Klasifikasi Sentimen Ulasan Pengguna terhadap Aplikasi Video Editing Play Store Menggunakan Random Forest

Sentiment Classification of User Reviews on Video Editing Applications from the Play Store Using Random Forest

Hafizah Wadji Raff[1], Fajar Ratnawati[2]
[1,2] Jurusan Teknik Informatika, Politeknik Negeri Bengkalis
E-mail: [1]hafizahwadjiraff@gmail.com, [2]fajar@polbeng.ac.id
Received 26 May 2025; Revised 4 June 2025; Accepted 9 June 2025

Abstrak - Kemajuan teknologi mendorong meningkatnya penggunaan aplikasi *video editing* pada *Google Play Store,* sehingga menginterpretasikan ulasan dari pengguna secara akurat menjadi sangat penting. Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna terhadap aplikasi CapCut, InShot, Kinemaster, dan Filmora. Data ulasan diperoleh melalui *web scraping* dan selanjutnya diproses dengan metode *TF-IDF* untuk pembobotan kata. Dataset dipisahkan menjadi dua bagian, yakni 80% untuk pelatihan dan 20% untuk pengujian. *Random Forest* digunakan sebagai algoritma utama dalam pembuatan model klasifikasi sentimen, sementara penilaian performa model dilakukan melalui pengukuran akurasi, presisi, *recall,* dan *F1-score.* Temuan penelitian mengindikasikan algoritma *Random Forest* dapat mengelompokkan sentimen pengguna dengan tingkat akurasi mencapai 84% dari total 4.000 data ulasan di setiap aplikasi. Temuan ini mengindikasikan *Random Forest* dinilai handal dalam menangani dan menginterpretasikan opini yang diberikan oleh pengguna terhadap aplikasi *video editing*. Kesimpulan ini memperkuat peran analisis sentimen dalam pengembangan dan peningkatan kualitas aplikasi berbasis ulasan pengguna di platform digital.

Kata kunci: Analisis Sentimen, Random Forest, TF-IDF, Aplikasi Video Editing, Play Store.

Abstract *- Technological advancements have driven the increased use of video editing applications on the Google Play Store, making accurate interpretation of user reviews highly important. This study aims to analyze the sentiment of user reviews for the CapCut, InShot, Kinemaster, and Filmora applications. Review data was collected through web scraping and subsequently processed using the TF-IDF method for word weighting. The dataset was split into two parts: 80% for training and 20% for testing. Random Forest was employed as the primary algorithm for building the sentiment classification model, with model performance evaluated using accuracy, precision, recall, and F1-score metrics. The findings indicate that the Random Forest algorithm can classify user sentiments with an accuracy of up to 84% from 4,000 reviews for each application. The results confirm the reliability of Random Forest in processing and interpreting user feedback regarding video editing applications. This conclusion reinforces the role of sentiment analysis in the development and improvement of application quality based on user reviews on digital platforms.*

Keywords: Sentiment Analysis, Random Forest, TF-IDF, Video Editing Applications, Play Store.

1. PENDAHULUAN

Kemajuan teknologi digital telah menghasilkan transformasi signifikan dalam metode komunikasi dan pertukaran informasi antar manusia, khususnya melalui internet dan media sosial yang tercermin dari peningkatan pengguna layanan internet di Indonesia menjadi 215,63 juta di tahun 2024, naik sebesar 2,67% dibandingkan tahun sebelumnya [1]. Fenomena ini mendorong berkembangnya beragam layanan digital yang memungkinkan orang untuk saling bertukar informasi dan membagikan pengalaman, termasuk melalui platform media sosial [2].

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

Pengguna dapat membagikan pengalaman mereka dalam bentuk foto, video, dan konten visual lainnya. Peningkatan minat berbagi konten video menciptakan kebutuhan aplikasi pengeditan video yang efisien, karena konten video dianggap lebih menarik dan mampu meningkatkan keterlibatan audiens dibandingkan format lain [3].

Kemudahan akses terhadap perangkat seluler dan aplikasi digital juga mendorong masyarakat untuk lebih aktif menghasilkan dan membagikan konten visual. Aplikasi pengeditan video tersedia di perangkat Android memungkinkan pengguna dapat mengedit dengan mudah dan cepat tanpa memerlukan komputer khusus, di mana pun dan kapan pun [4]. Ulasan serta penilaian dari pengguna aplikasi tersebut menyediakan informasi berharga untuk mengetahui tingkat kepuasan dan harapan mereka.

Google Play Store sebagai platform distribusi aplikasi pada sistem operasi Android menjadi sumber utama bagi pengguna dalam mengunduh aplikasi pengeditan video. Dalam penelitian ini, *Google Play Store* dimanfaatkan sebagai sumber data untuk mengumpulkan ulasan pengguna terkait aplikasi video editing. Analisis sentimen merupakan aplikasi *text mining* yang bertujuan untuk memahami opini pengguna melalui pemrosesan data teks [5]. Sentimen yang dianalisis diklasifikasikan menjadi kategori positif, negatif, maupun netral. Analisis ini banyak digunakan untuk pengembang dalam menilai efektivitas aplikasi melalui sudut pandang dari pengguna [6].

