

Klasifikasi Ulasan Pengguna Aplikasi Magic Chess: Go Go Pada Play Store Menggunakan Metode *Random Forest*

Muhamad Ikbal^{1,*}, Yuni Wibawanti², Nurfidah Dwitiyanti³

^{1,2,3} Fakultas Teknik dan Ilmu Komputer, Teknik Informatika, Universitas Indraprasta PGRI, Jakarta, Indonesia
Email: ^{1,*}mikbal91@gmail.com, ²yuniwib206@gmail.com, ³nurfidah.pulungan@gmail.com

^{*)} Email Penulis Utama

Abstrak— Pertumbuhan pesat aplikasi *mobile* telah mendorong peningkatan signifikan dalam interaksi pengguna melalui *platform* digital, salah satunya adalah Google Play Store. Ulasan pengguna pada platform ini menjadi sumber informasi penting yang dapat dimanfaatkan untuk memahami persepsi publik terhadap suatu aplikasi. Salah satu aplikasi yang populer di kalangan pengguna adalah Magic Chess: Go Go, permainan strategi otomatis yang merupakan bagian dari ekosistem Mobile Legends. Meskipun terdapat ribuan ulasan yang tersedia, karakteristik teks bebas dan jumlahnya yang besar menyulitkan proses analisis secara manual. Oleh karena itu, diperlukan metode otomatis untuk mengklasifikasikan ulasan tersebut ke dalam kategori sentimen. Penelitian ini bertujuan untuk menganalisis dan mengklasifikasikan ulasan pengguna aplikasi Magic Chess: Go Go yang diambil dari Google Play Store menggunakan algoritma *Random Forest*. Proses penelitian mencakup beberapa tahap, yaitu pengumpulan data ulasan melalui pustaka *google-play-scraper*, *data preprocessing* (*cleansing*, *case folding*, *tokenization*, *stopword removal*, dan *stemming*), pelabelan data menggunakan kamus opini, serta ekstraksi fitur menggunakan metode TF-IDF. Data yang digunakan dibatasi pada ulasan berbahasa Indonesia selama bulan Mei 2025. Hasil dari penelitian menunjukkan bahwa algoritma *Random Forest* mampu mengklasifikasikan sentimen dengan akurasi sebesar 96,33%, di mana ulasan positif mendominasi sebanyak 63,7% dan ulasan negatif sebesar 36,3%. Hal ini menunjukkan bahwa secara umum, pengguna memberikan tanggapan yang cukup baik terhadap aplikasi tersebut. Temuan ini membuktikan bahwa pendekatan klasifikasi berbasis *Random Forest* dapat digunakan sebagai alat bantu yang efektif untuk mengekstrak opini pengguna dalam skala besar secara otomatis, sehingga dapat memberikan insight yang bernilai bagi pengembang aplikasi dalam pengambilan keputusan strategis ke depan.

Kata Kunci: Magic Chess, Klasifikasi Sentimen, Ulasan Pengguna, *Data Preprocessing*, *Random Forest*, Google Play Store.

Abstract— *The rapid growth of mobile applications has significantly increased user interaction through digital platforms, one of which is the Google Play Store. User reviews on this platform serve as a valuable source of information that can be leveraged to understand public perception of an application. One such popular app among users is Magic Chess: Go Go, an auto-battler strategy game that is part of the Mobile Legends ecosystem. Despite the abundance of available reviews, their unstructured nature and large volume pose challenges for manual analysis. Therefore, an automated method based on artificial intelligence is required to classify these reviews into sentiment categories. This study aims to analyze and classify user reviews of the Magic Chess: Go Go application, retrieved from the Google Play Store, using the Random Forest algorithm. The research process includes several stages: data collection using the google-play-scraper library, data preprocessing (cleansing, case folding, tokenization, stopwords removal, and stemming), data labeling using an opinion lexicon, and feature extraction using the TF-IDF method. The dataset is limited to Indonesian-language reviews collected during May 2025. The results indicate that the Random Forest algorithm achieved a sentiment classification accuracy of 96.33%, with 63.7% of the reviews classified as positive and 36.3% as negative. These findings suggest that, in general, users provided favorable feedback on the application. Moreover, the study demonstrates that the Random Forest-based classification approach is an effective tool for automatically extracting large-scale user opinions, providing valuable insights for developers in making strategic decisions.*

Keywords: Magic Chess, Sentiment Classification, User Reviews, Data Preprocessing, Random Forest, Google Play Store.

1. PENDAHULUAN

Perkembangan industri *game* di Indonesia dalam beberapa tahun terakhir mengalami peningkatan yang sangat pesat dan menjadi salah satu sektor digital yang paling dinamis. Berdasarkan laporan Adikara [1], jumlah pemain *game* di Indonesia diperkirakan akan mencapai 192,1 juta orang pada tahun 2025. Hal ini sejalan dengan data dari Kementerian Komunikasi dan Informatika yang saat ini telah berganti nama menjadi Kementerian Komunikasi dan Digital, yang mencatat bahwa sekitar 35 juta masyarakat Indonesia merupakan gamer aktif daring [2]. Fenomena ini menunjukkan bahwa *game* telah menjadi bagian dari gaya hidup masyarakat modern, terutama bagi generasi muda yang semakin terbiasa berinteraksi melalui platform digital. Salah satu kategori permainan yang digemari oleh para pemain adalah *auto battler*, dengan *Magic Chess: Go Go* sebagai salah satu judul populer yang dikembangkan oleh Moonton [3]. Game ini berhasil menarik perhatian jutaan pemain karena menggabungkan strategi dan hiburan dalam satu platform interaktif. Dengan semakin banyaknya pengguna aktif, jumlah ulasan atau *review* yang ditulis di platform seperti Google Play Store juga meningkat secara signifikan. Ulasan-ulasan tersebut berisi beragam opini pengguna mengenai pengalaman bermain, fitur yang disukai, kendala teknis, hingga saran untuk pengembang. Dengan demikian, analisis terhadap tanggapan pengguna menjadi hal yang penting

untuk memahami persepsi dan tingkat kepuasan terhadap aplikasi, sehingga dapat dijadikan bahan evaluasi dan perbaikan kualitas layanan [4].

Analisis terhadap teks ulasan dapat dilakukan menggunakan pendekatan *sentiment analysis* atau analisis sentimen, yang memiliki tujuan untuk mengenali dan mengelompokkan pendapat pengguna ke dalam kategori positif atau negatif. Analisis ini menjadi salah satu metode populer pada ranah *text mining* dan *natural language processing* karena mampu mengekstraksi makna dari teks secara otomatis. Salah satu algoritma yang dikenal efektif dalam analisis sentimen adalah *Random Forest* (RF), yang merupakan bagian dari pendekatan *ensemble learning*. Algoritma ini bekerja dengan memadukan beberapa pohon keputusan (*decision tree*) guna memperoleh hasil prediksi yang lebih tepat dan konsisten, sekaligus mengurangi risiko *overfitting* [5]. Dalam berbagai penelitian, *Random Forest* terbukti mampu memberikan performa yang tinggi dalam tugas klasifikasi. Sebagai contoh, penelitian oleh Aldean et al. [6] menunjukkan bahwa RF dapat digunakan secara efektif untuk mengklasifikasikan opini publik terkait vaksin COVID-19 pada platform Twitter dengan hasil yang signifikan. Penelitian lain oleh Rohman et al. [7] juga membuktikan keberhasilan algoritma ini dalam menganalisis ulasan restoran di Surabaya dengan tingkat akurasi yang memuaskan. Selain itu, Larasati et al. [8] dan Indrayanto et al. [9] mengimplementasikan RF pada analisis sentimen aplikasi Dana dan MyPertamina dan menemukan bahwa algoritma ini memiliki tingkat akurasi yang lebih tinggi daripada algoritma lain seperti *Naïve Bayes* dan *Support Vector Machine*. Berbagai hasil tersebut menegaskan bahwa *Random Forest* tidak hanya kuat dalam mengolah data teks berskala besar, tetapi juga mampu menangani variasi bahasa dan konteks yang kompleks.

