembang secara pesat, terutama pada bidang telekomunikasi seluler. Perkembangan tersebut mendorong munculnya berbagai operator seluler sebagai penyedia layanan komunikasi yang dapat memahami kebutuhan dan preferensi setiap pengguna. Di Indonesia telah tersebar berbagai operator seluler yang memiliki berbagai karakteristik yang berbeda, yang meliputi kualitas jaringan, kestabilan koneksi, hingga biaya yang diberikan. PT Telekomunikasi Seluler (Telkomsel) termasuk kedalam salah satu perusahaan *provider* yang tersedia di Indonesia [1].

Telkomsel termasuk ke dalam salah satu *provider* terbesar di Indonesia, dengan persebaran pelanggan yang mencapai 159.6 juta yang tersebar diseluruh Indonesia pada tahun 2024 [2]. *Provider* Telkomsel menyediakan aplikasi MyTelkomsel yang dapat digunakan oleh setiap penggunanya untuk pembelian pulsa hingga paket internet. Pemilihan aplikasi MyTelkomsel sebagai objek penelitian ini dikarenakan termasuk dalam salah satu *provider* terbesar dan menjadi *provider* favorit pilihan masyarakat Indonesia [3]. Dengan banyaknya pengguna, maka beragam pula data yang dapat digunakan untuk menganalisis sentimen pengguna.

MyTelkomsel telah diunduh lebih dari 100 juta pengguna, dengan penilaian dari pengguna mencapai 4.0 dan memiliki ulasan yang mencapai 11 juta lebih di platform Google Play Store. Google Play Store adalah sebuah platform *marketplace* bagi pengguna *device* Android, yang menyediakan berbagai aplikasi, hingga dapat memberikan ulasan terhadap aplikasi yang telah digunakan sebagai opini publik terhadap aplikasi tersebut [4]. Berdasarkan ulasan pengguna aplikasi MyTelkomsel, pelanggan mengungkapkan ketidakpuasan dengan layanan yang ditawarkan oleh *provider* [5], didukung dengan persentase ulasan sentimen negatif yang digunakan pada penelitian ini yang mencapai 80.87% dari total keseluruhan data atau sebanyak 803 data dari total 993 jumlah data. Pernyataan tersebut didukung dengan keluhan pengguna yang mengungkapkan bahwa sering terjadi *error* saat menggunakan aplikasi MyTelkomsel [3]. Hal tersebut memunculkan berbagai ulasan atau komentar yang dapat dikategorikan positif dan negatif. Ulasan dari pengguna aplikasi dikumpulkan menjadi sebuah dataset yang mencerminkan kelayakan aplikasi tersebut, dan diharapkan menjadi saran pengembangan aplikasi MyTelkomsel di masa mendatang.

Penelitian ini menggunakan sebuah teknik Support Vector Machine (SVM), salah satu algoritma *machine learning* yang berfungsi untuk melakukan klasifikasi berdasarkan dataset yang diberikan. SVM menggunakan konsep memisahkan kelas menggunakan sebuah margin, hal tersebut menjadikan SVM memiliki kinerja yang baik dalam mengklasifikasikan sebuah kelas, termasuk dalam menganalisis klasifikasi sentimen [6]. Adapun berbagai jenis kernel SVM, seperti kernel Linear, Radial Basis Function (RBF), dan Polinomial. Kernel Linear berfungsi untuk menangani data linier, sedangkan kernel RBF dan Polinomial berfungsi untuk menangani data nonlinier dengan cara yang efisien [7].

Beberapa studi terdahulu telah banyak yang membahas analisis sentimen berbasis SVM, seperti pada [8] telah membandingkan performa kinerja pembobotan kata Term Frequency-Inverse Document Frequency (TF-IDF) dan Bag of Words (BoW) pada kernel Linear, Polinomial, dan RBF dengan menambahkan metode Synthetic Minority Over-sampling Technique (SMOTE), dan menggunakan berbagai nilai rasio *splitting* data. Studi [9] melakukan komparasi performa kinerja beberapa algoritma *machine learning*, salah satunya yaitu SVM, yang berfokus pada kernel Linear dalam konteks analisis sentimen sebuah aplikasi.

Penelitian lainnya [3] telah membahas penggunaan algoritma SVM dalam menganalisis sentimen aplikasi MyTelkomsel, dengan menambahkan beberapa metode untuk meningkatkan performa model, yaitu dengan menambahkan metode SMOTE untuk menangani ketidakseimbangan kelas dalam data, seleksi *fitur* Recursive Feature Elimination (RFE), dan metode *K-Fold Cross Validation* (k=10). Namun, penelitian tersebut hanya berfokus pada performa kernel Linear dan RBF, sehingga performa pada kernel Polinomial masih belum banyak dibahas. Oleh karena itu, penelitian ini dilakukan untuk mengisi celah penelitian tersebut dengan menganalisis performa kernel Linear, RBF, dan Polinomial dengan menambahkan metode SMOTE untuk mengatasi ketidakseimbangan kelas secara organik dalam data dan *K-Fold Cross Validation* (k=10) untuk memastikan tidak terjadi *overfitting* pada model.

Output penelitian ini akan divisualisasikan dalam *confusion matrix* berupa *heatmap* yang menampilkan perbandingan kinerja model berdasarkan perbedaan ketiga kernel SVM, dan saran pengembangan aplikasi MyTelkomsel berdasarkan hasil analisis sentimen pada penelitian ini. Temuan dari penelitian ini diharapkan mampu memberikan pemahaman yang tepat terkait tanggapan pengguna berdasarkan berbagai ulasan yang telah diberikan, memberikan saran kernel paling optimal yang dapat digunakan dalam teknik klasifikasi menggunakan SVM, dan menjadi landasan untuk optimalisasi layanan aplikasi MyTelkomsel di masa yang akan datang berdasarkan ulasan aktual pengguna pada platform Google Play Store.

2 Tinjauan Literatur

Berbagai penelitian terdahulu banyak dilakukan mengenai analisis sentimen aplikasi MyTelkomsel menggunakan berbagai teknik algoritma yang berbeda-beda berdasarkan ulasan pada platform Google Play Store, hingga analisis sentimen terkait topik yang sedang banyak dibicarakan berdasarkan sentimen pengguna Twitter.

Penelitian yang dilakukan [3] membahas analisis sentimen pengguna MyTelkomsel pada Google Play Store berbasis algoritma SVM yang dikombinasikan dengan metode SMOTE, seleksi *fitur* RFE sebesar $n=7$, dan *K-Fold Cross Validation* ($k=10$) pada kernel Linear dan RBF, yang menghasilkan tingkat akurasi tertinggi pada kernel RBF yang mencapai 99.7%, *precision* sebesar 100%, *recall* sebesar 100%, dan *F1-score* sebesar 99.68%. Dikarenakan penelitian tersebut belum membahas mengenai performa kernel Polinomial, studi ini dilakukan untuk mengisi celah dengan melakukan analisis performa kernel Linear, Polinomial, dan RBF dengan pengambilan dataset secara acak dan menambahkan metode SMOTE dan *K-Fold Cross Validation* untuk pengoptimalan kinerja performa model dan mencegah terjadinya *overfitting* pada model.