Algoritma *Random Forest* termasuk kedalam metode *supervised learning* yang diaplikasikan sebagai klasifikasi serta regresi. Algoritma ini digunakan pada penelitian sebagai penentuan kategori sentimen dari ulasan para pengguna. Proses ini dilakukan dengan membentuk pohon *ensemble.* Setiap pohon dalam *Random Forest* dibentuk dari subset data latih yang unik dengan pemilihan *split node* berdasarkan perhitungan *Gini Index* memilih nilai minimum antar kelas, demi memperoleh *gain* terbesar [7].

Studi sebelumnya menganalisis sentimen pengguna aplikasi Ruangguru menggunakan metode *Random Forest, Support Vector Machine,* dan *Naive Bayes* dengan nilai terbaik diperoleh *Random Forest* yang mencapai nilai sebesar 97,16% [8]. Penggunaan *Random Forest* sebagai analisis sentimen terhadap aplikasi pelayanan publik dan berhasil mencapai akurasi sebesar 84% setelah dilakukan penyetelan *hyperparameter* [9]. Perbandingan antara algoritma *Random Forest* dan *decision tree* pada klasifikasi sentimen ulasan jasa pengiriman kurir memperlihatkan *Random Forest* mempunyai akurasi yang lebih tinggi [10]. Penggunaan *Random Forest* sebagai analisis sentimen pada aplikasi dompet digital dan berhasil memperoleh akurasi sebesar 89,02% [11]. Penggunaan *Random Forest* diterapkan untuk analisis sentimen ulasan pengguna pada aplikasi Sirekap di *Play Store* dengan nilai 74% [12]. Penerapan *Naive Bayes* digunakan sebagai analisis sentimen ulasan pengguna terhadap aplikasi CapCut pada *Play Store* dengan hasil bahwa sentimen negatif sedikit lebih dominan [13]. Analisis sentimen ulasan pada aplikasi Genshin Impact di *Play Store* dengan *Random Forest* berhasil mencapai akurasi sebesar 92% [14].

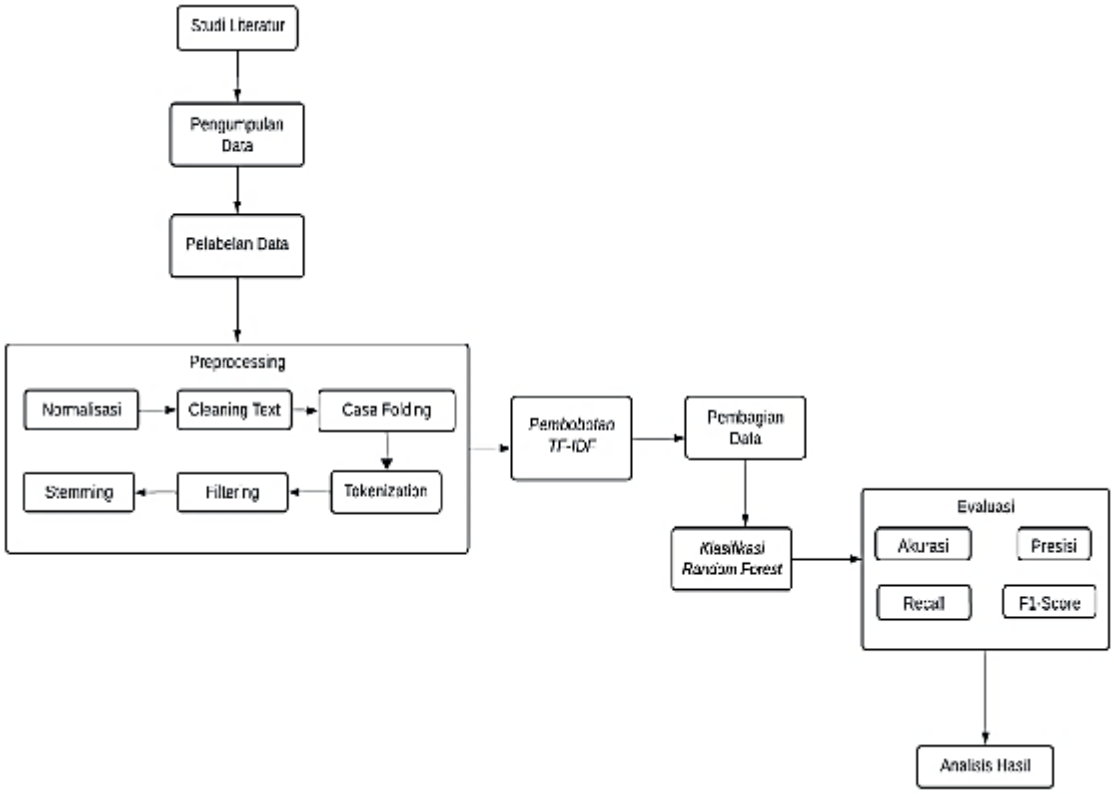
Berdasarkan studi-studi sebelumnya, algoritma *machine learning*, khususnya *Random Forest,* terbukti efektif dalam analisis sentimen dengan akurasi tinggi pada berbagai aplikasi. Namun, penelitian yang secara khusus menganalisis sentimen ulasan pengguna terhadap aplikasi video editing di *Play Store* masih terbatas, padahal permintaan konten visual terus meningkat seiring berkembangnya media sosial. Oleh karena itu, penelitian ini bertujuan menerapkan algoritma *Random Forest* untuk menganalisis sentimen pengguna aplikasi video editing di *Play Store.*

