LINIER
LITERATUR INFORMATIKA & KOMPUTER
e-ISSN: 3063-2218
DOI : 10.33096/linier.v2i3.3149
Literatur Informatika & Komputer
Vol 2, No 3 (2025), pp 380-389

Analisis Sentimen Review Aplikasi di Google Play Store Menggunakan *Random Forest*

Muhammad Faiq Rahmatullah[a], Poetri Lestari Lokapitasari Belluano[b], Herdianti Darwis[c]
Universitas Muslim Indonesia, Makassar, Indonesia
[a]13020200029@umi.ac.id; [b]poetrilestari@umi.ac.id; [c]herdianti.darwis@umi.ac.id

Received: 12-08-2025 | Revised: 20-08-2025 | Accepted: 14-09-2025 | Published: 29-09-2025

Abstrak

Google Play Store adalah salah satu platform distribusi aplikasi terbesar yang memungkinkan pengguna memberikan ulasan terhadap aplikasi yang mereka pakai. Di era digital saat ini, ulasan pengguna menjadi sumber data penting untuk menilai performa dan kualitas aplikasi. Namun, banyaknya jumlah ulasan membuat analisis secara manual menjadi kurang efisien. Oleh karena itu, peracangan ini ini mengadopsi pendekatan *machine learning* untuk mengklasifikasikan ulasan ke dalam kategori sentimen positif, negatif, atau netral. Proses analisis meliputi beberapa tahap, seperti pengumpulan data, praproses teks, ekstraksi fitur dengan TF-IDF, pelatihan model menggunakan *Random Forest,* serta evaluasi kinerja model. Hasil evaluasi menunjukkan bahwa model yang dikembangkan berhasil mengklasifikasikan sentimen dengan akurasi sebesar 68.5%, dengan performa terbaik pada sentimen negatif. Selain itu, penerapan metode *Random Forest* juga membuka peluang untuk pengembangan sistem analitik otomatis yang dapat digunakan oleh pengembang aplikasi dalam meningkatkan kualitas layanan mereka. Dengan memahami kecenderungan opini pengguna secara cepat dan akurat, pengambilan keputusan dalam pengembangan fitur baru atau perbaikan *bug* dapat dilakukan secara lebih terarah. Implementasi metode ini juga berpotensi untuk diterapkan pada sektor lain seperti *e-commerce*, layanan publik, atau media sosial, di mana opini pengguna menjadi salah satu aspek penting dalam evaluasi layanan.

Kata kunci**:** Analisis Sentimen, *Google Play Store*, *Random Forest*

Pendahuluan

Google Play Store merupakan platform distribusi aplikasi terbesar yang memungkinkan pengguna mengunduh, menilai, dan mengulas aplikasi. Ulasan tersebut mencerminkan opini pengguna yang berguna bagi pengembang dalam memahami kepuasan dan kebutuhan pengguna [1]. Namun, karena banyaknya ulasan, pengembang kesulitan menganalisis sentimen secara manual. Oleh karena itu, diperlukan metode otomatis untuk mengklasifikasikan sentimen guna memberikan wawasan yang lebih cepat dan akurat terkait performa aplikasi [2].

Penelitian ini menerapkan analisis sentimen pada ulasan aplikasi di Google Play Store menggunakan algoritma *Random Forest*, salah satu metode *machine learning* berbasis *ensemble learning* [3]. Algoritma ini dipilih karena efektif dalam mengolah data teks serta menghasilkan model yang stabil dan akurat [4]. Tahapan analisis meliputi *preprocessing* teks (pembersihan, tokenisasi, penghapusan *stopwords*, dan *stemming*), ekstraksi fitur dengan TF-IDF, serta klasifikasi sentimen ke dalam tiga kategori: positif, negatif, dan netral menggunakan *Random Forest* [5].

Random Forest dipilih karena keunggulannya dalam klasifikasi teks, khususnya untuk analisis sentimen [6]. Dibandingkan *Decision Tree*, metode ini lebih stabil dan mampu mengatasi overfitting melalui kombinasi beberapa pohon keputusan[7]. Sementara jika dibandingkan dengan *Naïve Bayes* dan *Support Vector Machine* (SVM), *Random Forest* lebih fleksibel dalam menangani data teks kompleks dan tetap efektif meski data tidak seimbang [8]. Selain itu, metode ini efisien secara komputasi karena mendukung paralelisasi, sehingga lebih cepat daripada model deep learning seperti LSTM dan BERT. Keunggulan tersebut menjadikan *Random Forest* pilihan tepat untuk analisis sentimen ulasan aplikasi di Google Play Store [9].