Objek penelitian serupa dilakukan pada studi [10] yang menganalisis sentimen pengguna MyTelkomsel pada Google Play Store berbasis Naive Bayes yang dikombinasikan dengan metode SMOTE dan *K-Fold Cross Validation*, pada *software* RapidMiner. Dengan hasil penelitian yang mencapai tingkat akurasi sebesar 86.05%, *precision* 87.56%, dan *recall* 84.04%.

Sedangkan, studi [8] membahas analisis sentimen pengguna Twitter terkait topik yang sedang banyak dibicarakan pada saat itu, yaitu terkait kebijakan kenaikan harga BBM menggunakan algoritma SVM yang dikombinasikan dengan metode SMOTE, yang berfokus pada membandingkan kinerja pembobotan kata dengan TF-IDF dan BoW di beberapa kernel, yaitu kernel Linear, RBF, dan Polinomial. Hasil dari penelitian menunjukkan keunggulan pembobotan kata menggunakan TF-IDF pada kernel RBF, dengan tingkat akurasi mencapai 87%.

Di sisi lain, penelitian [9] menganalisis sentimen pengguna aplikasi Mobile JKN menggunakan berbagai algoritma *machine learning*, diantaranya SVM yang berfokus pada kernel Linear, logistic regression, dan Clustered SVM (CSVM), dengan hasil performa pada kernel Linear SVM yang menunjukkan kinerja yang baik dengan tingkat akurasi mencapai 91%, *precision* 92%, dan *recall* 98%, %, dan *F1-score* sebesar 95%.

Selain itu, studi [11] membandingkan metode labeling data Inset dengan Valence Aware Dictionary and sEntiment Reasoner (VADER) terhadap data sentimen opini terkait topik Rohingya pada platform X menggunakan algoritma SVM. Hasil studi menunjukkan performa InSet lebih optimal disbanding VADER, dengan tingkat akurasi mencapai 87.83%.

Berdasarkan beberapa penelitian yang telah dilakukan sebelumnya, meskipun telah banyak yang melakukan studi analisis sentimen menggunakan algoritma SVM dengan menerapkan metode SMOTE dan *K-Fold Cross Validation*, tetapi masih belum banyak yang mengevaluasi performa kinerja ketiga kernel SVM yaitu Linear, RBF, dan Polinomial secara bersamaan terhadap ulasan pengguna di platform Google Play Store. Oleh karena itu, peneliti bermaksud untuk membuat studi dengan membandingkan kinerja ketiga kernel SVM, yang mencakup kernel Linear, RBF, dan Polinomial dalam menganalisis sentimen pada aplikasi MyTelkomsel untuk menentukan kinerja akurasi terbaik dalam mengkategorikan jenis ulasan positif dan negatif pada dataset ulasan pengguna menggunakan metode InSet. Pemilihan dataset sejumlah 1000 ulasan pengguna secara acak, didasari oleh penelitian sebelumnya [3] yang menggunakan jumlah dataset serupa, dan berhasil menunjukkan performa akurasi yang baik dalam mengklasifikasikan sentimen. Jumlah pengambilan dataset tersebut telah representatif mewakili sentimen positif dan negatif terhadap penggunaan aplikasi MyTelkomsel.

3 Metode Penelitian

Diagram metodologi penelitian yang digunakan dalam evaluasi kernel SVM berdasarkan analisis sentimen pengguna aplikasi MyTelkomsel terlihat pada Gambar 1.



Gambar 1 Alur penelitian

3.1 Identifikasi Masalah

Tahap identifikasi masalah berisikan perumusan dan analisis atas permasalahan utama yang terjadi. Masalah utama pada penelitian ini yaitu mengenai analisis sentimen pengguna MyTelkomsel berbasis algoritma SVM pada kernel Linear, RBF, dan Polinomial. Pemilihan SVM pada penelitian ini dikarenakan terbukti unggul sebagai model klasifikasi yang digunakan dalam mengatasi data berdimensi tinggi, dan kemampuannya dalam menangani data nonlinier menggunakan fungsi kernel dengan mengubah bentuk data ke dalam dimensi yang lebih tinggi agar menyederhanakan metode pembagian data [12].

3.2 Pengumpulan Kajian Literatur

Tahap ini, berisi pengumpulan kajian literatur dari berbagai *paper* maupun artikel yang berkaitan dengan topik analisis sentimen.

3.3 Pengumpulan Dataset

Dataset diambil sesuai dengan ulasan pengguna MyTelkomsel pada platform Google Play Store memakai teknik *data scraping* di Google Colab. Pengumpulan dataset berjumlah 1000 data dalam ulasan paling terbaru. Pengambilan dataset dilakukan secara acak (*random sampling*), untuk mempertahankan distribusi ulasan aktual pengguna aplikasi, tanpa menyeimbangkan kelas data secara manual.

3.4 Data Preprocessing

Tahap ini berisikan tahapan *case folding*, *tokenizing*, *filtering*, *stemming*, dan *detokenizing* yang dilakukan pada platform Google Colab. Hal tersebut dilakukan agar data menjadi bersih dan siap untuk diberikan pelabelan kelas.

Tahap pertama, proses *case folding* yang berguna agar data konsisten dalam konteks kepenulisan dengan mengubah keseluruhan teks ke dalam huruf kecil [13]. Selanjutnya, data yang berbentuk teks dipecah menjadi satuan kata atau disebut dengan token dalam proses *tokenizing*, dan dilakukan proses *filtering* menggunakan *stopword removal* untuk menghilangkan beberapa kata umum yang dianggap tidak terlalu penting. Lalu, proses *stemming* dilakukan untuk mengubah kata ke dalam bentuk dasarnya melalui penghapusan afiks [14]. Proses diakhiri dengan *detokenizing*, untuk menggabungkan kata atau token yang berupa *string* menjadi sebuah kalimat biasa agar dapat diproses pada simulator RapidMiner [15].