Meskipun banyak penelitian terdahulu yang menerapkan algoritma *Random Forest* dalam melakukan analisis sentimen di berbagai domain, penelitian yang secara spesifik berfokus pada ulasan pengguna aplikasi *Magic Chess: Go Go* masih jarang dilakukan. Padahal, aplikasi ini memiliki jutaan pengguna aktif dan potensi data ulasan yang besar untuk dianalisis. Beberapa penelitian terkait aplikasi sejenis seperti *Mobile Legends: Bang Bang* [4] dan *Capcut* [10] menunjukkan bahwa pendekatan *Random Forest* dapat menghasilkan klasifikasi sentimen yang efektif dan konsisten. Dengan demikian, penelitian ini dimaksudkan untuk menutup celah tersebut dengan mengembangkan model klasifikasi sentimen terhadap ulasan pengguna aplikasi *Magic Chess: Go Go* menggunakan algoritma *Random Forest*. Dalam penelitian ini, data ulasan diambil langsung dari platform Google Play Store untuk memastikan relevansi dan keaslian data. Setiap data ulasan kemudian melalui proses *text preprocessing* yang mencakup beberapa tahapan seperti *cleansing* (pembersihan teks dari angka, simbol, dan tanda baca), *case folding* (menyeragamkan format huruf), *tokenization* (membagi teks ke dalam satuan kata), *stopword removal* (menghilangkan kata-kata umum yang tidak memiliki makna signifikan), serta *stemming* (mengembalikan kata ke bentuk dasarnya). Tahapan ini dilakukan untuk memastikan bahwa data siap diproses oleh algoritma pembelajaran mesin dengan struktur yang bersih dan konsisten.

Selanjutnya, fitur teks diekstraksi memanfaatkan metode *Term Frequency–Inverse Document Frequency* (TF-IDF) [11], yang berfungsi untuk menilai tingkat kepentingan suatu kata dalam dokumen dibandingkan dengan keseluruhan kumpulan teks. Setelah fitur diekstraksi, data dilatih menggunakan algoritma *Random Forest* guna membentuk model klasifikasi sentimen. Kemudian dievaluasi menggunakan beberapa metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* yang diperoleh dari hasil *confusion matrix* [5]. Untuk memudahkan interpretasi hasil dan penerapannya dalam konteks praktis, penelitian ini juga mengimplementasikan model dalam bentuk *dashboard* visualisasi berbasis web dengan memakai *framework* Flask dan bahasa pemrograman Python [12]. Dashboard ini menyediakan fitur interaktif seperti unggah data ulasan, proses pelatihan model, hasil klasifikasi, serta visualisasi berupa diagram batang, *wordcloud*, dan tabel kata penting berdasarkan bobot TF-IDF. Melalui dashboard ini, analisis sentimen dapat dilakukan secara lebih mudah, efisien, dan dapat diakses oleh pengguna non-teknis, termasuk pengembang aplikasi dan peneliti di bidang terkait.

Untuk memastikan sistem berjalan dengan baik, penelitian ini juga menggunakan pendekatan pemodelan perangkat lunak berbasis *Unified Modeling Language* (UML), yang mencakup *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram* [13]. Pendekatan ini dipakai untuk memvisualisasikan hubungan antar komponen sistem, alur proses, serta interaksi antara pengguna dan sistem secara terstruktur. Dengan demikian, pengembangan sistem menjadi lebih terarah dan mudah dipahami oleh seluruh tim pengembang. Secara keseluruhan, penelitian ini berkontribusi dalam memberikan pemahaman mendalam tentang persepsi pengguna terhadap aplikasi *Magic Chess: Go Go* sekaligus menunjukkan efektivitas algoritma *Random Forest* dalam klasifikasi sentimen teks berbahasa Indonesia. Hasil penelitian ini diharapkan tidak hanya berguna bagi *software development* untuk memperbaiki tingkat kualitas produk berdasarkan masukan pengguna, tetapi juga menjadi referensi selanjutnya di bidang analisis sentimen serta kecerdasan buatan. Dengan memanfaatkan kekuatan metode *Random Forest* yang telah terbukti handal, penelitian ini menjadi langkah konkret dalam mengintegrasikan teknologi *machine learning* dengan kebutuhan praktis industri game yang terus berkembang di Indonesia.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen terapan dalam bidang *machine learning* dan *text mining*, yang difokuskan pada pengembangan sistem klasifikasi sentimen berbasis teks sekaligus

implementasinya dalam bentuk dashboard visual interaktif. Proses penelitian dirancang melalui enam tahapan utama, yaitu: pengumpulan data, praproses data (*data preprocessing*), pelabelan sentimen, ekstraksi fitur menggunakan TF-IDF, pelatihan model menggunakan algoritma *Random Forest*, serta evaluasi performa klasifikasi [17]. Seluruh tahapan tersebut diintegrasikan ke dalam sistem berbasis web yang dikembangkan menggunakan *framework* Flask dan bahasa pemrograman Python untuk memudahkan pengguna dalam mengakses, melatih, dan memvisualisasikan hasil analisis sentimen. Pendekatan ini mengacu pada kerangka kerja umum penelitian *text mining* dan analisis sentimen yang banyak diterapkan dalam studi klasifikasi teks modern [14].



Gambar 1. Alur Penelitian

2.1 Pengumpulan Data

Langkah awal dalam penelitian ini dimulai dengan proses pengambilan data ulasan pengguna *Magic Chess: Go Go*. Data diambil melalui *Google Play Store* dengan bantuan pustaka *python* bernama *google-play-scraper*. Proses ini mengacu pada metode yang digunakan oleh Larasati et al. [8] dan Indrayanto et al. [9] yang juga meneliti aplikasi berbasis Android. Alat ini memungkinkan peneliti untuk mengakses ulasan publik secara otomatis berdasarkan *package name* aplikasi, yaitu *com.mobilechess.gp*. Agar hasil analisis lebih fokus dan kontekstual, data yang diambil dibatasi pada periode satu bulan, yakni Mei 2025 dengan jumlah data yang didapatkan 6.542, dan hanya mencakup ulasan berbahasa Indonesia.

Ulasan ini dikemas dalam format *dataframe* menggunakan pustaka *pandas* dan diekspor ke format CSV untuk keperluan lanjutan [15]. Setiap entri ulasan terdiri dari kolom teks, tanggal, serta informasi *rating* (nilai bintang). Data mentah ini masih mengandung banyak informasi yang tidak terstruktur, sehingga diperlukan tahapan *preprocessing* sebelum dapat digunakan untuk pelatihan model klasifikasi.

2.2 Data Preprocessing

Data preprocessing merupakan tahap krusial dalam pengolahan teks, karena kualitas data sangat memengaruhi performa. Metode *preprocessing* mengacu pada prosedur yang dijelaskan oleh Findawati dan Rosid [14] secara berurutan sebagai berikut:

- Cleansing*: Menghilangkan karakter yang tidak relevan seperti simbol, *URL*, tanda baca, angka serta *emoji*. Tujuannya adalah untuk menghapus elemen-elemen yang dapat mengganggu dalam konteks klasifikasi sentimen.
- Case Folding*: Proses mengubah seluruh karakter dalam teks menjadi huruf kecil untuk menghindari redundansi antara kata yang sama, misalnya "Bagus" dan "bagus".
- Tokenization*: Memecah teks menjadi potongan kata-kata agar dapat dianalisis lebih lanjut.
- Stopword Removal*: Proses menghapus kata-kata yang sering muncul namun tidak memberikan kontribusi makna signifikan dalam menentukan sentimen, seperti "yang", "dengan", "di" dan sebagainya.
- Stemming*: Proses mengembalikan kata ke bentuk akar atau dasar. Misalnya, kata "bermain" dan "memainkan" akan distem menjadi "main".