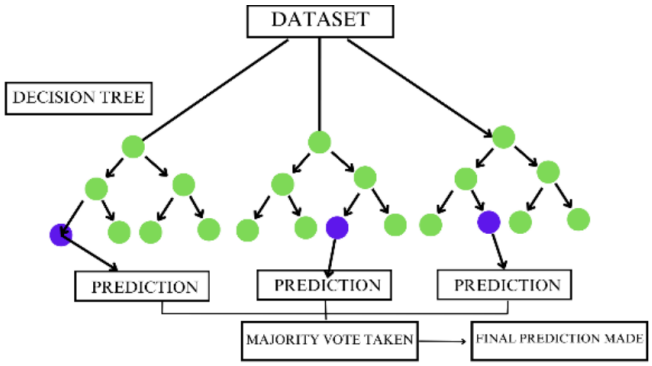
Beberapa studi sebelumnya telah menerapkan beragam pendekatan untuk analisis sentimen ulasan aplikasi. Salah satu penelitian menunjukkan bahwa algoritma *Random Forest* mampu mencapai akurasi 79% [10]. Sementara studi lainnya bahkan mencapai 90% [11]. Temuan-temuan ini menegaskan bahwa *Random Forest* merupakan opsi yang menjanjikan dalam menjaga keseimbangan antara akurasi dan efisiensi.

Perancangan ini bertujuan melakukan analisis sentimen terhadap ulasan pengguna aplikasi di Google Play Store menggunakan *Random Forest*, sekaligus mengevaluasi kinerjanya dalam mengklasifikasikan ulasan ke dalam tiga kategori. Selain itu, model ini dibandingkan dengan metode lain yang umum digunakan dalam

analisis sentimen untuk mengukur efektivitasnya. Hasil penelitian diharapkan dapat memberikan masukan berharga bagi pengembang dalam meningkatkan layanan dan kepuasan pengguna melalui analisis sentimen yang terstruktur.

Metode

Penelitian ini memanfaatkan algoritma *Random Forest* yang diimplementasikan menggunakan perangkat lunak Python. *Random Forest* merupakan pengembangan dari metode *Decision Tree*, yang menggabungkan sejumlah pohon keputusan (*Decision Tree*) dalam proses prediksinya [12]. Setiap pohon keputusan dilatih menggunakan sampel data yang berbeda, dengan pemilihan atribut yang dilakukan secara acak pada masing-masing pohon [13]. Algoritma ini memiliki beberapa keunggulan, antara lain peningkatan akurasi prediksi meskipun terdapat data yang hilang, ketahanan terhadap data outlier, serta efisiensi dalam penyimpanan [14]. Selain itu, *Random Forest* juga memiliki mekanisme seleksi fitur yang memungkinkan pemilihan fitur paling relevan, sehingga dapat meningkatkan performa model klasifikasi. Berkat kemampuan ini, *Random Forest* cocok digunakan untuk menangani big data dengan parameter yang kompleks [15]. Representasi algoritma *Random Forest* ditunjukkan pada Gambar 1 [11].



[11] Gambar 1. Algoritma Random Forest.

Struktur metode ini terdiri dari tiga jenis simpul: *root node, internal node,* dan *leaf node*. *Root node* adalah simpul utama di bagian atas pohon keputusan yang menjadi titik awal proses. Internal node berfungsi sebagai titik percabangan dengan satu input dan minimal dua output untuk membagi data ke jalur selanjutnya. Sementara itu, *leaf node* atau terminal node merupakan simpul akhir yang hanya memiliki input tanpa output. Penentuan nilai output pada algoritma *Random Forest* dilakukan melalui perhitungan tertentu sebagaimana dijelaskan pada Persamaan 1 [16].

$$F(x) = \frac{1}{J}\sum_{j=1}^{J} h_j(x) \quad (1)$$

Keterangan:

$F(x)$: *Output* dari *Random Forest*
J : Jumlah Pohon *Ensemble*
$h_j(x)$: *Output* dari pohon ke-(j)

Confusion matrix merupakan tabel yang digunakan dalam bidang *machine learning* untuk menilai performa suatu model klasifikasi [17], Melalui tabel ini, kita dapat mengevaluasi berbagai metrik penting seperti akurasi, presisi, *recall*, dan *F1-score* dari algoritma yang digunakan [18]. Akurasi sendiri didefinisikan sebagai perbandingan antara jumlah prediksi yang tepat (baik untuk kelas positif maupun negatif) terhadap total keseluruhan data. Nilai akurasi tersebut dapat dihitung menggunakan Persamaan 2.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2)$$

Presisi (*precision*) adalah rasio yang menunjukkan proporsi prediksi positif yang benar, yaitu perbandingan antara jumlah *true positive* (TP) dengan total prediksi positif yang terdiri dari *true positive* (TP) dan *false positive* (FP). Nilai presisi ini dapat dihitung menggunakan Persamaan 3.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