3.5 Pelabelan Data

Untuk menjamin validitas dan menghindari bias subjektivitas pada label, dilakukan proses labeling berdasarkan kamus Indonesia Sentiment Lexicon (InSet) yang diproses dengan Python di platform Google Colab. Dikarenakan keterbatasan validitas *expert* dan membutuhkan proses yang memakan waktu dalam proses pelabelan manual, sehingga penelitian ini menggunakan metode InSet sebagai pelabelan otomatis yang didasarkan dengan *list* kata pada kamus Lexicon, sebanyak 3609 kata positif dan 6609 kata negatif. Metode ini menilai setiap kata dari kalimat ulasan pengguna

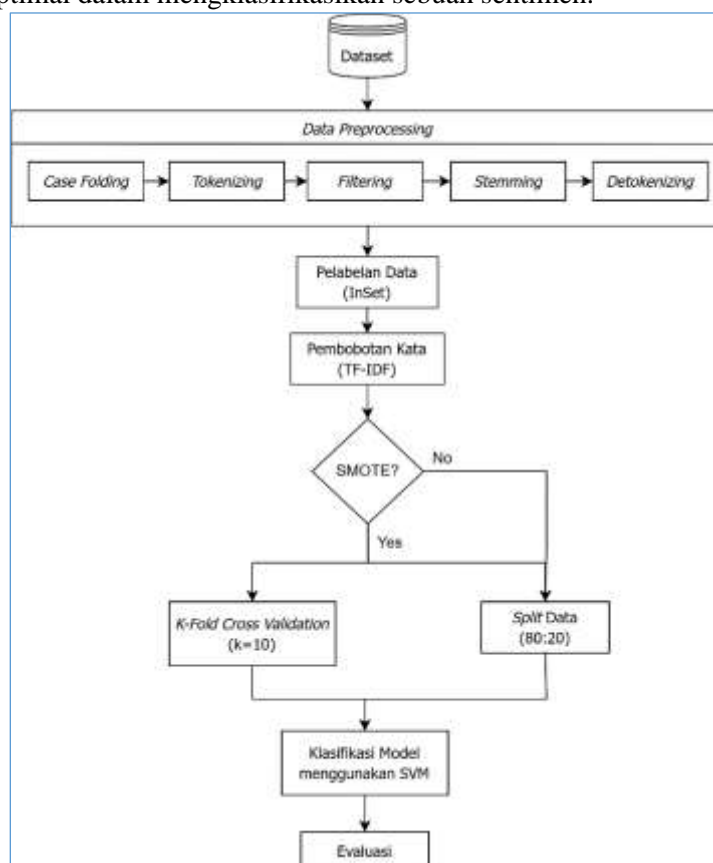
<http://sistemasi.ftik.unisi.ac.id>

berdasarkan bobot yang telah ditentukan dengan *range* penilaian dari -5 hingga +5 [16]. Penilaian setiap kata dalam kalimat ulasan tersebut dijumlahkan, dan jika nilai kurang dari atau sama dengan 0, maka ulasan diklasifikasikan kedalam kelas negatif. Sedangkan, jika total penilaian lebih dari 0, maka ulasan diklasifikasikan kedalam kelas positif. Namun, InSet memiliki keterbatasan dalam mendeteksi kata singkatan, *slang*, hingga kata yang sebagian kecil berbahasa asing, membuat InSet memberikan penilaian mutlak 0 untuk beberapa kata yang tidak dapat dideteksi tersebut. Proses pelabelan data ini dilakukan setelah tahap *preprocessing* dikarenakan agar terhindar dari *noise* dan memastikan kualitas teks ulasan lebih terstruktur dan bersih, sehingga dapat meminimalisir kesalahan InSet dalam memberikan penilaian untuk setiap kata.

3.6 Perancangan Sistem

Tahap perancangan sistem ditunjukkan oleh Gambar 2. Tahapan ini dilakukan pada simulator RapidMiner. Diawali dengan tahap TF-IDF, yang merupakan salah satu teknik pembobotan kata dengan cara mengubah setiap kata menjadi representasi numerik [17]. Metode ini digunakan karena algoritma SVM hanya dapat membaca sebuah data numerik. Lalu model akan dilatih dan diuji menggunakan algoritma SVM dalam kernel Linear, RBF, dan Polinomial. Untuk membandingkan performa kinerja model, dilakukan pengujian model dengan membagi data untuk *training* dan *testing* menggunakan rasio 80:20 dan *K-Fold Cross Validation* agar stabilitas performa model tetap terjaga dengan mencegah terjadinya *overfitting* pada model.

Model diuji berdasarkan 3 skenario. Yang pertama, menggunakan *split* data secara manual berdasarkan rasio 80:20 tanpa menggunakan metode SMOTE. Kedua, masih menggunakan *split* data manual, tetapi dengan menerapkan metode SMOTE untuk meningkatkan performa model. Dan, yang ketiga dengan menggunakan metode SMOTE dan *K-Fold CrFoss Validation* ($k=10$) untuk meminimalisir terjadinya *overfitting* pada model, dengan membaginya kedalam 10 bagian, lalu dilakukan 10 kali putaran pengujian. Di setiap putaran, 9 bagian digunakan sebagai *training* dan 1 bagian lainnya digunakan sebagai *testing* secara bergantian. Hal ini dilakukan untuk melihat model mana yang paling optimal dalam mengklasifikasikan sebuah sentimen.



Gambar 2 Diagram perancangan sistem

3.7 Evaluasi Performa Sistem

Pada tahap akhir ini, dilakukan evaluasi terhadap model pada ketiga kernel tersebut, untuk melihat kernel yang paling optimal berdasarkan metode *confusion matrix* untuk menghitung parameter *accuracy*, *precision*, *recall*, dan *F1 score*. Hal tersebut bertujuan agar mengetahui sejauh mana model dapat mengklasifikasikan sentimen dengan tepat [18].

1. Accuracy

Parameter *accuracy*, berfungsi untuk mengevaluasi ketepatan model dalam mengkategorikan data dengan benar, dapat dihitung secara manual menggunakan rumus persamaan (1) [19]:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (1)$$

Keterangan:

True Positive (TP): Model mengklasifikasikan data positif menjadi kelas positif

True Negative (TN): Model mengklasifikasikan data negatif menjadi kelas negatif

False Positive (FP): Model mengklasifikasikan data negatif menjadi kelas positif

False Negative (FN): Model mengklasifikasikan data positif menjadi kelas negatif

2. Precision

Parameter *precision*, merupakan persentase ketepatan model dalam memprediksi data menjadi positif, dapat dihitung secara manual menggunakan rumus persamaan (2) [19]:

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

3. Recall

Parameter *recall*, merupakan persentase seberapa banyak model dalam mendeteksi data positif dengan benar, dapat dihitung secara manual menggunakan rumus persamaan (3) [19]:

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

4. F1 score

Parameter *F1 score*, merupakan rata-rata dari *precision* dan *recall* agar mengetahui performa model [19]. Untuk menghasilkan nilai *F1 score* yang optimal, dibutuhkan nilai parameter *precision* dan *recall* yang optimal pula, yang dapat dihitung secara manual menggunakan rumus persamaan (4):

$$F1\ score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

4 Hasil dan Pembahasan

4.1 Data Scraping

Proses *scraping* terlihat pada Gambar 3 yang dilakukan pada platform Google Colab.

```
from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.telkomsel.telkomselcm',
    lang='id',
    country='id',
    sort=Sort.NEWEST,
    count=1000,
    filter_score_with=None,
)

print(f>Data yang berhasil ditarik: {len(result)} ulasan")
```

Gambar 3 Data scraping

Pengambilan dataset dilakukan berdasarkan ulasan pengguna aplikasi MyTelkomsel pada platform Google Play Store, dengan kategori ulasan paling terbaru sebanyak 1000 ulasan, dan dukungan bahasa pemrograman Python. *Library* Python yang digunakan untuk mendukung proses ini yaitu *google_play_scraper* dan *pandas*.