Hasil dari tahap *data preprocessing* ini adalah teks bersih dan konsisten yang siap untuk diberikan label dan diekstraksi fiturnya.

2.3 Pelabelan Data

Setelah *data preprocessing* selesai, data perlu diberikan label sentimen. Penelitian ini menggunakan pendekatan berbasis *lexicon*, yaitu kamus opini berbahasa Indonesia. Kamus ini memuat daftar kata dengan muatan sentimen positif maupun negatif. Setiap ulasan dianalisis berdasarkan frekuensi kemunculan kata dari kamus tersebut. Jika ulasan mengandung lebih banyak kata positif, maka akan diberi label positif, dan sebaliknya diberi label negatif. Untuk menjaga kualitas label, ulasan netral atau tidak mengandung kata dari kamus opini dikeluarkan dari dataset pelatihan. Setelah tahap ini, dataset akhir terdiri dari dua kolom utama: *text* dan *label*.

2.4 Ekstraksi Fitur

Tahap selanjutnya adalah mengubah teks ke dalam format berbentuk numerik sehingga dapat diproses oleh algoritma pembelajaran mesin. Teknik yang diaplikasikan dalam proses ini adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). Ini mengukur tingkat kepentingan suatu kata di dalam sebuah dokumen terhadap

seluruh korpus. Dengan kata lain, TF-IDF mengedepankan kata-kata yang unik dan bermakna dalam konteks klasifikasi [16]. TF-IDF menghasilkan *vector* representasi dari setiap ulasan, yang nantinya menjadi input bagi algoritma klasifikasi. Proses ini dikerjakan dengan menggunakan pustaka *sklearn.feature_extraction.text.TfidfVectorizer* memakai parameter default. Jumlah fitur akhir yang dihasilkan bergantung pada kosakata unik dari seluruh dokumen, biasanya mencapai ratusan hingga ribuan dimensi.

2.5 Implementasi *Random Forest*

Setelah data diekstrak menjadi vektor numerik, tahap selanjutnya adalah pelatihan memanfaatkan algoritma *Random Forest*. Pemilihan algoritma ini didasarkan pada alasan bahwa kestabilannya dalam mengelola data yang besar juga kemampuannya terkait menghindari *overfitting*. Data dilatih menggunakan pustaka *sklearn.ensemble.RandomForestClassifier*. Dataset dipisahkan menjadi dua subset, yakni 80% sebagai data pelatihan dan 20% sebagai data pengujian. Parameter default digunakan untuk jumlah *estimators* (100 pohon keputusan), namun dapat disesuaikan untuk eksplorasi lebih lanjut. Selama proses pelatihan, model membangun beberapa pohon keputusan dan memilih mayoritas suara (*majority voting*) untuk menentukan prediksi akhir.

2.6 Evaluasi Klasifikasi

Sesudah tahap pelatihan rampung, *Random Forest* dievaluasi memakai data uji guna mengukur sejauh mana kemampuannya dalam mengklasifikasi sentimen ulasan secara akurat. Evaluasi dilakukan dengan menggunakan beberapa metrik kinerja yang umum digunakan dalam klasifikasi [5], yang disediakan oleh pustaka *sklearn.metrics*. diantaranya:

- a. *Accuracy* (akurasi): Proporsi hasil prediksi yang tepat dari seluruh prediksi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- b. *Precision* (presisi): Proporsi prediksi positif yang benar terhadap seluruh prediksi positif.

$$Precision = \frac{TP}{TP + FP}$$

- c. *Recall*: Kemampuan model dalam menemukan seluruh data aktual positif.

$$Recall = \frac{TP}{TP + FN}$$

- d. *F1-Score*: Harmonik nilai rerata antara *precision* (presisi) dan *recall*.

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- e. *Confusion Matrix*

Tabel 1. *Confusion Matrix*

Keterangan	Prediksi Positif	Prediksi Negatif
Aktual Positif	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
Aktual Negatif	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Keterangan: *TP* (*True Positive*): Ulasan positif yang diklasifikasikan positif

TN (*True Negative*): Ulasan negatif yang diklasifikasikan negatif

FP (*False Positive*): Ulasan negatif yang diklasifikasikan positif

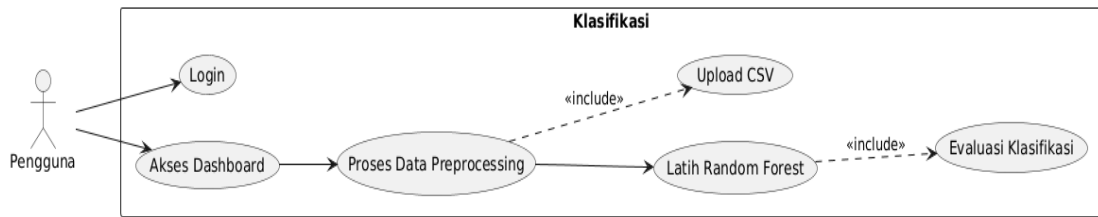
FN (*False Negative*): Ulasan positif yang diklasifikasikan negatif

Dengan menggunakan metrik-metrik tersebut, evaluasi dilakukan secara menyeluruh untuk memastikan bahwa klasifikasi tidak hanya akurat, tetapi juga andal dalam mengidentifikasi ulasan positif maupun negatif. Hasil dari evaluasi ini menjadi dasar dalam menilai kinerja dan efektivitas dari penerapan algoritma *Random Forest* dalam klasifikasi sentimen.

2.7 Perancangan Sistem

- a. *Use Case Diagram*

Diagram ini menggambarkan fungsionalitas utama sistem dari sudut pandang aktor (pengguna). Aktor utama adalah Admin/Pengguna yang akan *login*, akses ke *dashboard*, proses *data preprocessing* dengan *upload CSV* dan latih *Random Forest* yang menghasilkan evaluasi klasifikasi.



Gambar 2. Use Case Diagram

b. Desain Fitur

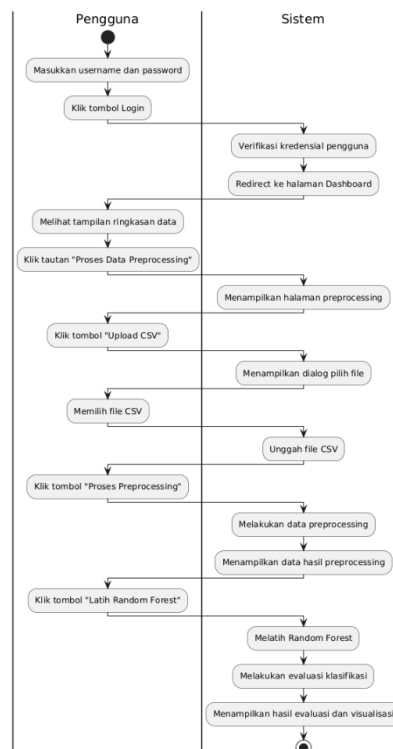
Berikut ini adalah tabel berupa uraian fitur-fitur yang terdapat dalam sistem sesuai dengan *Use Case Diagram*:

Tabel 2. Desain Fitur

No	Fitur	Deskripsi Singkat
1	Login	Fitur autentikasi untuk mengakses sistem. Menjamin keamanan dan pembatasan akses.
2	Akses Dashboard	Halaman utama yang menampilkan ringkasan statistik, visualisasi hasil, dan navigasi.
3	Proses Data Preprocessing	Menjalankan tahapan pembersihan dan normalisasi teks ulasan secara otomatis.
4	Upload File CSV	Memungkinkan pengguna mengunggah file ulasan (.csv) untuk dianalisis.
5	Latih <i>Random Forest</i>	Melakukan pelatihan klasifikasi menggunakan algoritma <i>Random Forest</i> .
6	Evaluasi Klasifikasi	Menampilkan performa klasifikasi melalui metrik evaluasi seperti akurasi, precision, dsb.

c. Activity Diagram

Dalam penelitian ini, *activity diagram* digunakan untuk merepresentasikan alur proses dari berbagai fungsi utama sistem klasifikasi sentimen, yaitu proses *login*, akses *dashboard*, unggah dan *data preprocessing*, hingga pelatihan dan evaluasi *Random Forest*.