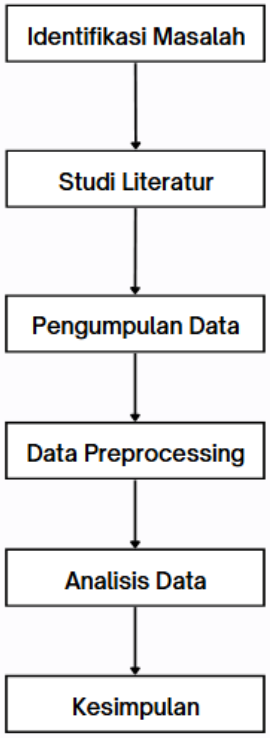
Recall merupakan rasio yang menggambarkan seberapa banyak data positif yang berhasil diklasifikasikan dengan benar, yaitu perbandingan antara jumlah *true positive* (TP) dengan total data aktual yang termasuk dalam kategori positif. Nilai *recall* ini dapat dihitung menggunakan Persamaan 4.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

F1-score merupakan rata-rata harmonis dari *precision* dan *recall*, yang digunakan untuk menyeimbangkan keduanya dalam satu metrik evaluasi. Perhitungan *F1-score* biasanya dilakukan menggunakan rumus yang ditunjukkan pada Persamaan 5.

$$F1\ Score = 2 \times \frac{precision \times recall}{precision+recall} \quad (5)$$

Berdasarkan persamaan tersebut, TP (*True positive*) merujuk pada prediksi yang tepat terhadap data yang memang termasuk dalam kategori positif, sedangkan TN (*True negative*) adalah prediksi yang benar terhadap data yang sebenarnya negatif. FP (*False positive*) merupakan prediksi yang menyatakan data sebagai positif padahal kenyataannya negatif [19] sedangkan FN (*False negative*) adalah prediksi yang mengklasifikasikan data sebagai negatif padahal sebenarnya termasuk dalam kategori positif [20]. Tahapan-tahapan dalam penelitian ini digambarkan pada Gambar 2.



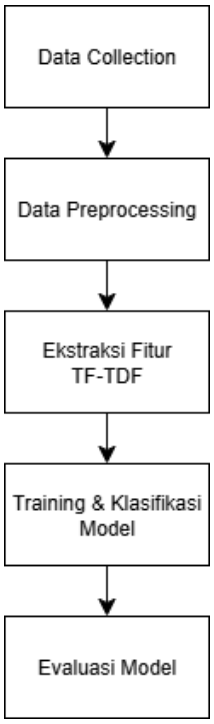
Gambar 2. Tahapan Penelitian

Proses perancangan dalam Analisis Sentimen Review Aplikasi di Google Play Store menggunakan metode *Random Forest* melibatkan beberapa langkah utama. Pertama, dilakukan Identifikasi Masalah untuk menentukan fokus penelitian terkait ulasan pengguna terhadap aplikasi. Selanjutnya, studi literatur dilakukan untuk memahami konsep analisis sentimen, algoritma *Random Forest*, serta tinjauan penelitian terdahulu yang relevan. Tahap berikutnya adalah pengumpulan data dengan mengambil review pengguna dari Google Play Store sebagai dataset yang akan dianalisis. Data tersebut kemudian diproses melalui tahap analisis data yang mencakup praproses, ekstraksi fitur, dan penerapan model *Random Forest* untuk mengklasifikasikan sentimen ulasan. Akhirnya, pada tahap kesimpulan, hasil analisis dievaluasi dan diinterpretasikan guna memahami pola sentimen pengguna terhadap aplikasi yang diteliti.

Perancangan

Dalam tahap perancangan, penelitian ini akan dilaksanakan melalui lima langkah, yaitu pengumpulan data, praproses data, ekstraksi fitur menggunakan TF-IDF, pelatihan dan klasifikasi model, serta tahap terakhir

yaitu evaluasi model. Tahapan-tahapan tersebut digambarkan pada Gambar 3.



Gambar 3. Perancangan Penelitian

Perancangan ini diawali dengan pengumpulan berupa ulasan berjumlah 1000 data pengguna dari Google Play Store yang akan dijadikan dataset untuk analisis. Selanjutnya, dilakukan tahap praproses data yang meliputi pembersihan teks, penghapusan karakter khusus dan *stopwords*, serta normalisasi kata agar data siap digunakan dalam analisis. Setelah itu, fitur diekstraksi menggunakan metode TF-IDF untuk mengonversi teks ulasan menjadi representasi numerik yang dapat dipahami oleh algoritma *machine learning*.

Tahap berikutnya adalah pelatihan dan klasifikasi model, di mana algoritma *Random Forest* digunakan untuk membangun model klasifikasi sentimen berdasarkan data yang telah diproses. Setelah model selesai dibuat, dilakukan evaluasi performa menggunakan metrik seperti akurasi, *precision*, *recall*, dan *f1-score*. Pada akhirnya, perancangan ini mencapai tahap kesimpulan dengan menganalisis hasil yang diperoleh serta menginterpretasikan pola sentimen pengguna terhadap aplikasi di Google Play Store.