Atribut yang digunakan yaitu berdasarkan nama pengguna (*username*), skor (*score*), waktu (*at*), dan rangkaian kalimat ulasan (*content*). Setelah ulasan terkumpul, dataset disimpan menggunakan format Excel (*xlsx*) agar lebih mudah terbaca.

4.2 Data Preprocessing

Tahap *preprocessing* bertujuan untuk pembersihan dataset, agar data dapat terstruktur dengan baik dan siap untuk diolah dalam merancang sistem. Tahapan ini dilakukan menggunakan platform Google Colab, meliputi tahapan *case folding*, *tokenizing*, *filtering*, *stemming*, dan *detokenizing*. *Library* Python yang digunakan untuk mendukung proses ini, yaitu *pandas*, *nlTK*, dan *Sastrawi*.

4.2.1 Case Folding

Proses *case folding* berfungsi untuk memodifikasi keseluruhan huruf dalam dataset menjadi *lowercase*, seperti terlihat pada Gambar 4.

```
# case folding
df['casefolding'] = df['CONTENT'].astype(str).str.lower()
df.head()
```

Gambar 4 Case folding

4.2.2 Tokenizing

Proses *tokenizing* bertujuan untuk memisahkan kalimat menjadi satuan kata atau token, seperti terlihat pada Gambar 5.

```
# tokenizing
df['tokenizing'] = df['casefolding'].apply(word_tokenize)
df['tokenizing'] = df['tokenizing'].apply(
    lambda tokens: [t for t in tokens if t.isalpha()]
)
df.head()
```

Gambar 5 Tokenizing

4.2.3 Filtering

Filtering dalam penelitian ini menggunakan *stopword removal*. Proses ini terlihat pada Gambar 6. yang bertujuan untuk menghilangkan kata yang dianggap tidak terlalu penting. Dikarenakan khawatir bila kata penting ikut terhapus, maka disisipkan variabel “kata_penting” sebagai pengecualian agar tidak terhapus.

```
# filtering (stopword removal)
stopword_id = set(stopwords.words('indonesian'))

# kata penting
kata_penting = {
    'telkomsel',
    'baik',
    'buruk',
    'jelek',
    'bagus',
    'mantap',
    'oke',
    'ok',
    'lemot',
    'luar',
    'biasa',
    'tidak',
    'sangat',
    'cukup',
    'puas',
    'suka',
    'keren'
}

stopword_id = stopword_id - kata_penting

df['stopword'] = df['tokenizing'].apply(
    lambda x: [word for word in x if word not in stopword_id]
)
df.head()
```

Gambar 6 Filtering

4.2.4 Stemming

Proses *stemming* terlihat pada Gambar 7 yang mencakup pengecilan indeks kata, hingga penghapusan afiks dalam kata tersebut.

```
# stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()

kata_penting_2 = {
    'lemot',
    'jaringan',
    'error',
    'crash',
    'bug',
    'lag',
    'loading'
}

# normalisasi whitelist
kata_penting_2 = {w.lower().strip() for w in kata_penting_2}

def custom_stemming(tokens):
    result = []
    for w in tokens:
        w = w.lower().strip()
        if w in kata_penting_2:
            result.append(w) # kata penting, agar tidak distem
        else:
            result.append(stemmer.stem(w))
    return result

df.head()
```

Gambar 7 Stemming

Berdasarkan gambar tersebut, penambahan variabel *whitelist* “kata_penting_2” dalam tahapan ini bertujuan agar kata-kata tertentu tidak mengalami *overstemming* yang dapat merubah makna yang sebenarnya.

4.2.5 Detokenizing

Proses *detokenizing* bertujuan untuk mengubah token yang berupa *string*, menjadi sebuah kalimat agar dapat melanjutkan proses pembobotan kata dan pemodelan sistem dalam simulator RapidMiner. Setelah proses *preprocessing* selesai, dataset disimpan menggunakan format Excel (xlsx), seperti terlihat pada Gambar 8.

```
# gabungkan token menjadi teks biasa/kalimat akhir setelah stemming
df['stemming'] = df['stopword'].apply(custom_stemming)

df['hasil_akhir_preprocessing'] = df['stemming'].apply(lambda x: ' '.join(x))
df.head()
```

Gambar 8 Detokenizing

4.3 Labeling

Labeling pada dataset dilakukan secara otomatis menggunakan metode InSet untuk menjamin validitas dan menghindari bias subjektivitas pada data yang diproses dengan menggunakan Python pada platform Google Colab. Dikarenakan terdapat beberapa data yang hilang setelah melalui tahap *preprocessing*, untuk menghindari hasil pelabelan yang tidak akurat, pada saat labeling ditambahkan fungsi untuk menghapus data hilang atau tidak mengandung informasi tersebut. Dengan perolehan label negatif mencapai 803 data, sedangkan label positif hanya 190 data dari total data bersih sebanyak 993 data, seperti yang ditunjukkan pada Gambar 9. Kondisi tersebut menunjukkan adanya ketidakseimbangan kelas pada data, yang didominasi oleh kelas negatif.

```
=====
HASIL LABELING LEXICON
=====
label
NEGATIF      803
POSITIF      190
Name: count, dtype: int64
-----
Total Data: 993
=====
```

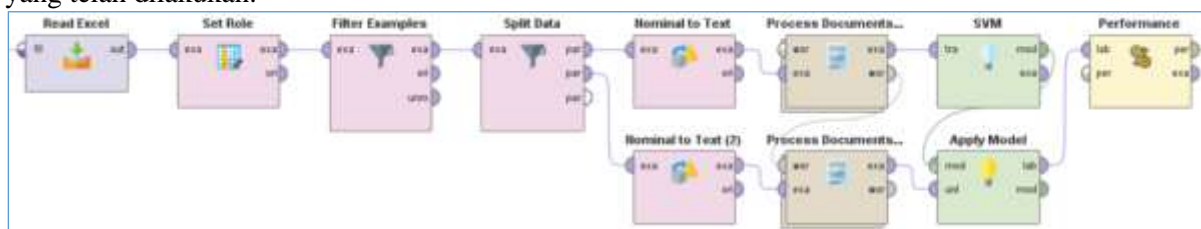
Gambar 9 Hasil labelling

4.4 Pembobotan Kata TF-IDF & Klasifikasi Model menggunakan SVM

Proses TF-IDF dilakukan setelah tahap data *preprocessing*, yang bertujuan untuk mengubah keseluruhan kata dalam data menjadi representasi numerik. Gambar 10, Gambar 11, dan Gambar 12 menunjukkan tahap TF-IDF hingga pemodelan sistem menggunakan algoritma SVM dilakukan pada simulator RapidMiner.