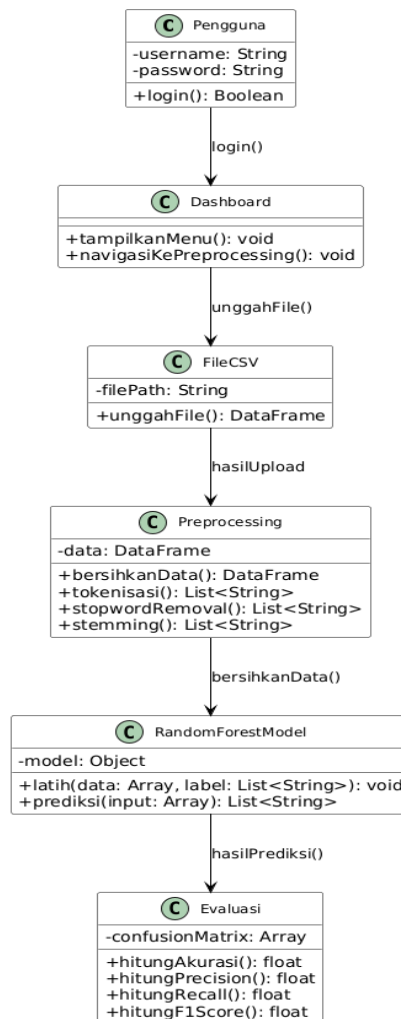


Gambar 3. Activity Diagram

Aktivitas sistem secara keseluruhan dimulai dengan pengguna menyetikkan *username* serta *password* di halaman *login*, lalu mengklik *login button*. Sistem selanjutnya memverifikasi kredensial pengguna dan mengarahkan pengguna ke halaman *dashboard* apabila login berhasil. Di halaman *dashboard*, pengguna dapat melihat ringkasan data dan memilih untuk melanjutkan ke proses *data preprocessing*. Setelah diarahkan ke halaman *preprocessing*, pengguna melakukan unggah *file* CSV yang akan dianalisis. Sistem menampilkan dialog pemilihan file, dan setelah pengguna memilih file, sistem mengunggah file tersebut. Proses dilanjutkan dengan pengguna menekan tombol "Proses *Preprocessing*", yang akan memicu sistem untuk melakukan serangkaian tahap *preprocessing*. Hasil *preprocessing* ditampilkan sebagai umpan balik kepada pengguna. Langkah selanjutnya adalah proses pelatihan dengan algoritma *Random Forest*. Pengguna menekan tombol "Latih *Random Forest*", dan sistem akan menjalankan proses pelatihan, kemudian melakukan evaluasi klasifikasi untuk menghasilkan ukuran kinerja yang terdiri dari akurasi, *precision*, *recall*, dan *f1-score*. Akhirnya, evaluasi dan visualisasi ditampilkan kepada pengguna.

d. *Class Diagram*

Diagram kelas merupakan salah satu jenis diagram statis dalam *Unified Modeling Language* (UML) yang digunakan untuk merepresentasikan struktur antar kelas dalam suatu sistem perangkat lunak.

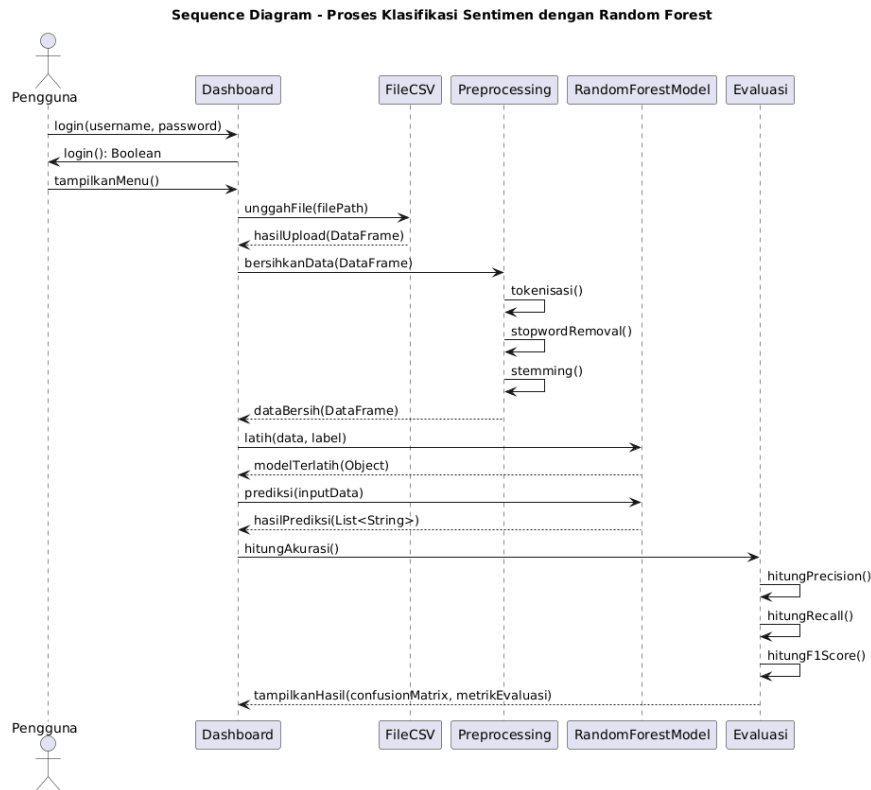


Gambar 4. *Class Diagram*

e. *Sequence Diagram*

Sequence diagram ini menggambarkan alur interaksi antar objek dalam sistem klasifikasi sentimen berbasis *Random Forest*. Proses dimulai ketika Pengguna melakukan *login()* melalui Dashboard untuk mengakses fitur utama sistem. Pengguna kemudian mengunggah dataset melalui kelas FileCSV, yang membaca file CSV dan mengubahnya menjadi *DataFrame* untuk diproses. Data yang berhasil diunggah diteruskan ke kelas Preprocessing, di mana dilakukan tahapan *cleansing*, *tokenisasi*, *stopword removal*, dan *stemming* agar teks siap untuk pelatihan model. Setelah data bersih, kelas RandomForestModel melakukan proses *latih()* untuk membangun model klasifikasi dan *prediksi()* untuk menghasilkan label

sentimen. Hasil prediksi selanjutnya dievaluasi oleh kelas Evaluasi menggunakan metrik seperti *akurasi*, *presisi*, *recall*, dan *F1-score*, serta ditampilkan kembali melalui dashboard. Secara keseluruhan, diagram ini menunjukkan alur logis sistem dari input pengguna hingga evaluasi hasil klasifikasi, mencerminkan keterpaduan antara proses preprocessing, pelatihan model, dan analisis performa sistem.