Pemodelan

A. *Data Collection*

Dataset yang terlihat pada gambar 4 kemungkinan besar berasal dari ulasan pengguna aplikasi yang dikumpulkan melalui platform seperti Google Play Store atau App Store. Data tersebut berisi tanggapan pengguna mengenai aplikasi yang mereka pakai. Ulasan ini biasanya mencakup opini, pengalaman, maupun keluhan yang disampaikan setelah pengguna menggunakan aplikasi tersebut. Dalam konteks analisis sentimen, data ini dimanfaatkan untuk menentukan apakah ulasan tersebut mengandung sentimen positif, negatif, atau netral terhadap aplikasi yang dibahas.

```
# Menampilkan data sebelum diproses
print("Data Sebelum Diproses:")
print(df.head())

# --- FUNGSI KLASIFIKASI KATEGORI (TETAP SAMA) ---
def klasifikasi_review(review_text):
    if not isinstance(review_text, str): return 'Opini'
    review_lower = review_text.lower()
    if any(kata in review_lower for kata in ['lambat', 'gagal', 'error', 'bug', 'force close', 'crash', 'tidak bisa']):
        return 'Pengalaman'
    elif any(kata in review_lower for kata in ['buruk', 'jelek', 'bingung', 'membingungkan', 'kecewa']):
        return 'Keluhan'
    else: return 'Opini'

df['Kategori'] = df['Review'].apply(klasifikasi_review)
print("\nData Setelah Dipisahkan Berdasarkan Kategori:")
print(df[['Review', 'Sentiment', 'Kategori']].head())
```

```
# Fungsi untuk memisahkan kata relevan dan tidak relevan dalam satu langkah
def pisahkan_kata(teks):
    kata_semua = teks.split()
    kata_relevan = [kata for kata in kata_semua if kata not in stop_words]
    kata_tidak_relevan = [kata for kata in kata_semua if kata in stop_words]
    return ' '.join(kata_relevan), ' '.join(kata_tidak_relevan)

# Terapkan fungsi untuk membuat dua kolom baru secara efisien
df[['kata_relevan', 'kata_tidak_relevan']] = df['cleaned_review_full'].apply(
    lambda x: pd.Series(pisahkan_kata(x))
)
```

```
Data Sebelum Diproses:
                                              Review
0           Sangat bagus, sangat bermanfaat!
1  Mungkin bisa lebih baik, tapi tidak jelek.
2                  Aplikasi cukup biasa saja.
3      Sangat buruk, antarmuka tidak menarik.
4    Aplikasi sangat lambat dan sering gagal.
```

```
Data Setelah Dipisahkan Berdasarkan Kategori:
                                     Review  Sentiment    Kategori
0           Sangat bagus, sangat bermanfaat!          1       Opini
1  Mungkin bisa lebih baik, tapi tidak jelek.         2     Keluhan
2                  Aplikasi cukup biasa saja.         2       Opini
3      Sangat buruk, antarmuka tidak menarik.         0     Keluhan
4    Aplikasi sangat lambat dan sering gagal.         0  Pengalaman
```

Gambar 4. Hasil *Data Collection*

Data pada Gambar 4, berguna untuk memahami persepsi pengguna terhadap aplikasi dan dapat menjadi acuan bagi pengembang dalam melakukan perbaikan atau pembaruan berdasarkan masukan yang diterima. Dataset ini biasanya diperoleh melalui teknik *web scraping* atau melalui API dari platform yang menyediakan akses ke data ulasan pengguna. Pengumpulan data tersebut memiliki berbagai kegunaan, antara lain untuk penelitian, analisis tren, atau sebagai bahan pelatihan model *machine learning* yang mampu mengklasifikasikan sentimen ulasan secara otomatis.