2. METODE PENELITIAN

Proses penelitian ini mengikuti beberapa tahapan utama, mulai dari studi literatur, pengumpulan data, pelabelan data, prapemrosesan data, pembobotan kata dengan *Term Frequency Inverse Document Frequency* (TF-IDF), hingga klasifikasi dengan menerapkan algoritma *Random Forest*. Kinerja klasifikasi dievaluasi melalui metrik presisi, akurasi, *recall*,

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

serta *F1-score,* yang kemudian dianalisis lebih lanjut. Alur penelitian secara keseluruhan dapat dilihat pada Gambar 1.



Gambar 1 Alur Penelitian

2.1 Studi Literatur

Penelitian ini dilaksanakan dengan mengumpulkan serta menyeleksi secara menyeluruh referensi-referensi terkait dan relevan, baik dari teori maupun penelitian sebelumnya yang memiliki hubungan erat dengan topik. Sumber referensi yang digunakan meliputi buku, jurnal, dan situs web yang membahas tentang *sentiment analysis*, *data mining*, *random forest, text mining*, serta pendekatan *TF-IDF* untuk pemberian bobot kata.

2.2 Pengumpulan Data

Data penelitian dikumpulkan dari *Google Play Store* menggunakan bantuan *google-play-scraper* sebagai alat *web scraping* pada April 2024. Sebanyak 4.000 ulasan aplikasi video editing populer, seperti CapCut, InShot, Kinemaster, dan Filmora, diambil sebagai sampel. Setiap ulasan berisi teks dan label sentimen (positif/negatif) berdasarkan penilaian bintang pengguna.

2.3 Pelabelan Data

Untuk mengukur konsistensi antar pelabel, dilakukan perhitungan *inter-annotator agreement* menggunakan *Cohen's Kappa,* yang menunjukkan tingkat kesepakatan yang tidak disebabkan oleh kebetulan. *Cohen's Kappa* digunakan untuk mengukur tingkat kesepakatan antar dua pelabel dalam proses pelabelan data [15]. Metode ini memperhitungkan kesepakatan yang terjadi secara kebetulan, sehingga lebih akurat dibandingkan dengan persentase kesepakatan sederhana. Dengan rumus sebagai berikut.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

$$K = (Po - Pe)/(1 - Pe) \tag{1}$$

Keterangan:
Po : proporsi kesepakatan aktual antar pelabel
Pe : proporsi kesepakatan yang diharapkan terjadi secara kebetulan

2.4 Prapemrosesan Teks

Beberapa sub-tahapan dilakukan dalam fase ini untuk memastikan data siap digunakan oleh model *Machine Learning*. Proses prapemrosesan mencakup beberapa langkah berikut.

1. Normalisasi: Semua karakter teks diubah menjadi huruf kecil guna menjaga konsistensi data.
2. Pembersihan Teks: Menghapus elemen-elemen yang tidak relevan dalam teks seperti tanda baca, angka, dan simbol khusus.
3. *Case Folding:* Mengonversi seluruh karakter teks ke dalam huruf kecil untuk konsistensi.
4. *Tokenization:* Memisahkan teks menjadi elemen-elemen kata yang disebut token.
5. *Filtering:* Mengeliminasi kata-kata umum yang kurang berkontribusi pada hasil analisis.
6. *Stemming:* Proses mengonversi kata-kata ke bentuk dasar atau akar katanya.

2.5 Pembobotan Kata (TF-IDF)

Tahap pembobotan kata bertujuan memberikan skor pada tiap kata untuk menemukan *frequency* kemunculan *term* dan dikalikan dengan *inverse* dari *frequency* dokumen [16]. Data praproses diubah ke format numerik menggunakan *TF-IDF*, yaitu hasil kali antara frekuensi istilah dalam dokumen (TF) dan tingkat kelangkaannya dalam seluruh dokumen (IDF) [17]. Pembobotan menggunakan *TF-IDF* dijelaskan pada persamaan berikut.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \tag{2}$$

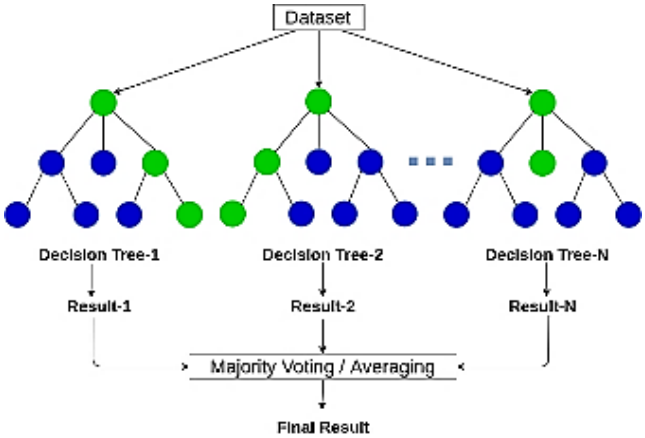
Keterangan:
$tf_{i,j}$: Jumlah kata i pada dokumen j
N : Total dokumen
df_i : Jumlah dokumen mengandung kata ke i

2.6 Klasifikasi *Random Forest*

Random Forest ialah metode *ensemble* menggabungkan beberapa *decision tree* yang dilatih pada data dan fitur acak untuk meningkatkan akurasi. Berikut mekanisme kerjanya [18].