Gambar 5. Sequence Diagram

4. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan Data

Dalam penelitian ini, data dikumpulkan dari aplikasi *Magic Chess: Go Go* yang tersedia di *Google Play Store* menggunakan pustaka *google-play-scraper*. Proses pengambilan data dilakukan pada bulan Mei 2025, dengan total 6.542 ulasan yang berhasil diperoleh. Ulasan yang terkumpul terdiri atas berbagai ekspresi opini pengguna terkait fitur permainan, performa aplikasi, pembaruan terkini, serta kepuasan mereka terhadap *gameplay* secara umum.

```

from google_play_scraper import reviews, sort
import pandas as pd
from datetime import datetime

all_reviews = []
token = None
done = False

print("📦 Mengambil ulasan dari Google Play...")

while not done:
    result, token = reviews(
        'com.magicchess.gp',
        lang='id',
        country='id',
        sort=sort.NEWEST,
        count=100,
        continuation_token=token
    )

    if not result:
        break # Hentikan jika tidak ada lagi ulasan

    # Filter langsung di sint
    for r in result:
        at = r['at']
        if at.month == 5 and at.year == 2025:
            all_reviews.append(r)
        elif at < datetime(2025, 5, 1): # Jika sudah masuk April ke bawah, selesai
            done = True
            break

print(f"✅ Total ulasan bulan Mei 2025 yang berhasil diambil: {len(all_reviews)}")

# konversi ke dataframe
df = pd.DataFrame(all_reviews)

# Simpan kalau penting
if not df.empty:
    df = df[['username', 'score', 'content', 'at']]
    df.to_csv('ulasan_mei2025_magicchess.csv', index=False, encoding='utf-8')
    print("📁 Data berhasil disimpan ke ulasan_mei2025_magicchess.csv")
else:
    print("❌ Tidak ada ulasan dari bulan Mei 2025 yang ditemukan.")
    
```

Gambar 6. Proses dan Hasil Pengumpulan Data

3.2 Hasil Data Preprocessing

Data yang diperoleh kemudian melalui tahapan *data preprocessing*, yaitu:

a. Case Folding

Pada proses ini, seluruh ulasan yang berjumlah 6.542 ulasan akan dinormalisasi dengan mengubah seluruh huruf besar yang terdapat pada ulasan dikonversi seluruhnya ke dalam format huruf kecil.

content	at	content_caselfold
makin kesini makin kayak tailliiiiii ngawurrrr gak jlasssss..	2025-05-31 15:59:59	makin kesini makin kayak tailliiiiii ngawurrrr gak jlasssss..
game gk jelas masa ngasih hero pelit amat. donpo	2025-05-31 15:58:37	game gk jelas masa ngasih hero pelit amat. donpo
GAME NYA ENAK BUAT MAIN KLO LAGI MOOD	2025-05-31 15:43:14	game nya enak buat main klo lagi mood
memiliki game yang sangat bagus yahhh itu aja	2025-05-31 15:40:57	memiliki game yang sangat bagus yahhh itu aja
game banyak bug aneh developer belajar coding lagi	2025-05-31 15:37:06	game banyak bug aneh developer belajar coding lagi
iv udah masuk ke adven server tapi mana mode practise nya mana mode yg lama!!!!	2025-05-31 15:33:55	udah masuk ke adven server tapi mana mode practise nya mana mode yg lama!!!!
si gini enak kan mcgg sebelum di update sinergi juga pada enak daripada sekarang	2025-05-31 15:30:48	ini enak kan mcgg sebelum di update sinergi juga pada enak daripada sekarang
jarang dapet hero yg gua mau	2025-05-31 15:21:13	jarang dapet hero yg gua mau
inginkan banyak pilihan dan nambah seru juga 🍀 biar bintang5 yang berbicara	2025-05-31 15:10:02	inginkan banyak pilihan dan nambah seru juga 🍀 biar bintang5 yang berbicara
seruuuu	2025-05-31 14:55:31	seruuuu
say TIDAK SUKA game ini. saya akan menghapus game ini karna jelek dan nga adil	2025-05-31 14:51:55	say tidak suka game ini. saya akan menghapus game ini karna jelek dan nga adil
Bagus bgt, dah itu aja	2025-05-31 14:47:27	bagus bgt, dah itu aja
ampes	2025-05-31 14:46:29	ampes
mcel cuma muncul lufisan memuat mohon tunggu. gw udah nunggu 15 menitan loh	2025-05-31 14:45:03	et cuma muncul lufisan memuat mohon tunggu. gw udah nunggu 15 menitan loh
KASI LAWAN BOT	2025-05-31 14:44:59	kasi lawan bot
keren sih	2025-05-31 14:44:36	keren sih
sinergi gajelas semua mendindo sinergi ya lama	2025-05-31 14:39:37	sinergi gajelas semua mendindo sinergi ya lama
Maaf, lebih seru MC yang lama 🤔	2025-05-31 14:37:45	maaf, lebih seru mc yang lama 🤔

Gambar 7. Hasil Proses Case Folding

b. Cleansing

Proses *cleansing* bertujuan untuk menyaring data teks dari elemen-elemen yang tidak diperlukan atau dapat mengganggu proses analisis, seperti angka, simbol, tanda baca, serta emotikon.

content	at	content_clean
game asu... pas main nortuoper seriyar peranti rageseg masean	2025-05-21 19:40:32	game asu pas main nortuoper seriyar peranti rageseg masean
sering hilang jaringan sendiri saat posisi bagus. memang jelek atau karena ceat?	2025-05-21 19:40:01	sering hilang jaringan sendiri saat posisi bagus memang jelek atau karena ceat?
bagus tapi hero nya lebih banyakin dikit	2025-05-21 19:39:56	bagus tapi hero nya lebih banyakin dikit
3 gara gara afk. Game cacat, jangan langsung munculin di play store kalah masih cacat dasar tolot	2025-05-21 19:37:46	juara gara gara afk game cacat jangan langsung munculin di play store kalah masih cacat dasar tolot
seruu parahh	2025-05-21 19:28:00	seruu parahh
saat puas	2025-05-21 19:25:12	saat puas
bagus	2025-05-21 19:22:29	bagus
game tolot	2025-05-21 19:12:55	game tolot
ah gratisnya cuma emot,udah glu emot banyak" buat apa,kasih gratis apa ke selain emot banyakin.	2025-05-21 19:11:53	hadiah gratisnya cuma emot,udah glu emot banyak buat apa kasih gratis apa ke selain emot banyakin
burik	2025-05-21 19:10:52	burik
ke 4... seterusnya zonk sdh... card dan pilihan hero juga sprti sdh di setting sprti itu 🍀🍀🍀🍀	2025-05-21 19:07:34	bisa juara main ke seterusnya zonk sdh card dan pilihan hero juga sprti sdh di setting sprti itu
seruuuuu	2025-05-21 19:07:13	seruuuuu
salah, dulu gw mantan poin reban, lah sekarang apa cik hero susah lose strek pula, ah lah kawan	2025-05-21 19:04:21	yang salah dulu gw mantan poin reban lah sekarang apa cik hero susah lose strek pula ah lah kawan
game paling santuy	2025-05-21 18:57:43	game paling santuy
bagus sekali 🍀	2025-05-21 18:47:28	bagus sekali
lawak sinyal gangguan terus sistem aneh	2025-05-21 18:43:42	lawak sinyal gangguan terus sistem aneh
game nyebel. main di legend, ketemu enemy honorigrory terus	2025-05-21 18:41:46	game nyebel main di legend ketemu enemy honor glory terus
bagus	2025-05-21 18:35:44	bagus
ro nya cuma dpat 1 itu pun kiau menang, pas hero tinggal 1 mau bintang 3 dtang nya miah hero lain	2025-05-21 18:35:39	n udah hero nya cuma dpat ut pun kiau menang pas hero tinggal mau bintang dtang nya miah hero lain