B. *Data Pre-processing*

Hasil *pre-processing* pada dataset ulasan menunjukkan perubahan signifikan dari data mentah yang awalnya berisi ulasan pengguna lengkap dengan tanda baca dan kata-kata umum (*stopwords*) yang kurang relevan untuk analisis sentimen. Setelah menjalani proses seperti konversi seluruh teks ke huruf kecil, penghilangan tanda baca, serta pembersihan karakter khusus, teks menjadi lebih rapi dan fokus pada kata-kata yang mengandung informasi penting untuk analisis lebih lanjut seperti contoh pada Gambar 5.

```
Data Setelah Diproses:
                                     Review  \
0           Sangat bagus, sangat bermanfaat!
1  Mungkin bisa lebih baik, tapi tidak jelek.
2                  Aplikasi cukup biasa saja.
3      Sangat buruk, antarmuka tidak menarik.
4    Aplikasi sangat lambat dan sering gagal.

                          cleaned_review
0           sangat bagus  sangat bermanfaat
1  mungkin bisa lebih baik  tapi tidak jelek
2                aplikasi cukup biasa saja
3     sangat buruk  antarmuka tidak menarik
4   aplikasi sangat lambat dan sering gagal

Data Setelah Penghapusan Stopwords:
                                     Review          cleaned_review
0           Sangat bagus, sangat bermanfaat!        bagus bermanfaat
1  Mungkin bisa lebih baik, tapi tidak jelek.                   jelek
2                  Aplikasi cukup biasa saja.                aplikasi
3      Sangat buruk, antarmuka tidak menarik.  buruk antarmuka menarik
4    Aplikasi sangat lambat dan sering gagal.    aplikasi lambat gagal
```

Gambar 5. Hasil *Pre-processing Data*

Tabel 1. Data setelah diproses dan setelah penghapusan *stopwords*

No.	Review	Cleaned Review
1	Sangat bagus, sangat bermanfaat!	Bagus bermanfaat
2	Mungkin bisa lebih baik, tapi tidak jelek.	Jelek
3	Aplikasi cukup biasa aja.	Aplikasi
4	Sangat buruk, antar muka tidak menarik.	Buruk antar muka menarik
5	Aplikasi sangat lambat dan sering gagal.	Aplikasi gagal lambat

Tabel 1, sebagai contoh kalimat seperti "Sangat bagus, sangat bermanfaat!" diubah menjadi "sangat bagus sangat bermanfaat" melalui proses pembersihan teks. Tujuan dari langkah ini adalah menyederhanakan teks agar lebih siap untuk analisis atau pelatihan model. Setelah itu, kata-kata umum yang sering muncul namun kurang memberikan kontribusi signifikan dalam analisis sentimen, seperti "dan", "untuk", dan "adalah", dihapus melalui proses penghilangan *stopwords*. Dengan demikian, teks menjadi lebih bersih dan lebih fokus pada kata-kata yang penting, seperti "bagus" dan "bermanfaat". Hasil akhir berupa teks yang lebih singkat untuk keperluan analisis sentimen seperti contoh pada Tabel 2 dan 3.

Tabel 2. Kata Relevan

No.	Kata Relevan (positif)	Kata Relevan (negatif)	Kata Relevan (netral)
1	Bagus	Jelek	Aplikasi
2	Bermanfaat	Buruk	Fitur
3	Cepat	Lambat	Antarmuka
4	Mudah	Gagal	Update
5	Nyaman	Error	fungsi
6	Stabil	Crash	Desain
7	Memuaskan	Bug	Aplikasi
8	Menarik	Tidak menarik	

Tabel 3. Kata Tidak Relevan

No.	Kata Sambung	Kata Keterangan Umum	Kata Ganti	Kata Bantu
1	Dan	Sangat	Saya	Adalah
2	Atau	Sekali	Kami	Yang
3	Tetapi	Cukup	Dia	Telah
4	Namun	Lebih	Mereka	Sudah
5	Karena		Ini	Akan
6	Sehingga		Itu	

C. *Ekstraksi Fitur*

Hasil dari proses ekstraksi fitur dengan metode TF-IDF mengungkapkan kata-kata yang memiliki peran penting atau signifikan dalam kumpulan data ulasan yang telah dianalisis. Metode TF-IDF bekerja dengan mengukur tingkat kepentingan sebuah kata berdasarkan frekuensi kemunculannya dalam dokumen tertentu serta seberapa jarang kata tersebut muncul di seluruh dataset. Beberapa kata yang termasuk dalam daftar penting tersebut antara lain "antarmuka", "aplikasi", "bagus", "bermanfaat", "bug", "buruk", "crash", "diperbaiki", "error", dan "fitur", yang menjadi fokus utama dalam analisis sentimen ulasan yang dapat dilihat pada gambar 6.

```
Fitur Teratas setelah Ekstraksi Fitur (TF-IDF):
['antarmuka' 'aplikasi' 'bagus' 'bermanfaat' 'bug' 'buruk' 'crash'
 'diperbaiki' 'error' 'fitur']
```

Gambar 6. Hasil *Ekstraksi Fitur*

Penggunaan kata seperti "antarmuka" dan "aplikasi" mengindikasikan bahwa pengguna lebih

memperhatikan elemen-elemen penting dari aplikasi, seperti desain visual dan fungsi yang ditawarkan. Sementara itu, kata-kata seperti "bagus" dan "bermanfaat" merefleksikan respons yang bersifat positif. Sebaliknya, istilah seperti "bug", "buruk", "crash", dan "error" menandakan adanya keluhan atau penilaian negatif terhadap performa aplikasi. Selain itu, kemunculan kata "fitur" menunjukkan bahwa banyak ulasan membahas berbagai fitur yang tersedia dalam aplikasi tersebut.