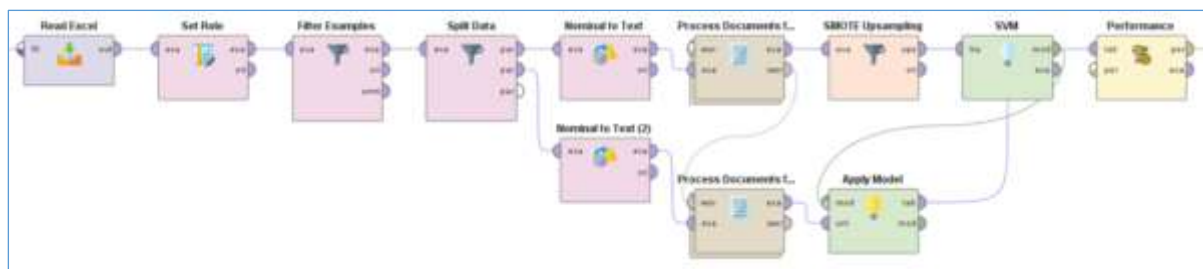
Langkah awal, *install* terlebih dahulu *extension* “Text Processing”, yang akan digunakan sebagai operator utama proses TF-IDF. Selanjutnya, gunakan operator “Read Excel” untuk mengimpor dataset. Operator “Set Role” untuk penentuan label sebagai *role*, agar model dapat dengan mudah mengklasifikasikan sentimen berdasarkan label positif maupun negatif yang telah dibuat. Operator “Filter Examples” untuk memastikan dataset tidak hilang dengan menerapkan kondisi *no missing attributes*. Namun, jika terdapat data yang hilang pada salah satu atribut atau kolom, maka akan otomatis terhapus agar tidak mempengaruhi data yang utuh lainnya. Operator “Split Data” untuk membagi data *training* dan testing, menggunakan rasio 80:20.

Operator “Nominal to Text” untuk mengubah tipe atribut pada atribut/kolom *text* ulasan pengguna dari nominal menjadi *text*, agar bisa diproses untuk tahap pembobotan kata (TF-IDF). Operator “Process Documents from Data”, sebagai operator utama TF-IDF untuk mengubah data menjadi representasi numerik, menggunakan operator “Tokenize” dan “Filter Token by Length” dengan maksimal 10 karakter di dalamnya. Dalam sistem, terdapat 2 operator “Nominal to Text” dan “Process Documents from Data”, hal tersebut dikarenakan penggunaan kedua operator yang terpisah pada data *training* dan testing. Operator “SVM (LibSVM)”, sebagai operator utama dalam klasifikasi model menggunakan algoritma SVM, yang digunakan untuk melatih model pada data *training*. Dengan mengganti satu per satu bagian tipe kernel untuk menguji kernel Linear, RBF, dan Polinomial, tanpa mengubah parameter C dan Y (sehingga menggunakan nilai parameter *default*). Operator “Apply Model” untuk menerapkan model SVM yang telah dilatih pada data *training*, ke dalam data testing. Operator “Performance (Classification)” untuk evaluasi hasil klasifikasi model yang telah dilakukan.



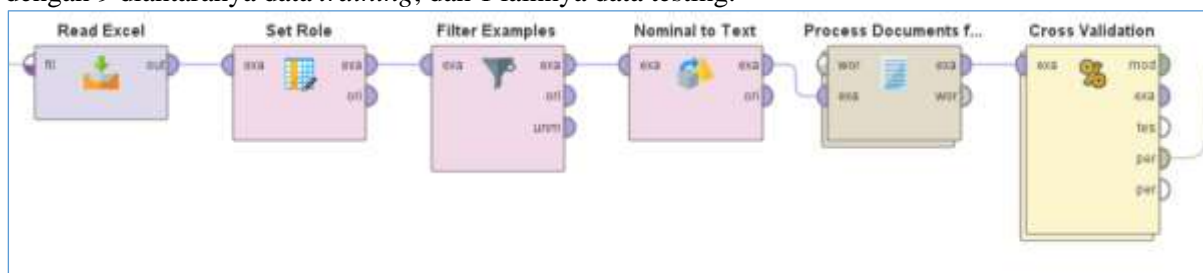
Gambar 10 Perancangan sistem tanpa metode SMOTE

Untuk pengoptimalan kinerja model, dilakukan pengujian dengan menerapkan metode SMOTE guna menyelesaikan masalah karena ketidakseimbangan kelas secara organik pada data, serta meningkatkan performa *recall* dan *precision* pada kernel Polinomial. Penambahan operator “SMOTE Upsampling”, sebagai operator SMOTE, dengan mengatur nilai “nominal change rate” menjadi 1.0 atau 100%, dan menggunakan nilai *default* pada parameter “number of neighbours”, serta tanpa mengubah parameter C dan Y pada setiap kernel yang digunakan.



Gambar 11 Perancangan sistem dengan metode SMOTE

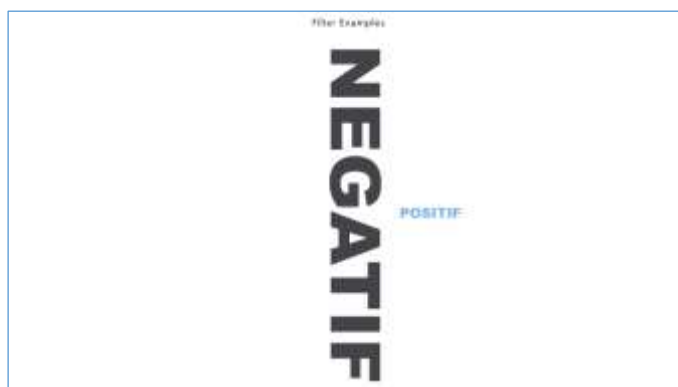
Untuk memastikan tidak adanya *overfitting* dan memastikan performa model tetap stabil, dilakukan pengujian dengan menggabungkan metode SMOTE 100% dan *K-Fold Cross Validation* (k=10), yang berarti dilakukan sebanyak 10 kali pengujian model yang terbagi ke dalam 10 bagian, dengan 9 diantaranya data *training*, dan 1 lainnya data testing.