Gambar 8. Hasil Cleansing

c. Tokenizing

Setelah melalui proses cleansing, tahap selanjutnya dalam praproses data adalah *tokenizing*. *Tokenizing* merupakan proses membagi teks atau kalimat yang telah melalui tahap pembersihan menjadi bagian-bagian kecil yang disebut token, biasanya dalam bentuk kata-kata. Tahapan ini sangat penting karena memberikan peluang sistem untuk mengelola dan menganalisis teks secara terpisah berdasarkan kata per kata. Sebagai contoh, kalimat ulasan seperti "game ini sangat bagus dan seru" akan diubah menjadi daftar token: [game, ini, sangat, bagus, dan, seru].

content_clean	tokens
kok aku punya saya gak bisa di pake padahal baru bikin	['kok', 'aku', 'punya', 'saya', 'gak', 'bisa', 'di', 'pake', 'padahal', 'baru', 'bikin']
seru	['seru']
pilih kasih antara player normal dan content creators	['pilih', 'kasih', 'antara', 'player', 'normal', 'dan', 'content', 'creators']
sangat bagus tapi tolong sinergi kangkung ditambah kan	['sangat', 'bagus', 'tapi', 'tolong', 'sinergi', 'kangkung', 'ditambah', 'kan']
bagus lumayan	['baguss', 'lumayan']
setelah update jdI sering lag perlu optimalisasi jaringan	['setelah', 'update', 'jdI', 'sering', 'lag', 'perlu', 'optimalisasi', 'jaringan']
aku download karena bang windah basudara	['aku', 'download', 'karena', 'bang', 'windah', 'basudara']
sangat baik	['sangat', 'baik']
keren sekali	['keren', 'sekali']

Gambar 9. Hasil Tokenizing

d. Stopword Removal

Stopword removal merupakan tahapan untuk menghilangkan kata-kata yang kerap muncul dalam teks tetapi tidak memberikan makna penting dalam proses analisis, seperti “dan”, “yang”, “di”, “ke”, serta kata umum lainnya. Kata-kata ini dikenal sebagai stopwords karena dianggap tidak memiliki makna khusus dalam konteks pemrosesan bahasa alami.

tokens	tokens_nostop
[okeh, 'dah]	[okeh, 'dah]
[kangen, 'magic', 'chess', 'dulu]	[kangen, 'magic', 'chess', 'dulu]
[kembalikan, 'megic', 'chess', 'seperti', 'dulu', 'yang', 'sekarang', 'rumit', 'monton', 'bikin', 'sulit', 'aja]	[kembalikan, 'megic', 'chess', 'dulu', 'sekarang', 'rumit', 'monton', 'bikin', 'sulit', 'aja]
[nyari, 'thero', 'sulit', 'amat', 'yang', 'lain', 'gk', 'roll', 'brutal', 'pada', 'jadi]	[nyari, 'thero', 'sulit', 'gk', 'roll', 'brutal', 'jadi]
[lag, 'bangek', 'main', 'ml', 'main', 'codm', 'aman', 'aja]	[lag, 'bangek', 'main', 'ml', 'main', 'codm', 'aman', 'aja]

Gambar 10. Hasil Stopword Removal

e. Stemming

Tahap *stemming* merupakan proses untuk mengkonversi setiap kata dalam teks menjadi bentuk dasarnya atau bentuk akar katanya. Proses ini bertujuan untuk menyatukan berbagai bentuk kata yang memiliki akar yang sama ke dalam satu bentuk dasar, sehingga membantu menyederhanakan representasi kata dalam analisis teks. Misalnya, kata-kata seperti "bermain", "memainkan", dan "permainan" semuanya akan dikembalikan ke bentuk dasarnya, yaitu "main". Dalam konteks analisis sentimen, hal ini penting agar sistem tidak menganggap kata-kata tersebut sebagai kata yang berbeda, padahal memiliki makna yang serupa.

tokens_nostop	tokens_stemmed
[perbaiki, 'sistem', 'ganti', 'akunnya', 'dong', 'susah', 'ganti', 'akun', 'mlh']	[baik, 'sistem', 'ganti', 'akun', 'dong', 'susah', 'ganti', 'akun', 'mlh']
[kemahalan, 'toto', 'wanwan', 'lukas', 'nerf', 'miya', 'nerf', 'wanwan', 'pokoknya', 'dinerf', 'ampe', 'mmpus']	[kemahalan, 'toto', 'wanwan', 'lukas', 'nerf', 'miya', 'nerf', 'wanwan', 'pokok', 'dinerf', 'ampe', 'mmpus']
[commader, 'nya', 'gak', 'sama', 'dulu', 'kembalikan', 'mob', 'rush', 'gamenya', 'seru', 'gak', 'sama', 'lainnya']	[commader, 'nya', 'gak', 'sama', 'dulu', 'kembali', 'mob', 'rush', 'gamenya', 'seru', 'gak', 'sama', 'lain']
[agus, 'bikin', 'happy', 'semoga', 'mas', 'moonton', 'sehat', 'selalu', 'panjang', 'umurnya', 'rezeki', 'aminin']	[agus, 'bikin', 'happy', 'moga', 'mas', 'moonton', 'sehat', 'selalu', 'panjang', 'umur', 'rezeki', 'aminin']
[mangkin, 'mudah', 'kalo', 'cuma', 'main', 'mc']	[mangkin, 'mudah', 'kalo', 'cuma', 'main', 'mc']
[sangat, 'bagus', 'moontun']	[sangat, 'bagus', 'moontun']

Gambar 11. Hasil Stemming

3.3 Hasil Pelabelan Data

Pemberian label dikerjakan secara otomatis dengan pendekatan polaritas kata (*lexicon-based polarity scoring*). Dalam pendekatan ini, setiap kata dalam ulasan dibandingkan dengan dua daftar kosakata yang telah tersedia, yaitu kamus kata positif dan kamus kata negatif yang bersumber dari repositori publik masdevid/ID-OpinionWords di GitHub.

Dari total 6.542 ulasan yang telah melalui proses pelabelan, ditemukan bahwa sebanyak 1.302 ulasan tergolong dalam sentimen positif, 742 ulasan termasuk sentimen negatif, dan sisanya, yaitu 4.498 ulasan, masuk dalam kategori netral. Dikarenakan pada penelitian ini fokus antara positif dan negatif saja, maka pilah kembali ulasan tersebut. maka didapatkan 63.7% yang berlabel positif dan 36.3% negatif. Setelah itu buatlah dalam bentuk diagram *pie chart*. Warna hijau yang berlabel positif dan warna merah negatif.

Distribusi Sentimen Ulasan Pengguna (Positif vs Negatif)

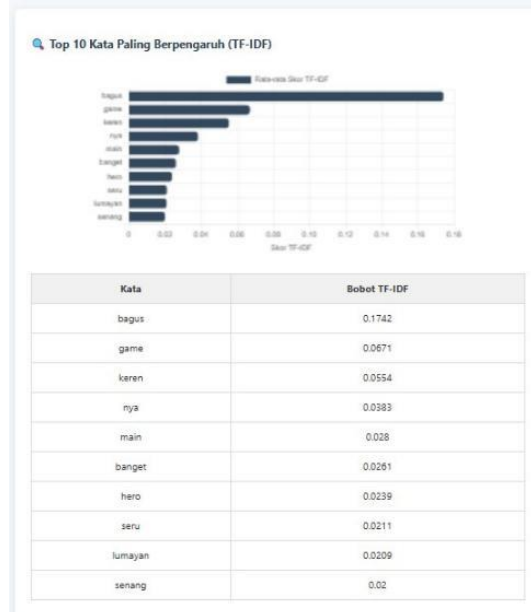


Gambar 12. Hasil Pelabelan Data

3.4 Hasil Ekstraksi Fitur

Setelah data ulasan berhasil dibersihkan, dinormalisasi, dan diberi label sentimen (positif atau negatif), tahap penting berikutnya analisis adalah ekstraksi fitur. Pada dasarnya, *machine learning* seperti *Random Forest* tidak

dapat langsung memahami teks dalam bentuk kalimat atau kata-kata. Oleh karena itu, diperlukan konversi dari data yang semula teks menjadi bentuk numerik menggunakan metode TF-IDF (*Term Frequency-Inverse Document Frequency*) dimanfaatkan untuk mengukur tingkat kepentingan suatu kata dalam sebuah dokumen relatif terhadap keseluruhan kumpulan dokumen. Kata yang sering muncul dalam satu ulasan akan diberikan nilai atau bobot yang lebih besar. Namun, jika kata tersebut juga terdapat di ulasan lainnya, bobotnya justru akan diturunkan karena dianggap kurang memberikan informasi yang spesifik.