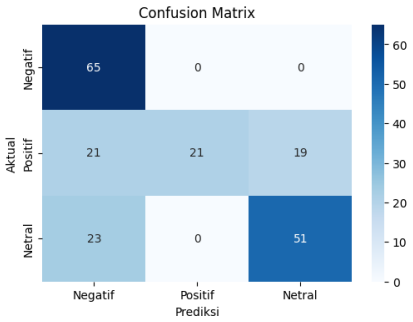
D. *Training dan Klasifikasi Model*

Hasil pada tahap ini memperlihatkan bahwa model memiliki performa yang cukup baik, meskipun masih terdapat beberapa aspek yang perlu ditingkatkan. Algoritma yang digunakan adalah *RandomForestClassifier*, yang sebelumnya telah dilatih menggunakan dataset hasil transformasi TF-IDF, serta diseimbangkan dengan metode SMOTE guna menangani ketidakseimbangan distribusi kelas. Berdasarkan classification report yang dihasilkan, model mencapai tingkat akurasi keseluruhan sebesar 68.5%, yang menunjukkan kemampuan cukup baik dalam memprediksi sentimen, meskipun masih terdapat peluang untuk meningkatkan akurasi, terutama pada kategori tertentu.

Nilai *precision* dan *recall* mengindikasikan bahwa model lebih andal dalam mengidentifikasi sentimen negatif dibandingkan kategori netral maupun positif. Untuk sentimen negatif, *precision* sebesar 0.60 dan *recall* 0.75 mencerminkan kinerja yang cukup baik dalam mendeteksi ulasan dengan nada negatif. Di sisi lain, model mengalami kesulitan dalam membedakan antara sentimen netral dan positif. Pada sentimen netral, baik *precision* maupun *recall* hanya mencapai 0.51, sedangkan pada kategori positif, *precision* berada di angka 0.73, namun *recall*-nya hanya 0.66. Hal ini menunjukkan bahwa model seringkali keliru dalam mengklasifikasikan ulasan netral dan positif, yang kemungkinan besar disebabkan oleh kemiripan kosakata atau ekspresi yang digunakan dalam kedua jenis ulasan tersebut.

E. *Evaluasi Model*

Pada tahap ini, diperoleh akurasi total model sebesar 68.5%. Model menunjukkan kinerja yang cukup baik dalam mendeteksi sentimen negatif, dengan nilai *precision* sebesar 0.60 dan *recall* 0.75, yang mengindikasikan bahwa model lebih efektif dalam mengidentifikasi sentimen negatif dibanding kategori lainnya. Sementara itu, untuk sentimen netral, meskipun *precision* mencapai angka sempurna (1.00), nilai *recall* yang hanya 0.51 menunjukkan bahwa model hanya mampu mengenali sebagian kecil dari data netral yang ada. Artinya, prediksi netral memang sangat tepat, namun cakupannya masih terbatas. Adapun pada sentimen positif, model mencatat *precision* sebesar 0.73 dan *recall* 0.66, yang menandakan performa yang cukup baik, meskipun masih terdapat tantangan dalam membedakan sentimen positif dari negatif.



Gambar 7. *Confusion matrix*

Mengacu pada *confusion matrix* yang disajikan pada gambar 7, model menunjukkan tingkat akurasi yang tinggi dalam mengklasifikasikan sentimen negatif, dengan jumlah prediksi benar sebanyak 65. Namun demikian, terdapat kekeliruan dalam membedakan antara sentimen netral dan positif, yang terlihat dari adanya kesalahan klasifikasi pada kedua kategori tersebut. Secara umum, performa model cukup baik, namun masih diperlukan peningkatan khususnya dalam mendeteksi sentimen netral, terutama untuk meminimalkan kekeliruan antara kategori netral dan positif. Berikut merupakan perhitungan manual yang diambil dari *confusion matrix* pada Table 4.

Table 4. Perhitungan Manual

	Pred: Negatif	Pred: Netral	Pred: Positif	Total per Kelas Asli
Asli: Negatif	65	5	10	80
Asli: Netral	8	26	17	51
Asli: Positif	6	8	46	60
Total Prediksi	79	39	73	200

Akurasi = $\frac{TP+TN+TNetral}{Total} = \frac{65+26+46}{65+5+10+8+26+17+6+8+46} = \frac{137}{200} = 0.685 = 68.5\%$

Negatif:

Precision (negatif) = $\frac{TP}{TP+FP} = \frac{65}{65+14} = \frac{65}{79} = 0.823$