1. Persiapan Data: Pemberian label dan pembagian data untuk membentuk Pohon keputusan.
2. *Bootstrap Sampling (Bagging):* Membuat beberapa subset data dari dataset asli dengan pengambilan sampel secara acak, memungkinkan duplikasi sampel.
3. *Random Feature Selection:* Memilih secara acak subset fitur setiap node untuk membagi data.
4. *Decision Tree Construction*: Membangun pohon keputusan untuk setiap subset data menggunakan algoritma CART *(Classification and Regression Tree).*
5. *Voting* (*Agregasi*): Hasil prediksi dari semua pohon digabungkan. Untuk klasifikasi, digunakan *majority voting*; untuk regresi, digunakan rata-rata hasil prediksi.
   Gambar 2 berikut menyajikan visualisasi dari *Random Forest.*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873



Gambar 2 Skema *Random Forest*

Rumus *Entropy€:*

$$E(S) = \sum_{i=1}^{k} Pi\ log2\ (Pi) \tag{3}$$

Keterangan:
S : Sekumpulan sampel (contoh, ulasan dalam subset data)
Pi : *Probabilitas* kemunculan kelas iii (650issal, Positif atau Negatif)

Rumus *Information Gain (IG)*:

$$IG(S,A) = E(S) - \sum_{v \in Values(A)} \frac{|Sv|}{|S|} E(Sv) \tag{4}$$

Keterangan:
A : Fitur untuk memisahkan data.
$Values(A)$: Kumpulan nilai yang mungkin dari fitur A
Sv : Subset data S yang dipisahkan oleh fitur A yang memiliki nilai v
$|Sv|$: Jumlah data pada subset sv
$|S|$: Jumlah total data

2.7 Evaluasi dan Analisis Hasil

Evaluasi performa algoritma menggunakan empat metrik klasifikasi utama: akurasi, presisi, *recall,* dan *F1-score.* Dataset dibagi dengan metode *Stratified K-Fold Cross Validation* sebanyak lima fold untuk memastikan proporsi label yang seimbang di setiap fold [19]. Penjelasan metrik evaluasi sebagai berikut.

1. Akurasi: Proporsi prediksi yang tepat dibandingkan dengan seluruh hasil prediksi.
2. Presisi: Proporsi prediksi positif yang benar terhadap keseluruhan prediksi positif.
3. *Recall:* Proporsi prediksi tepat pada kelas positif terhadap seluruh data positif sebenarnya.
4. *F1-Score:* Kalkulasi rata-rata harmonis antara presisi dengan *recall.*

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Google Play Store merupakan sumber utama data ulasan pengguna aplikasi video editing untuk penelitian ini. Pengambilan data dilakukan dengan *web scraping* memakai *google-*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

play-scraper. Sebanyak 4.000 ulasan dari aplikasi populer seperti CapCut, InShot, Kinemaster, dan Filmora diambil sebagai sampel. Setiap ulasan berisi teks pengguna dan label sentimen positif atau negatif berdasarkan penilaian bintang. Berikut adalah *library* yang digunakan untuk pengumpulan data.

```
import pandas as pd

from google_play_scraper import reviews, Sort
```

3.2 Labelisasi Data

Berdasarkan hasil pelabelan oleh dua validator terhadap data ulasan aplikasi:

Po (proportion observed agreement) = 1.0
Pe (proportion expected agreement) = 0.0
K = (1.0 - 0.0) / (1 - 0.0) = 1.0

Nilai *Cohen's Kappa* = 1.0, yang menunjukkan bahwa kedua pelabel memiliki kesepakatan sempurna dalam proses pelabelan. Ini berarti tidak ada perbedaan klasifikasi antara pelabel 1 dan pelabel 2, sehingga hasil pelabelan dapat dianggap sangat konsisten dan reliabel.
Kriteria pelabelan meliputi:
Kata kunci positif: “bagus”, “cepat”, “mudah digunakan”, “efisien”.
Kata kunci negatif: “lemot”, “bug”, “sering crash”, “banyak iklan”.
Konteks kalimat: penilaian tidak hanya berdasarkan satu kata, tetapi juga dilihat dari keseluruhan makna ulasan, termasuk sarkasme atau kontradiksi seperti “awalnya bagus, tapi sekarang jelek”.
Contoh pelabelan: “Aplikasi sangat bagus dan fiturnya lengkap” adalah Positif
“Dulunya lancar, sekarang sering crash dan penuh iklan” adalah Negatif

	A	B	
1	username	text	sentimen
2	Pengguna Google	Makin lama makin KIKIR aja developer capcut. Fitur yang dul	negative
3	Pengguna Google	"Tidak dapat dimuat, ketuk untuk coba lagi." perbaiki tuh bug	negative
4	Pengguna Google	aplikasi ini sudah lumayan bagus tapi, semenjak capcut punyi	negative
5	Pengguna Google	Terbaik, saya minta tolong untuk tambahan fitur pisah suara	negative