Gambar 12 Perancangan sistem dengan metode SMOTE dan K-Fold cross validation

4.5 Evaluasi

Penelitian ini menggunakan 1000 data mentah yang berisikan ulasan pengguna aplikasi MyTelkomsel, dengan hasil akhir dataset yang utuh setelah dilakukan tahap *preprocessing* sebanyak 993 data. Pembagian data *training* dan testing berdasarkan rasio 80:20, dengan rincian sebanyak 794 data *training*, dan 199 data testing. Berdasarkan dataset utuh tersebut, diperoleh sebanyak 803 ulasan negatif, dan 190 ulasan positif, yang telah dilabeli menggunakan metode InSet seperti terlihat pada *wordcloud* Gambar 13.



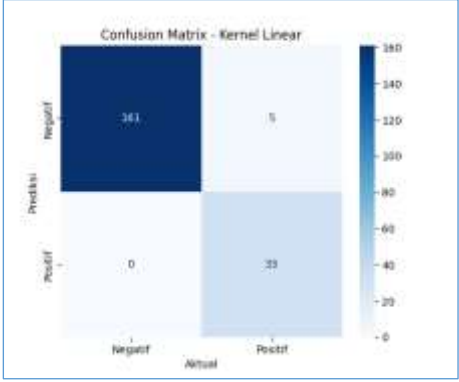
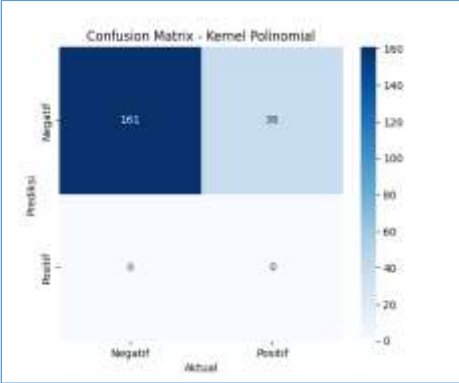
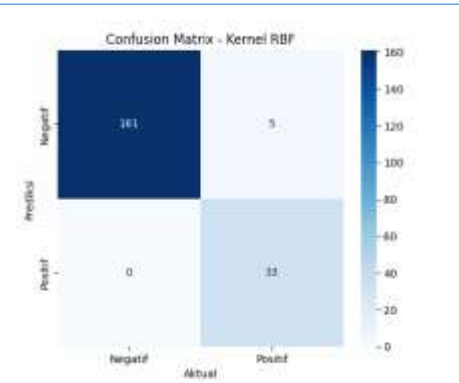
Gambar 13 Wordcloud kelas sentimen InSet

Evaluasi dari performa klasifikasi model pada data testing, disajikan menggunakan *confusion matrix* berupa *heatmap*. Terdapat 3 kernel dengan metode tanpa SMOTE dan menggunakan metode SMOTE yang dilakukan evaluasi performa, untuk menentukan kernel paling optimal yang digunakan dalam model klasifikasi sentimen ulasan pengguna aplikasi MyTelkomsel berdasarkan parameter *accuracy*, *precision*, *recall*, dan *F1 score*.

4.5.1 Model Tanpa Metode SMOTE

Output performa model dengan *split* data secara manual menggunakan rasio 80:20, tanpa menerapkan metode SMOTE pada ketiga kernel dengan nilai c dan y *default* terlihat pada Tabel 1.

Tabel 1 Performa model tanpa metode SMOTE

Kernel	Accuracy	Recall	Precision	F1 Score	Confusion Matrix
Linear	97.49%	86.84%	100%	92.96%	
Polinomial	80.90%	0	0	0	
RBF	97.49%	86.84%	100%	92.96%	

4.5.2 Model dengan Metode SMOTE

Output performa model dengan *split* data secara manual menggunakan rasio 80:20, dengan menerapkan metode SMOTE sebesar 100% pada ketiga kernel dengan nilai *c* dan *y default* seperti yang terlihat pada Tabel 2.

Tabel 2 Performa model dengan metode SMOTE

Kernel	Accuracy	Recall	Precision	F1 Score	Confusion Matrix
Linear	100%	100%	100%	100%	<p>Confusion Matrix - Kernel Linear - SMOTE 100%</p> <p>This heatmap shows a perfect classification performance. The top-left cell (True Positive) is dark blue with a value of 160. The top-right cell (False Positive) is white with a value of 0. The bottom-left cell (False Negative) is white with a value of 0. The bottom-right cell (True Negative) is light blue with a value of 160. The x-axis is labeled 'Aktual' with 'Negatif' and 'Positif' categories. The y-axis is labeled 'Prediksi' with 'Negatif' and 'Positif' categories. A color scale on the right ranges from 0 to 160.</p>
Polinomial	49.75%	100%	27.54%	43.18%	<p>Confusion Matrix - Kernel Polinomial - SMOTE 100%</p> <p>This heatmap shows poor classification performance. The top-left cell (True Positive) is medium blue with a value of 40. The top-right cell (False Positive) is white with a value of 0. The bottom-left cell (False Negative) is dark blue with a value of 109. The bottom-right cell (True Negative) is light blue with a value of 70. The x-axis is labeled 'Aktual' with 'Negatif' and 'Positif' categories. The y-axis is labeled 'Prediksi' with 'Negatif' and 'Positif' categories. A color scale on the right ranges from 0 to 100.</p>
RBF	100%	100%	100%	100%	<p>Confusion Matrix - Kernel RBF - SMOTE 100%</p> <p>This heatmap shows a perfect classification performance, identical to the Linear kernel. The top-left cell (True Positive) is dark blue with a value of 160. The top-right cell (False Positive) is white with a value of 0. The bottom-left cell (False Negative) is white with a value of 0. The bottom-right cell (True Negative) is light blue with a value of 160. The x-axis is labeled 'Aktual' with 'Negatif' and 'Positif' categories. The y-axis is labeled 'Prediksi' with 'Negatif' and 'Positif' categories. A color scale on the right ranges from 0 to 160.</p>

4.5.3 Model dengan Metode SMOTE dan K-Fold Cross Validation

Dikarenakan kernel Linear dan RBF yang mencapai performa sempurna hingga mencapai 100%, hal tersebut mengakibatkan adanya kemungkinan terjadi *overfitting* saat pengujian, sehingga model diujikan dengan penerapan metode *K-fold Cross Validation* sebesar $k=10$ untuk meminimalisir bias saat penggunaan *split* data secara manual. Pada model ini, keseluruhan jumlah data pada penelitian ini dilakukan pengujian. Tabel 3 menunjukkan performa model dengan menerapkan metode SMOTE sebesar 100% pada ketiga kernel dengan nilai c dan γ default, dan metode *K-fold Cross Validation*.