Gambar 13. Hasil Ekstraksi Fitur

3.5 Implementasi *Random Forest*

Dalam penelitian ini, data yang telah diubah ke format numerik dengan TF-IDF dan telah memiliki label sentimen (positif atau negatif) dibagi dengan proporsi 80:20% untuk data latih dan data uji. Pembagian ini diproses dengan acak namun terkontrol dan memakai parameter *random_state* untuk mendapatkan output yang dapat direproduksi. Dengan pendekatan ini, dapat mempelajari pola dari data latih dan diuji kemampuannya dalam mengklasifikasikan sentimen dari data uji yang belum pernah dilihat sebelumnya. Didapatkan data latih sebanyak 1.635 dan data uji sebanyak 409.

Dalam implementasinya, *Random Forest* dilatih memakai data latih (X_{train} , y_{train}) yang telah dipisahkan sebelumnya dengan proporsi 80:20 untuk pelatihan dan pengujian. Klasifikasi dikonstruksi menggunakan pustaka *scikit-learn*, dengan jumlah pohon ($n_{estimators}$) diset sebanyak 100 dan parameter acak (*random_state*) diatur agar hasil eksperimen dapat direproduksi.

```

from sklearn.ensemble import RandomForestClassifier

# 1. Inisialisasi model Random Forest
rf_model = RandomForestClassifier(
    n_estimators=100, # jumlah pohon dalam hutan
    random_state=42, # untuk hasil yang konsisten
    n_jobs=-1 # gunakan semua core CPU
)

# 2. Latih model dengan data Latih
rf_model.fit(X_train, y_train)

print("✅ Model Random Forest berhasil dilatih.")

```

✅ Model Random Forest berhasil dilatih.

Gambar 14. Hasil Pelatihan *Random Forest*

3.6 Evaluasi Klasifikasi

Sebelum melakukan evaluasi klasifikasi berupa *accuracy*, *precision*, *recall* dan *F1-score*, terlebih dahulu menghitung *confusion matrix* agar terlihat nilai dari *True Positive (TP)*, *False Negative (FN)*, *False Positive (FP)*, dan *True Negative (TN)*. Diperoleh bahwa dari total data uji yang dianalisis, klasifikasi menggunakan

Random Forest menghasilkan 257 True Positive (TP), 3 False Negative (FN), 12 False Positive (FP), dan 137 True Negative (TN).

	Prediksi Positif	Prediksi Negatif
Aktual Positif	257	3
Aktual Negatif	12	137

Gambar 15. Hasil Confusion Matrix

a. Accuracy

Pada tahap ini memperlihatkan akurasi dari hasil prediksi data uji, dimana nilai kebenaran dari hasil prediksi yang dihasilkan akan dibandingkan dengan nilai aktual dari data uji tersebut.

$$Accuracy : \frac{TP + TN}{TP + TN + FP + FN} = \frac{257 + 137}{257 + 137 + 12 + 3}$$

$$= \frac{394}{409} = 0.9633 \text{ atau } 96.33\%$$

b. Precision

Metrik yang mengukur ketepatan prediksi terhadap kelas positif. Artinya, berapa presentase prediksi yang benar-benar positif.

$$Precision : \frac{TP}{TP + FP} = \frac{257}{257 + 12}$$

$$= \frac{257}{269} = 0.9554 \text{ atau } 95.54\%$$

c. Recall

Mengukur kemampuan dalam menemukan semua data aktual positif.

$$Recall : \frac{TP}{TP + FN} = \frac{257}{257 + 3}$$

$$= \frac{257}{260} = 0.9885 \text{ atau } 98.85\%$$

d. F1-Score

Rerata harmonik dari precision (presisi) dan recall.

$$F1-Score : 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.9554 \times 0.9885}{0.9554 + 0.9885}$$

$$= 2 \times \frac{0.9445}{1.9439}$$

$$= 0.9716 \text{ atau } 97.16\%$$

Tabel 3. Metrik Evaluasi

No	Metrik	Nilai
1	Accuracy (Akurasi)	96.33%
2	Precision (Presisi)	95.54%
3	Recall	98.85%
4	F1-Score	97.16%

3.7 Perbandingan dengan Studi Lainnya

Jika dibandingkan dengan studi dari Larasati et al. [15] dan Indrayanto et al. [12] yang juga menggunakan Random Forest pada aplikasi lain seperti Dana dan MyPertamina, hasil akurasi penelitian ini sedikit lebih tinggi. Hal ini diduga karena data Magic Chess bersifat lebih ekspresif dan spesifik dalam mengungkapkan kepuasan pengguna terkait performa game.

Selain itu, pada penelitian oleh Aldean et al. [4], penerapan *Random Forest* pada ulasan vaksin Sinovac menunjukkan hasil yang serupa, tetapi tantangan utama di sana adalah konteks opini yang jauh lebih kompleks secara sosial.

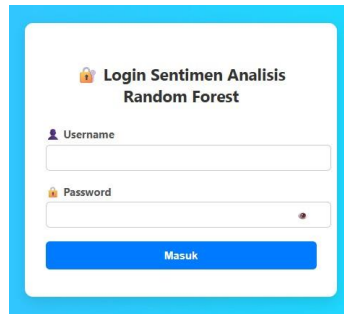
3.8 Dampak dan Implikasi

Hasil klasifikasi ini dapat menjadi acuan penting bagi pengembang *Magic Chess* dalam memahami suara pengguna secara cepat. Misalnya, munculnya kata "lag", "error", dan "bug" secara signifikan dapat dijadikan dasar prioritas dalam pengembangan update aplikasi.

Selain itu, sistem otomatisasi klasifikasi seperti ini bisa diintegrasikan ke dalam sistem monitoring *Google Play Store* secara *real-time*, untuk mendeteksi penurunan performa aplikasi berdasarkan tren sentimen.