Gambar 3 Hasil Pelabelan Validator 1

	A	B	
1	username	text	sentimen
2	Pengguna Google	Makin lama makin KIKIR aja developer capcut. Fitur yang dul	negative
3	Pengguna Google	"Tidak dapat dimuat, ketuk untuk coba lagi." perbaiki tuh bug	negative
4	Pengguna Google	aplikasi ini sudah lumayan bagus tapi, semenjak capcut punyi	negative
5	Pengguna Google	Terbaik, saya minta tolong untuk tambahan fitur pisah suara	negative

Gambar 4 Hasil Pelabelan Validator 2

3.3 Implementasi Prapemrosesan Teks

Pada tahap Prapemrosesan ini sistem akan mengubah kalimat yang diinput menjadi bahasa alami. *Import library* yang akan digunakan yaitu:

```
import pandas as pd
import re
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

Hasil prapemrosesan teks dari CapCut, InShot, Kinemaster, dan Filmora ditampilkan pada Gambar 5 sampai Gambar 8, berikut.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

username	comment	processed_comment
Pengguna Google	Makin lama makin KIKIR aja developer capcut. Fitur yang dulunya gratis setelah update malah jadi premium, sering terjadi bug seperti durasi video yg awalnya 10 mnt bisa jadi 400+ jam (gak bisa dipotong), keluar bentar progres langsung ilang, banyak iklan gak jelas, sering terjadi lag, dan masih banyak lagi. MOHON DIPERBAIKI LAGI 🙏	makin lama makin kikir aja developer capcut fitur dulunya gratis update malah jadi premium sering jadi bug durasi video yg awal mnt jadi jam gak potong keluar bentar progres langsung ilang banyak iklan gak jelas sering jadi lag masih banyak mohon baik

Gambar 5 Hasil Prapemrosesan CapCut

username	comment	processed_comment
Pengguna Google	Cukup memuaskan mudah mengedit video apa saja foto dll kalo bisa tambahim fitur teks yg gak hilang hanya itu saja dulu lain kali kalo ada yg kurang di infokan lagi . Trims pengembang 👍🙏	cukup muas mudah edit video apa foto kalo tambahim fitur teks yg gak hilang itu dulu kali kalo yg kurang info trims kembang

Gambar 6 Hasil Prapemrosesan InShot

username	comment	processed_comment
Pengguna Google	Udah berlangganan, cuman banyak fitur yang kurang, contohnya percepat audionya saja 1x atau seterusnya tidak ada, jadi harus menggunakan aplikasi lain dulu agar mempercepat audio itu, divideo ada fitur itu tapi waktu audio atau musiknya tidak ada. Semoga next ditambah fitur itu	udah langgan cuman banyak fitur kurang contoh cepat audio x terus jadi guna aplikasi dulu cepat audio video ada fitur tapi waktu audio musik ada moga next tambah fitur

Gambar 7 Hasil Prapemrosesan Kinemaster

username	comment	processed_comment
Pengguna Google	Ada bug dan kekurangan di versi pro -Seperti di fitur tekstovideoAI gk muncul gambar atau blackscreen. -jika nambah PIP/gambar suara narasi AI nya berhenti -jika nambah clip baru lapisan²nya jadi acak-acakan -di tekstovideo script yg di masukan ke promth gk bisa pakai script asli harus pakai script yg di rubah. -clip yg di tambah in animation terkadang hasilnya gk lancar Tambahan: Tolong segera tambah suara narasi AI bahasa Indonesia	bug kurang versi pro fitur tekstovideoai gk muncul gambar blackscreen nambah pipgambar suara narasi ai nya henti nambah clip baru lapis jadi acakacakan tekstovideo script yg masuk promth gk pakai script asli pakai script yg rubah clip yg di tambah in animation terkadang hasil gk lancar tambah segera tambah suara narasi ai bahasa indonesia

Gambar 8 Hasil Prapemrosesan Filmora

3.4 Pelatihan *Model Random Forest*

1. Import *library* yang digunakan

```
import pandas as pd
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from imblearn.over_sampling import SMOTE

columns=tfidf_vectorizer.get_feature_names_out())

# Menyimpan hasil TF-IDF ke file CSV baru
tfidf_output_file = "capcut_reviews_tfidf.csv"
tfidf_df.to_csv(tfidf_output_file, index=False, encoding="utf-8-sig")
```

Berbagai library seperti *pandas*, *train_test_split*, *TfidfVectorizer*, dan *RandomForestClassifier* digunakan dalam proses pelatihan model. Untuk mengatasi ketidakseimbangan data, diterapkan metode SMOTE pada data latih. Teks dikonversi ke vektor dengan *TF-IDF*, lalu dilatih memakai *Random Forest* dan dievaluasi dengan *classification_report*.

2. Membaca data set

```
data = pd.read_csv("data.csv")
df = pd.DataFrame(data)
```

3. Pembagian data

```
# Split data latih dan uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
stratify=y, random_state=42)
```


e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

Dataset dipisahkan menjadi data pelatihan sebesar 80% dan data pengujian sebesar 20% memakai fungsi *train_test_split*, dengan *stratify* untuk mempertahankan proporsi kelas dan *random_state* guna menjamin hasil yang konsisten.

4. Pembobotan TF-IDF