Recall (negatif) = $\frac{TP}{TP+FN} = \frac{65}{65+15}\ \frac{65}{80} = 0.8125$

F1-score (negatif) = F1 = 2 x $\frac{Precision\ x\ Recall}{Precision+Recall} = 2 \times \frac{0.823\ x\ 0.8125}{0.823\ x\ 0.8125} = 0.817$

Netral:

Precision (Netral) = $\frac{TP}{TP+FP} = \frac{26}{26+13} = \frac{26}{39} = 0.666$

Recall (Netral) = $\frac{TP}{TP+FN} = \frac{26}{26+25}\ \frac{26}{51} = 0.509$

F1-score (negatif) = F1 = 2 x $\frac{Precision\ x\ Recall}{Precision+Recall} = 2 \times \frac{0.666\ x\ 0.509}{0.666\ x\ 0.509} = 0.577$

Positif:

Precision (Netral) = $\frac{TP}{TP+FP} = \frac{46}{46+27} = \frac{46}{73} = 0.630$

Recall (Netral) = $\frac{TP}{TP+FN} = \frac{46}{46+14}\ \frac{46}{60} = 0.767$

F1-score (negatif) = F1 = 2 x $\frac{Precision\ x\ Recall}{Precision+Recall} = 2 \times \frac{0.630\ x\ 0.767}{0.630\ x\ 0.767} = 0.692$

Kesimpulan

Berdasarkan hasil analisis dan evaluasi model, perancangan ini menyimpulkan bahwa algoritma *Random Forest* dapat secara efektif mengklasifikasikan sentimen dalam ulasan aplikasi di Google Play Store, dengan tingkat akurasi keseluruhan yang dicapai sebesar 68.5%. Model menunjukkan performa paling kuat dalam mengidentifikasi ulasan dengan sentimen negatif, yang dibuktikan dengan nilai presisi 82.3% dan *recall* 81.25% berdasarkan perhitungan manual. Meskipun demikian, model menghadapi tantangan dalam membedakan secara akurat antara sentimen netral dan positif. Untuk kategori netral, model mencapai presisi sebesar 66.6% namun dengan *recall* yang lebih rendah yaitu 50.9%, mengindikasikan bahwa separuh dari ulasan netral tidak teridentifikasi dengan benar. Sementara itu, kategori positif mencatatkan *recall* yang cukup baik sebesar 76.7%, namun dengan presisi 63.0%, yang menunjukkan adanya ulasan dari kelas lain yang keliru diklasifikasikan sebagai positif. Secara keseluruhan, perancangan ini menegaskan bahwa *Random Forest* adalah metode yang mumpuni untuk analisis sentimen pada ulasan aplikasi. Namun, ada peluang untuk peningkatan di masa depan, terutama dalam menyempurnakan kemampuan model untuk membedakan kelas sentimen yang memiliki kemiripan karakteristik, seperti netral dan positif, guna mencapai akurasi yang lebih tinggi.

Daftar Pustaka

[1] N. P. Husain, S. Sukirman, and S. SAJIAH, "Analisis Sentimen Ulasan Pengguna Tiktok pada Google Play Store Berbasis TF-IDF dan *Support Vector Machine*," *J. Syst. Comput. Eng.*, vol. 5, no. 1, pp. 91–102, 2024, doi: 10.61628/jsce.v5i1.1105.

[2] F. A. D. P. Febrianti, F. Hamami, and R. Y. Fa'rifah, "Aspect-Based Sentiment Analysis Terhadap Ulasan Aplikasi Flip Menggunakan Pembobotan *Term Frequency-Inverse Document Frequency* (Tf-Idf) Dengan Metode Klasifikasi *K-Nearest Neighbors* (K-Nn)," *J. Indones. Manaj. Inform. dan*

Komun., vol. 4, no. 3, pp. 1858–1873, 2023, doi: 10.35870/jimik.v4i3.429.

[3] D. E. Sondakh, S.Kom, M.T, Ph.D, S. W. Taju, M. G. Tene, and A. E. T. Pangaila, "Sistem Analisis Sentimen Ulasan Aplikasi Belanja Online Menggunakan Metode *Ensemble learning*," *CogITo Smart J.*, vol. 9, no. 2, pp. 280–291, 2023, doi: 10.31154/cogito.v9i2.525.280-291.

[4] V. No, H. Sutrisno, N. Anisa, and S. Winarsih, "Klasifikasi Kategori Produk untuk Manajemen Keuangan Remaja menggunakan Algoritma *Long Short-Term Memory*," *Edumatic J. Pendidik. Inform.*, vol. 8, no. 2, pp. 685–693, 2024, doi: 10.29408/edumatic.v8i2.27959.

[5] S. P. Azzahra, A. Azzahra, Y. A. Apriyanto, and A. Wijaya, "Analisis Ulasan Produk Amazon Menggunakan *Random Forest* Sentimen dan Probabilistic Retrieval Model," *J. Informatics busines*, vol. 02, no. 04, pp. 519–528, 2025.