Tabel 3 Performa model dengan metode SMOTE dan K-Fold Cross validation

Kernel	Accuracy	Recall	Precision	F1 Score	Confusion Matrix									
Linear	100%	100%	100%	100%	<p>Confusion Matrix - Kernel Linear - SMOTE 100% - K=10</p> <table border="1"> <tr> <td>Prediksi \ Aktual</td> <td>Negatif</td> <td>Positif</td> </tr> <tr> <td>Negatif</td> <td>800</td> <td>0</td> </tr> <tr> <td>Positif</td> <td>0</td> <td>150</td> </tr> </table>	Prediksi \ Aktual	Negatif	Positif	Negatif	800	0	Positif	0	150
Prediksi \ Aktual	Negatif	Positif												
Negatif	800	0												
Positif	0	150												
Polinomial	49.55%	100%	27.50%	43.13%	<p>Confusion Matrix - Kernel Polinomial - SMOTE 100% - K=10</p> <table border="1"> <tr> <td>Prediksi \ Aktual</td> <td>Negatif</td> <td>Positif</td> </tr> <tr> <td>Negatif</td> <td>0</td> <td>501</td> </tr> <tr> <td>Positif</td> <td>501</td> <td>150</td> </tr> </table>	Prediksi \ Aktual	Negatif	Positif	Negatif	0	501	Positif	501	150
Prediksi \ Aktual	Negatif	Positif												
Negatif	0	501												
Positif	501	150												
RBF	95.37%	100%	80.51%	89.20%	<p>Confusion Matrix - Kernel RBF - SMOTE 100% - K=10</p> <table border="1"> <tr> <td>Prediksi \ Aktual</td> <td>Negatif</td> <td>Positif</td> </tr> <tr> <td>Negatif</td> <td>0</td> <td>157</td> </tr> <tr> <td>Positif</td> <td>157</td> <td>150</td> </tr> </table>	Prediksi \ Aktual	Negatif	Positif	Negatif	0	157	Positif	157	150
Prediksi \ Aktual	Negatif	Positif												
Negatif	0	157												
Positif	157	150												

4.6 Analisis

Berdasarkan hasil pengujian analisis sentimen pengguna aplikasi MyTelkomsel memakai algoritma SVM, dengan kernel Linear, RBF, dan Polinomial pada RapidMiner, dengan performa *accuracy* paling optimal yang diperoleh kernel Linear sebesar 100%, *precision* sebesar 100%, *recall* sebesar 100%, dan *F1 score* sebesar 100%. Model ini memiliki performa yang sangat baik, setelah dilakukan pengujian dengan menerapkan metode SMOTE dan *K-fold Cross Validation* sebesar k=10 untuk meminimalisir *overfitting*, tetapi kernel Linear konsisten menunjukkan tingkat akurasi yang sempurna saat pengujian dengan *split* data manual menggunakan rasio 80:20, maupun teknik *K-fold Cross Validation*. Hal tersebut menunjukkan bahwa data dapat dipisahkan hanya dengan garis linear atau lurus dengan baik.

Pada kernel RBF, memperoleh performa *accuracy* sebesar 95.37%, *precision* sebesar 80.51%, *recall* sebesar 100%, dan *F1 score* sebesar 89.20% setelah menerapkan metode SMOTE dan *K-fold*

Cross Validation sebesar $k=10$. Saat pengujian dengan metode *split* data manual menggunakan rasio 80:20, performa model mencapai performa yang sempurna, karena dikhawatirkan adanya kemungkinan *overfitting*, maka dilakukan pengujian dengan menambahkan metode *K-fold Cross Validation*. Terjadi penurunan nilai *precision* saat penerapan *K-fold Cross Validation*, dikarenakan model memiliki keterbatasan memprediksi kelas positif dengan baik. Hal tersebut didukung karena adanya ketidakseimbangan kelas pada data, dengan mayoritas kelas negatif sebesar 80.87% dari jumlah keseluruhan data.

Kernel Polinomial menunjukkan performa terendah dengan perolehan *accuracy* sebesar 49.55%, *precision* sebesar 27.50%, *recall* sebesar 100%, dan *F1 score* sebesar 43.13% setelah menerapkan pengoptimalan dengan metode SMOTE dan *K-fold Cross Validation* sebesar $k=10$. Meskipun sebelumnya mendapatkan tingkat *accuracy* sebesar 80.90% saat pengujian tanpa metode SMOTE dan *K-fold Cross Validation*, akan tetapi model sama sekali tidak dapat mendeteksi dan memprediksi kelas positif dengan nilai *recall* dan *precision* yang sama-sama berada di angka 0. Tetapi, setelah dilakukan pengoptimalan dan mencegah adanya *overfitting*, model justru mengalami penurunan tingkat akurasi.

5 Kesimpulan

Setelah dilakukan penelitian, maka disimpulkan kernel Linear efektif dalam mengklasifikasikan sentimen ulasan pengguna MyTelkomsel, pada kondisi ketidakseimbangan kelas sentimen dalam dataset. Dikarenakan performa Linear yang paling optimal, menunjukkan bahwa data memiliki karakteristik *linearly separable* atau dapat dipisahkan hanya dengan garis lurus. Jumlah dataset mentah pada penelitian ini sebanyak 1000 ulasan yang diambil secara acak (*random sampling*), untuk mempertahankan distribusi ulasan murni pengguna aplikasi, dengan dataset akhir yang utuh mencapai 993 ulasan setelah melalui rangkaian tahap *preprocessing*. Dengan rincian hasil performa terbaik setelah pengoptimalan menggunakan metode SMOTE dan *K-Fold Cross Validation* ($k=10$) diperoleh kernel Linear yang memiliki tingkat *accuracy* sebesar 100%, nilai *precision* sebesar 100%, nilai *recall* sebesar 100%, dan nilai *F1 score* sebesar 100%. Kernel RBF memiliki tingkat *accuracy* sebesar 95.37%, nilai *precision* sebesar 80.51%, nilai *recall* sebesar 100%, dan nilai *F1 score* sebesar 89.20% setelah dilakukan pengoptimalan performa dengan metode SMOTE dan *K-Fold Cross Validation* ($k=10$). Kernel Polinomial menunjukkan performa terendah dengan tingkat *accuracy* sebesar 49.55%, nilai *precision* sebesar 27.50%, nilai *recall* sebesar 100%, dan nilai *F1 score* yang hanya sebesar 43.13% meskipun telah dilakukan pengoptimalan performa dengan metode SMOTE dan *K-Fold Cross Validation* ($k=10$). Kernel Polinomial memiliki keterbatasan dalam memprediksi kelas positif, hal tersebut terlihat pada nilai *precision* yang tergolong rendah meskipun telah dilakukan pengoptimalan performa dengan metode SMOTE dan *K-Fold Cross Validation*. Hal ini disebabkan oleh adanya ketidakseimbangan kelas sentimen, dengan dominasi kelas negatif dalam dataset, sehingga model terlihat bias terhadap kelas negatif yang tergolong mayoritas dan kurang mampu mengklasifikasikan kelas positif yang tergolong minoritas. Adapun saran metode lain untuk meningkatkan performa akurasi pada model dapat dengan penambahan jumlah dataset, penambahan kelas sentimen netral, atau dengan menerapkan filter berdasarkan rating ulasan pengguna pada tahap data *scraping* agar kelas sentimen negatif maupun positif dapat seimbang, hingga menambahkan parameter *tuning* pada setiap nilai c dan γ . Secara keseluruhan, penggunaan algoritma SVM dalam mengklasifikasikan sentimen pengguna aplikasi MyTelkomsel menunjukkan hasil yang baik, dan dapat dijadikan acuan optimalisasi kualitas aplikasi berdasarkan persepsi dan masukan berdasarkan ulasan pengguna.