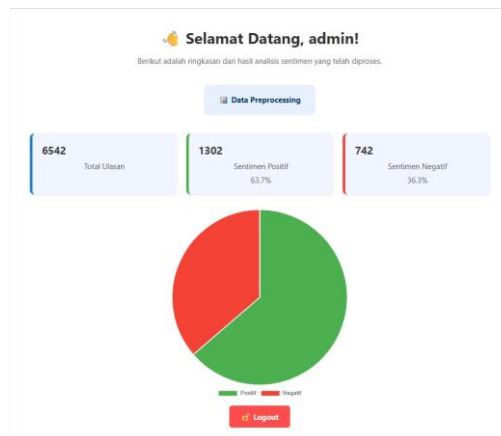
3.9 Implementasi Sistem

a. Tayangan Login



Gambar 16. Tampilan Login

b. Tayangan Dashboard



Gambar 17. Tampilan Dashboard

c. Tayangan Data Preprocessing



Gambar 18. Tampilan Data Preprocessing

d. Tayangan Latih Random Forest





Gambar 19. Tampilan Latih *Random Forest*

5. KESIMPULAN

Penelitian ini berhasil menjawab permasalahan utama terkait klasifikasi ribuan ulasan pengguna aplikasi *Magic Chess: Go Go* di Google Play Store dengan pendekatan yang efisien, akurat, dan mudah dipahami dengan menyajikan hasil klasifikasi tersebut dalam bentuk visual yang mudah dipahami oleh pengguna non-teknis. Melalui tahapan pengumpulan data, dilakukan *scraping* terhadap 6.542 ulasan pengguna selama bulan Mei 2025 menggunakan pustaka *google-play-scraper*. Selanjutnya, data diolah melalui tahapan pra-proses teks yang meliputi pembersihan data (*cleansing*), perubahan huruf menjadi kecil (*case folding*), pemecahan kalimat menjadi kata-kata (*tokenisasi*), penghapusan kata umum yang tidak penting (*stopword removal*), dan pengembalian kata ke bentuk dasar (*stemming*). Label sentimen positif dan negatif diberikan menggunakan pendekatan *lexicon-based* dengan kamus kata opini berbahasa Indonesia. Fitur dari teks kemudian diekstraksi menggunakan teknik TF-IDF agar dapat digunakan dalam pelatihan klasifikasi. *Random Forest* yang digunakan menghasilkan kinerja yang sangat baik, dengan akurasi mencapai 96,33%, *precision* sebesar 95,53%, *recall* 98,84%, dan *F1-score* 97,16%. Hasil ini menunjukkan bahwa metode ini sangat mampu membedakan antara sentimen positif dan negatif secara seimbang. Visualisasi data yang diintegrasikan ke dalam *dashboard*, seperti *confusion matrix*, *bar chart* dan *pie chart*, dan *wordcloud*, berhasil menampilkan informasi dalam format yang mudah dimengerti oleh pengguna akhir. Integrasi proses klasifikasi ke dalam *dashboard* interaktif menjadi solusi praktis dalam memudahkan proses analisis opini pengguna aplikasi secara otomatis. Pengguna dapat mengunggah data, melakukan *data preprocessing*, implementasi *Random Forest* beserta evaluasi klasifikasi tanpa memerlukan pengetahuan teknis yang mendalam. Secara keseluruhan, penelitian ini membuktikan bahwa pendekatan klasifikasi menggunakan *Random Forest* dengan dukungan *dashboard* visual mampu menjadi alat bantu analisis sentimen yang efektif, efisien, dan aplikatif dalam konteks *real-world*, khususnya untuk pengembang aplikasi yang ingin memahami persepsi pengguna dalam skala besar dan berbasis data.

UCAPAN TERIMAKASIH

Penulis menyampaikan apresiasi yang sebesar-besarnya kepada seluruh pihak yang telah memberikan dukungan dan kontribusinya dalam proses penyusunan penelitian ini.

REFERENCES

- [1] B. Adikara, "Jumlah Gamer di Indonesia Terus Meningkat, Diperkirakan Capai 192,1 Juta Orang di 2025, Salah Satu yang Paling Banyak Adalah Pemain Mobile Legends," *jawapos.com*, Oktober 2024. [Online]. Available:

- <https://www.jawapos.com/hobi-kesenangan/015230108/jumlah-gamer-di-indonesia-terus-meningkat-diperkirakan-capai-1921-juta-orang-di-2025>
- [2] Y. Valdi Seriang Ginta and S. Hardiyanto, “Kemenkominfo Sebut 35 Juta Penduduk Indonesia ‘Gamer’ Online Aktif, tapi Perputaran Uang yang Masuk Sedikit,” *kompas.com*, Oktober 2024. [Online]. Available: <https://regional.kompas.com/read/2024/10/12/180422978/kemenkominfo-sebut-35-juta-penduduk-indonesia-gamer-online-aktif-tapi>
- [3] *Magic Chess: Go Go.* (2023). Moonton. [Online]. Available: <https://play.google.com/store/apps/details?id=com.mobilechess.gp&hl=id>
- [4] M. Al Aziiz, F. Herlando, M. J. Palepa, and R. Q. Rohmansa, “Analisis Sentimen Aplikasi Mobile Legends : Bang Bang Menggunakan Latent Dirichlet Allocation,” *E-Link J. Tek. Elektro Dan Inform.*, vol. 19, no. 1, p. 1, May 2024, doi: 10.30587/e-link.v19i1.6942.
- [5] H. C. Morama, D. E. Ratnawati, and I. Arwani, “Analisis Sentimen berbasis Aspek terhadap Ulasan Hotel Tentrem Yogyakarta menggunakan Algoritma Random Forest Classifier,” 2022.
- [6] M. Y. Aldean, P. Paradise, and N. A. Setya Nugraha, “Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 di Twitter Menggunakan Metode Random Forest Classifier (Studi Kasus: Vaksin Sinovac),” *J. Inform. Inf. Syst. Softw. Eng. Appl. INISTA*, vol. 4, no. 2, pp. 64–72, June 2022, doi: 10.20895/inista.v4i2.575.
- [7] T. B. Rohman, D. D. Purwanto, and J. Santoso, “Sentiment Analysis Terhadap Review Rumah Makan di Surabaya Memanfaatkan Algoritma Random Forest,” 2022.
- [8] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, “Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest,” 2022.
- [9] C. G. Indrayanto, D. E. Ratnawati, and B. Rahayudi, “Analisis Sentimen Data Ulasan Pengguna Aplikasi MyPertamina di Indonesia pada Google Play Store menggunakan Metode Random Forest,” 2023.
- [10] A. Sagita, A. Faqih, G. Dwilestari, B. Siswoyo, and D. Pratama, “Penerapan Metode Random Forest Dalam Menganalisis Sentimen Pengguna Aplikasi Capcut Di Google Play Store,” vol. 7, no. 6, 2023.
- [11] B. B. Baskoro, I. Susanto, and S. Khomsah, “Analisis Sentimen Pelanggan Hotel di Purwokerto Menggunakan Metode Random Forest dan TF-IDF (Studi Kasus: Ulasan Pelanggan Pada Situs TRIPADVISOR),” 2021.
- [12] S. Junaidi, M. Devegi, and H. Kurniawan, “Pelatihan Pengolahan dan Visualisasi Data Penduduk menggunakan Python,” *ADMA J. Pengabd. Dan Pemberdaya. Masy.*, vol. 4, no. 1, pp. 151–162, July 2023, doi: 10.30812/adma.v4i1.2963.
- [13] K. Nistrina and L. Sahidah, “Unified Modelling Language (Uml) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru Di Smk Marga Insan Kamil,” *J. Sist. Inf.*, vol. 04, 2022.
- [14] Y. Findawati and M. A. Rosid, *Buku Ajar Text Mining*. Sidoarjo: UMSIDA PRESS, 2020. [Online]. Available: <https://press.umsida.ac.id/index.php/umsidapress/article/view/978-623-6833-19-3/770>
- [15] M. Reza *et al.*, “Artificial Intelligence : Image Processing & Application with Python,” Oktober 2022.
- [16] D. Purnomo, W. Firgiawan, and N. Nur, “Komparasi Algoritma Random Forest, Naïve Bayes, dan SVM pada Sentimen Kebijakan PPN 12%,” vol. 19, no. 2.
- [17] N. Maftucha, S. Salma, N. Rahmayuna, and N. Wakhidah, “Perbandingan Algoritma Machine Learning Dalam Memprediksi Kelulusan Siswa,” vol. 19, no. 2.