[6] A. T. Hardianti, A. R. Manga, and H. Darwis, "Penerapan Metode *Naïve Bayes* pada Klasifikasi Judul Jurnal," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 97–101, 2018.

[7] N. Alfriyanto, B. C. Purnama, and F. K. Hasanah, "Analisis Emosi Terhadap Komentar Video Youtube ' Penyebab Kegagalan Adopsi Sistem Pendidikan Finlandia di Indonesia ' Menggunakan Metode *Random Forest*," *Conf. Electr. Eng. Informatics, Ind. Technol. Creat. Media*, pp. 812–827, 2024.

[8] D. Hamidah, "Aplikasi Evaluasi Pembelajaran Mahasiswa Jurusan Teknik Industri Universitas Sultan Ageng Tirtayasa Berbasis *Machine learning*." Fakultas Teknik Universitas Sultan Ageng Tirtayasa, 2025.

[9] L. Dan and B. Menggunakan, "Analisis sentimen aplikasi shopee, tokopedia, lazada dan blibli menggunakan leksikon dan *Random Forest*," *JITET J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024.

[10] I. Afdhal, R. Kurniawan, I. Iskandar, R. Salambue, E. Budianita, and F. Syafria, "Penerapan Algoritma *Random Forest* Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 1, pp. 122–130, 2022, doi: 10.47065/bits.v6i2.5368.

[11] R. R. S. Putri Kumala Sari, "Komparasi Algoritma *Support Vector Machine* Dan Random," *J. Mnemon.*, vol. 7, no. 1, pp. 31–39, 2024.

[12] Amaliah Faradibah, Dewi Widyawati, A Ulfah Tenripada Syahar, and Sitti Rahmah Jabir, "Comparison Analysis of *Random Forest* Classifier, *Support Vector Machine*, and Artificial Neural Network Performance in Multiclass Brain Tumor Classification," *Indones. J. Data Sci.*, vol. 4, no. 2, pp. 54–63, Jul. 2023, doi: 10.56705/ijodas.v4i2.73.

[13] N. Husin, "Komparasi Algoritma *Random Forest*, *Naïve Bayes*, dan Bert Untuk Multi-Class Classification Pada Artikel Cable News Network (CNN)," *J. Esensi Infokom J. Esensi Sist. Inf. dan Sist. Komput.*, vol. 7, no. 1, pp. 75–84, 2023, doi: 10.55886/infokom.v7i1.608.

[14] A. W. M. Gaffar, A. L. Isthi'Anah, P. Purnawansyah, A. M. Halis, S. Mujaddid, and A. U. T. Syahar, "Random Search Optimization of Hyperparameter in *Random Forest* Algorithm for Stunting Prediction," in *2024 7th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, IEEE, 2024, pp. 971–976.

[15] O. Manullang, C. Prianto, and N. H. Harani, "Analisis Sentimen Untuk Memprediksi Hasil Calon Pemilu Presiden Menggunakan Lexicon Based Dan *Random Forest*," *J. Ilm. Inform.*, vol. 11, no. 02, pp. 159–169, 2023, doi: 10.33884/jif.v11i02.7987.

[16] B. Prasojo and E. Haryatmi, "Analisa Prediksi Kelayakan Pemberian Kredit Pinjaman dengan Metode *Random Forest*," *J. Nas. Teknol. dan Sist. Inf.*, vol. 7, no. 2, pp. 79–89, 2021, doi: 10.25077/teknosi.v7i2.2021.79-89.

[17] N. A'ayunnisa, Y. Salim, and H. Azis, "Analisis performa metode *Gaussian Naïve Bayes* untuk klasifikasi citra tulisan tangan karakter arab," *Indones. J. Data Sci.*, vol. 3, no. 3, pp. 115–121, 2022.

[18] N. Made *et al.*, "Analisis Sentimen Berbahasa Inggris Dengan Metode Lstm Studi Kasus Berita Online Pariwisata Bali English Sentiment Analysis Using The Lstm Method Case Study Of Bali Tourism Online News," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, pp. 1325–1334, 2024, doi: 10.25126/jtiik.2024118792.

[19] E. Miana, A. Ernamia, and A. Herliana, "Analisis Sentimen Kuliah Daring Dengan Algoritma *Naïve Bayes*, K-Nn Dan *Decision Tree*," *J. Responsif*, vol. 4, no. 1, pp. 70–80, 2022, doi: 10.35870/jtik.v6i1.368.

[20] Y. Ansori and K. F. H. Holle, "Perbandingan Metode *Machine learning* dalam Analisis Sentimen Twitter," *J. Sist. dan Teknol. Inf.*, vol. 10, no. 4, p. 429, 2022, doi: 10.26418/justin.v10i4.51784.