```
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
```

Teks dikonversi ke bentuk numerik dengan *TF-IDF*, di mana data latih dipelajari dan ditransformasikan, lalu digunakan untuk mentransformasi data uji tanpa pelatihan ulang. Beberapa contoh ulasan yang digunakan dalam pembobotan antara lain:

D1: Pengguna mengeluhkan fitur gratis yang berubah menjadi premium, *bug* durasi video, progres yang hilang saat keluar aplikasi, serta banyaknya iklan dan *lag*.

D2: Ulasan menyebutkan kegagalan memuat template meski jaringan stabil, membuat aplikasi dianggap tidak berguna.

D3: Aplikasi dinilai bagus, namun terlalu banyak iklan saat ekspor dan perubahan bentuk template membuat pengguna tidak nyaman.

D4: Pengguna keberatan karena banyak efek menjadi berbayar, serta berharap *watermark* dapat dihilangkan untuk keperluan lomba.

D5: Aplikasi disukai karena fitur dan template menarik, namun iklan berlebihan dan durasi video terbatas menurunkan rating.

Tabel 1 memperlihatkan pembobotan kata yang digunakan.

Tabel 1 Pembobotan Kata

No	Kata	TF(Positif)	TF(Negatif)
1.	Makin	1	0
2.	Lama	1	0
3.	Pelit	0	1
4.	Developer	1	0
5.	CapCut	6	0
6.	Fitur	1	0
7.	Gratis	0	1
8.	Update	1	0
9.	Premium	0	1
10.	Bug	0	3
11.	Durasi	1	0
12.	Video	2	0
13.	Awal	1	0
14.	Bisa	7	0
15.	Jadi	1	0
16.	Jam	1	0
17.	Potong	1	0
18.	Keluar	1	0
19.	Sebentar	1	0
20.	Progres	1	0
21.	Langsung	1	0
Total:		**29**	**6**

5. Mengatasi data tidak seimbang dengan *SMOTE*

SMOTE digunakan pada data latih untuk mengatasi ketidakseimbangan kelas dengan menambah sampel pada kelas minoritas.

```
# Mengatasi data tidak seimbang menggunakan SMOTE
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train_tfidf,
y_train)
```

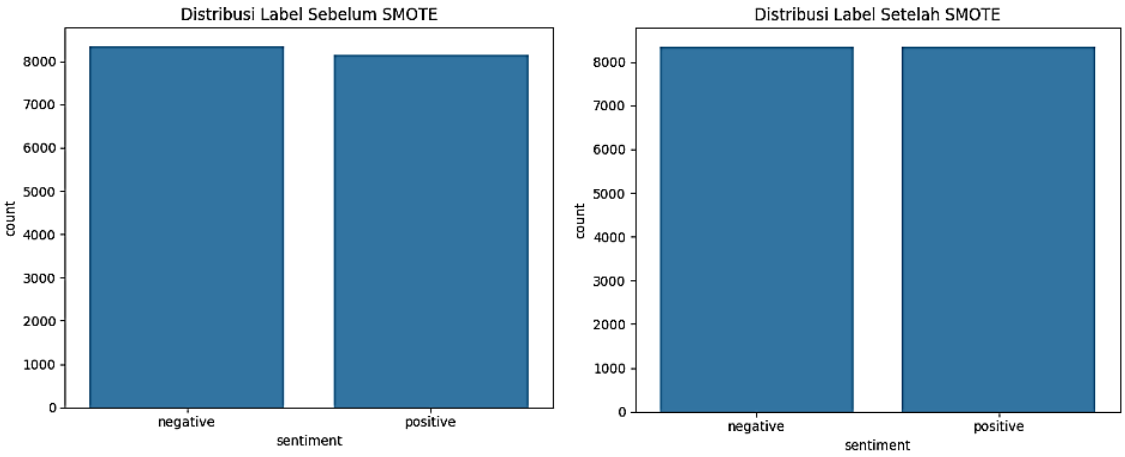
Distribusi Sebelum *SMOTE:*

sentiment negative: 8347

positive: 8134

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

Distribusi Setelah *SMOTE:*
sentiment negative: 8347
positive: 8347



Gambar 9 Diagram *SMOTE*

6. Melatih model

```
# Melatih model dengan RandomizedSearchCV
random_search.fit(X_train_resampled, y_train_resampled)
# Model terbaik
best_model = random_search.best_estimator_
```

Model dilatih menggunakan data yang telah diseimbangkan dengan *SMOTE,* dan model terbaik diperoleh melalui *random search* dan disimpan untuk prediksi selanjutnya.

7. Evaluasi model

```
# K-Fold Cross Validation
kf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
# K-Fold Cross Validation
kf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Classification Report
print(f"\n=== Classification Report Fold {fold} ===")
print(classification_report(y_test, y_pred, digits=2))

fold += 1
```

Pada kode ini *shuffle=True* dan *random_state*=42, data diacak terlebih dahulu sebelum dibagi, dan hasil pembagian akan konsisten setiap kali dijalankan. Dengan menggunakan fungsi *classification_report* dari *library scikit-learn*, kode ini mencetak metrik evaluasi seperti *precision, recall, f1-score*, dan *support* untuk data uji *(y_test)* dan prediksi model *(y_pred). Parameter digits*=2 mengatur jumlah digit desimal pada hasil output. Berikut hasil *clasification report* menggunakan *k-fold cross validation:*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