Referensi

- [1] F. S. Zainurrisalah, S. Supian, and A. Triska, 'Pemilihan Operator Seluler untuk Paket Internet di Kalangan Mahasiswa Universitas Padjadjaran dengan Metode *Analytical Hierarchy Process (AHP)*', *jmi*, Vol. 20, No. 2, pp. 217–232, Dec. 2024, DOI: 10.24198/jmi.v20.n2.58725.217-232.
- [2] M. Haerunnissa, A. Priyanto, C. Asnawi, and N. Alfi Sa'diya, 'Analisis Sentimen Kepuasan Pelanggan Perusahaan Telekomunikasi Seluler Telkomsel di Twitter', *Teknomatika*, Vol. 15, No. 2, pp. 68–75, Oct. 2022, DOI: 10.30989/teknomatika.v15i2.1117.

- [3] D. Saputra and M. R. Pribadi, 'Analisis Sentimen Masyarakat terhadap Layanan Provider Internet di Indonesia menggunakan SVM', in *MDP Student Conference*, Apr. 2023, pp. 32–41. DOI: 10.35957/mdp-sc.v2i1.4554.
- [4] R. D. Wahyuni and A. N. Utomo, 'Penggunaan Metode *Lexicon* untuk Analisis Sentimen pada Ulasan Aplikasi KAI Access di *Google Play Store*', *Jurnal Rekayasa Informasi*, Vol. 11, No. 2, 2022.
- [5] A. A. L. Nugraha and S. Kacung, 'Analisis Sentimen Layanan Aplikasi MyTelkomsel menggunakan Metode *K-Nearest Neighbors*', *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, Vol. 7, No. 2, 2025.
- [6] R. Astuti, R. A. Husen, A. Triono, and M. K. Anam, 'Peningkatan Metode *Support Vector Machines (SVM)* pada Data *Child-free* menggunakan *Oversampling*', in *Prosiding-Seminar Nasional Teknologi Informasi & Ilmu Komputer (SEMATER)*, 2023.
- [7] P. Fremmuzar and A. Baita, 'Uji Kernel SVM dalam Analisis Sentimen terhadap Layanan Telkomsel di Media Sosial *Twitter*', *Komputika*, Vol. 12, No. 2, pp. 57–66, Sep. 2023, DOI: 10.34010/komputika.v12i2.9460.
- [8] 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*, Vol. 3, No. 2, pp. 153–160, Oct. 2023, DOI: 10.57152/malcom.v3i2.897.
- [9] M. F. Fernando, D. A. Ahmad, N. F. Rachmanto, S. S. M. Wara, and K. M. Hindrayani, 'Analisis Sentimen terhadap Ulasan Aplikasi Mobile JKN menggunakan Metode *Machine Learning Logistic Regression, SVM, dan CSVM*', *ESTIMASI: Journal of Statistics and Its Application*, Vol. 6, No. 2, pp. 213–225, Jul. 2025, DOI: 10.20956/ejsa.v6i2.44943.
- [10] R. P. Sahartira and F. N. Hasan, 'Analisis Sentimen Ulasan Pengguna Aplikasi MyTelkomsel pada *Google Playstore* menggunakan *Algoritma Naive Bayes*', *JITET*, Vol. 2, No. 1, 2025, DOI: 10.23960/jitet.
- [11] M. F. N. Fathoni, E. Y. Puspaningrum, and A. N. Sihananto, 'Perbandingan Performa *Labeling Lexicon InSet* dan *VADER* pada Analisa Sentimen Rohingya di Aplikasi X dengan *SVM*', *Modem*, Vol. 1, No. 3, pp. 62–76, Jul. 2024, DOI: 10.62951/modem.v2i3.112.
- [12] M. Handayani, R. Rosnelly, and H. Hartono, '*Classification of Basurek Batik using Pre-Trained VGG-16 and Support Vector Machine*', in *International Conference on Information Science and Technology Innovation (ICoSTEC)*, Mar. 2023, pp. 40–44. DOI: 10.35842/icostec.v2i1.34.
- [13] F. Sidik, I. Suhada, A. H. Anwar, and F. N. Hasan, 'Analisis Sentimen terhadap Pembelajaran Daring dengan *Algoritma Naive Bayes Classifier*', *JLK*, Vol. 5, No. 1, p. 34, Apr. 2022, DOI: 10.26418/jlk.v5i1.79.
- [14] Zulkarnain, N. Rice, Angraini, and Zarnelly, 'Klasifikasi Sentimen Layanan pada Aplikasi by.U menggunakan *Algoritma Support Vector Machine*', *Jurnal Sistem Informasi*, Vol. 14, No. 4, pp. 1967–1976, 2025.
- [15] G. Kamoda, B. Heinzerling, T. Inaba, K. Kudo, K. Sakaguchi, and K. Inui, '*Weight-based Analysis of Detokenization in Language Models: Understanding the First Stage of Inference without Inference*', in *Findings of the Association for Computational Linguistics: NAACL 2025*, Association for Computational Linguistics, 2025, pp. 6339–6358.
- [16] F. Koto, *InSet*. (Agustus 2019). Accessed: Apr. 21, 2026. [Online]. Available: <https://github.com/fajri91/InSet>
- [17] R. Rahmaddeni and F. Akbar, '*Comparison of Naïve Bayes Algorithm, Support Vector Machine and Decision Tree in Analyzing Public Opinion on COVID-19 Vaccination in Indonesia*', *IJAIDM*, Vol. 6, No. 1, p. 8, Apr. 2023, DOI: 10.24014/ijaidm.v6i1.19966.
- [18] Tarwoto, R. Nugroho, N. Azka, and W. S. R. Graha, 'Analisis Sentimen Ulasan Aplikasi Mobile JKN di *Google PlayStore* menggunakan IndoBERT', *jtik*, Vol. 9, No. 2, pp. 495–505, Jan. 2025, DOI: 10.35870/jtik.v9i2.3340.
- [19] D. D. H. Asih and L. P. Manik, 'Analisis Sentimen Ulasan Review Aplikasi MyTelkomsel, IndosatM3 dan MyXl di *Google Play Store* menggunakan Metode *Bert*', *INTECOMS*, Vol. 8, No. 1, pp. 58–67, Jan. 2025, DOI: 10.31539/intecom.v8i1.12817.