```
=== Classification Report Fold 1 ===
              precision    recall  f1-score   support

    negative       0.83      0.86      0.85      1670
    positive       0.85      0.82      0.84      1627

    accuracy                           0.84      3297
   macro avg       0.84      0.84      0.84      3297
weighted avg       0.84      0.84      0.84      3297
```

```
=== Classification Report Fold 2 ===
              precision    recall  f1-score   support

    negative       0.84      0.87      0.86      1670
    positive       0.87      0.82      0.84      1626

    accuracy                           0.85      3296
   macro avg       0.85      0.85      0.85      3296
weighted avg       0.85      0.85      0.85      3296
```

```
=== Classification Report Fold 3 ===
              precision    recall  f1-score   support

    negative       0.82      0.87      0.84      1669
    positive       0.86      0.80      0.83      1627

    accuracy                           0.84      3296
   macro avg       0.84      0.83      0.83      3296
weighted avg       0.84      0.84      0.83      3296
```

```
=== Classification Report Fold 4 ===
              precision    recall  f1-score   support

    negative       0.83      0.88      0.86      1669
    positive       0.87      0.82      0.85      1627

    accuracy                           0.85      3296
   macro avg       0.85      0.85      0.85      3296
weighted avg       0.85      0.85      0.85      3296
```

```
=== Classification Report Fold 5 ===
              precision    recall  f1-score   support

    negative       0.82      0.87      0.84      1669
    positive       0.86      0.81      0.83      1627

    accuracy                           0.84      3296
   macro avg       0.84      0.84      0.84      3296
weighted avg       0.84      0.84      0.84      3296
```

```
=== Hasil Evaluasi Rata-rata (5-Fold) ===
Akurasi   : 0.8432 ± 0.0067
Presisi   : 0.8441 ± 0.0065
Recall    : 0.8432 ± 0.0067
F1 Score  : 0.8430 ± 0.0067
```

Gambar 10 Hasil Evaluasi Model

Berdasarkan hasil evaluasi model menggunakan metode *5-fold stratified cross validation,* kinerja model menunjukkan performa yang cukup stabil dan konsisten pada setiap *fold*. Setiap fold menghasilkan nilai akurasi yang berkisar antara 0.84 hingga 0.85, dengan rata-rata akurasi sebesar 0.84 ± (standar deviasi). *Precision* dan *recall* pada masing-masing *fold* juga menunjukkan keseimbangan yang baik, dengan nilai rata-rata *precision* sebesar 0.84, *recall* sebesar 0.84, dan *F1-score* sebesar 0.84.

8. Simpan model

```python
import pickle
# Simpan model ke dalam file
with open("model.pkl", "wb") as model_file:
    pickle.dump(model, model_file)
# Simpan TF-IDF vectorizer ke dalam file
with open("tfidf.pkl", "wb") as tfidf_file:
    pickle.dump(tfidf, tfidf_file)
print("Model dan TF-IDF vectorizer berhasil disimpan!")
```

Kode ini menyimpan *model* machine learning dan *TF-IDF vectorizer* ke file menggunakan modul *pickle*, dengan file *model.pkl* untuk *model* dan *tfidf.pkl* untuk *vectorizer.*

4. KESIMPULAN DAN SARAN

Studi ini berhasil mengklasifikasikan sentimen pengguna dengan akurasi sebesar 84% dari total 4000 data ulasan pada masing-masing aplikasi. Temuan ini menunjukkan bahwa *Random Forest* merupakan metode yang efektif dalam mengolah dan memahami opini

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

pengguna terhadap aplikasi video editing. Kesimpulan ini memperkuat peran *analisis sentimen* dalam pengembangan dan peningkatan kualitas aplikasi berbasis ulasan pengguna di platform digital. Penelitian ini dapat diperluas lebih lanjut dengan memperluas cakupan data ulasan serta mengeksplorasi algoritma lain seperti *Support Vector Machine*, *Naive Bayes*, atau pendekatan *deep learning* guna meningkatkan kemampuan generalisasi dan akurasi prediksi.

DAFTAR PUSTAKA

[1] E. Hulasoh, C. Sulastri, dan D. T. Andini, "Edukasi Menuju Masyarakat yang Cerdas dan Bijak dalam Pemanfaatan Teknologi Informasi dan Komunikasi di Kelurahan Ciater, Kecamatan Serpong, Tangerang Selatan," *Jurnal PKM COMMs*, vol. 1, no. 2, pp. 32–34, 2023.

[2] A. Damayanti, I. D. Delima, dan A. Suseno, "Pemanfaatan Media Sosial Sebagai Media Informasi dan Publikasi (studi deskriptif kualitatif pada akun Instagram @rumahkimkotatangerang)," *PIKMA: Jurnal Pendidikan Ilmu Komputer dan Matematika*, vol. 6, no. 1, pp. 173–190, Sep. 2023. doi: 10.24076/pikma.v6i1.1308.

[3] A. Saleem, R. Mehmood, A. Taj, M. U. Khalid, A. Moiz, dan A. Lakho, "Impact of Video Content Marketing on Consumer, Engagement," *Journa l of Policy Research*, vol. 10, no. 3, pp. 83–95, 2024. doi: 10.61506/02.00322.

[4] N. A. Shafie dan A. N. A Zulkifli "Review of the Lazada Application on Google Play Store: Sentiment Analysis," *Journal of, Computing Research and Innovation*, vol. 9, no. 1, pp. 43–55, 2024. doi: 10.24191/jcrinn.v9i1.412.

[5] P. Nandwani dan R. Verma, "A, Review on Sentiment Analysis and Emotion Detection from Text," *Social Network, Analysis and Mining*, vol. 11, no 1, p. 81, 2021. doi: 10.1007/s13278-021-00776-6.

[6] G. Brahmani, I. Maheen, K. Apoorva, dan M. Bharathi, "Appraisal or Critique? Analyzing Sentiments in Google Play Store Reviews with Python," *Journal of Data Structures and Computing*, vol. 1, no. 1, pp. 13–17, 2024. doi: 10.48001/jodsc.2024.1113-17.

[7] T. Tangirala, "Evaluating the Impact of, GINI Index and Information gain on, Classification using Decision Tree Classifier, Algorithm," *International Journal of Advanced Computer Science and Applications* vol. 11, no. 2, pp. 612–617, 2020. doi: 10.14569/IJACSA.2020.0110277.

[8] E. Fitri, "Analisis Sentimen Terhadap Aplikasi, Ruangguru Menggunakan Algoritma Naive Bayes, Random, Forest Dan Support Vector r Machine," *Jurnal Transformtika*, vol. 18, no. 1, pp. 71–80, Jul. 2020. doi: 10.26623/transformatika. v18i1.2317

[9] I. N. A. K. Ardika dan I. G. A. Wibawa, "Analisis Sentimen Ulasan Pengguna Aplikasi Pelayanan Masyarakat dengan Menggunakan Algoritma Random Forest," *Jurnal Nasional Teknologi Informasi dan Aplikasinya (JNATIA)*, vol. 1, no. 1, pp. 361–372, Nov. 2022. [Online]. Available: https://jurnal.harianregional.com/jnatia/id-92585.

[10] D. N. I. Huda, C. Prianto, dan R. M. Awangga, "Analisis Sentimen Perbandingan Layanan Jasa Pengiriman Kurir Pada Ulasan Play Store Menggunakan Metode Decision Tree dan Random Forest," *Jurnal Ilmiah Informatika*, vol. 11, no. 2, pp. 150–158, 2023. doi: 10.33884/jif.v11i02.7952.

[11] N. A. Hapsari dan A. D. Indriyanti, "Analisis Sentimen pada Aplikasi Dompet Digital Menggunakan Algoritma Random Forest," *Journal of Emerging Information System and Business Intelligence (JEISBI)*, vol. 4, no. 3, pp. 186–192, 2023. doi: 10.26740/jeisbi.v4i3.55696.

[12] M. F. Y. Herjanto dan C. Carudin, "Analisis Sentimen Ulasan Pengguna Aplikasi sirekap pada Play Store Menggunakan Algoritma Random Forest Classifier," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, pp. 1204–1210, Apr. 2024. [Online]. Available: http://dx.doi.org/10.23960/jitet.v12i2.4192.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873

[13] Meliyawati dan F. N. Hasan, "Analisis Sentimen Pengguna Aplikasi CapCut Pada Ulasan di Play Store Menggunakan Metode Naïve Bayes," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 4, pp. 2272–2280, 2024. doi: 10.30865/klik.v4i4.1555.

[14] D. R. Ananda dan Y. F. Riti, "Analisis Sentimen Ulasan Aplikasi Genshin Impact di Play Store Menggunakan Random Forest," *Prosiding Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika (SNESTIK)*, vol. 1, no. 1, pp. 578–586, Jun. 2024. doi: 10.31284/p.snestik.2024.5883.

[15] J. Mohanan, "How to use Cohen's Kappa Statistic for ML Model verification," *Medium*, 2 Mei 2024. [Online]. Available: https://medium.com/@jayamohanmohanan/how-to-use-cohens-kappa-statistic-for-ml-model-verification-6c66258b4ae9.

[16] F. A. Larasati, D. E. Ratnawati, dan B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 9, pp. 4305–4313, Sep. 2022. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11562.

[17] N. A. Hapsari dan A. D. Indriyanti, "Analisis Sentimen pada Aplikasi Dompet Digital Menggunakan Algoritma Random Forest," *Jurnal Ekonomi dan Ilmu Sosial Bisnis*, vol. 6, no. 9, pp. 4305–4313, 2022. [Online]. Available: http://j-ptiik.ub.ac.id.

[18] I. Saputra, "Data mining 5: Klasifikasi decision tree dan, random forest," Slide Share, 29 Agustus 2024. [Online]. Available: https://www.slideshare.net/slideshow/data-mining-5-klasifikasi-decision-tree-dan-random-forest/250193533.

[19] S. Widodo, H. Brawijaya, dan S. Samudi, "Stratified K-fold cross validation optimization on machine learning for prediction," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 6, no. 4, pp. 2407–2414, 2022. [Online]. Available: https://doi.org/10.33395/sinkron.v7i4.11792.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI : 10.62411/tc.v24i3.12873