Jurnal Infomedia : Teknik Informatika, Multimedia, dan Jaringan
Volume 9, Nomor 2,Desember 2024
ISSN 2527-9858 (online), 2549-1180 (print)
Artikel Penelitian

Klasifikasi Sentimen Review Pengguna terhadap Aplikasi Instagram menggunakan Algoritma Random Forest

Mursyidah[1], Muhammad Davi[2], Suri Dheya Novitri[3*]
[1] Program Studi Teknologi Rekayasa Multimedia, Politeknik Negeri Lhokseumawe, email: mursyidah@pnl.ac.id
[2] Program Studi Teknologi Rekayasa Komputer Jaringan, Politeknik Negeri Lhokseumawe, email: muhammad.davi@pnl.ac.id
[3] Program Studi Teknologi Rekayasa Komputer Jaringan, Politeknik Negeri Lhokseumawe, email: suridheyanvtr00@gmail.com
[*] Corresponding Author: suridheyanvtr00@gmail.com

Abstrak

Penelitian ini bertujuan untuk mengklasifikasikan sentimen pengguna terhadap aplikasi Instagram dengan menggunakan algoritma Random Forest. Seiring dengan perkembangan jumlah pengguna Instagram, volume data ulasan pengguna meningkat secara signifikan, yang memberikan peluang bagi analisis sentimen untuk mengidentifikasi dan mengelompokkan sentimen positif dan negatif. Penelitian ini menggunakan dataset ulasan pengguna Instagram yang diambil dari Google Play Store, mencakup data dari tahun 2018 hingga 2023. Dengan memanfaatkan teknik Natural Language Processing (NLP) dan algoritma Random Forest, penelitian ini mengevaluasi accuracy, precision, Recall, dan F1-score dalam klasifikasi sentimen serta membandingkan kecepatan proses antara single-node dan multi-node dalam pemrosesan data. Hasil penelitian menunjukkan bahwa penggunaan multi-node mengurangi rata-rata waktu eksekusi dari 58 detik menjadi 40 detik. Selain itu, evaluasi model dengan matrik kunci menunjukkan hasil yang memuaskan dengan akurasi rata-rata sebesar 0,587, precision 0,610, Recall 0,588, dan F1-Score 0,436. Hal ini mengindikasikan bahwa algoritma Random Forest dapat memberikan kinerja yang baik dalam klasifikasi sentimen pengguna, dan penggunaan multi-node dapat mempercepat proses klasifikasi dibandingkan dengan single-node.
Kata Kunci—Review Instagram, Algoritma Random Forest, Analisis Sentimen, Single-Node, Multi-Node

Abstract

The study aims to classify the user's feelings towards the Instagram app using the Random Forest algorithm. With the growth of the number of Instagram users, the volume of user review data has increased significantly, which provides an opportunity for sentimenal analysis to identify and group positive and negative sentimens. The study used a data set of Instagram user reviews taken from the Google PlayStore, covering data from 2018 to 2023. Using Natural Language Processing (NLP) techniques and Random Forest algorithms, the study evaluated accuracy, precision, Recall, and F1 scores in sentimen classification as well as comparing process speeds between single-node and multi-nodes in data processing. The results showed that the use of multi-node reduced the average execution time from 58 seconds to 40 seconds. In addition, the evaluation of models with key metrics showed satisfactory results with an average accuracy of 0.587, precision of 0.610, remember 0.588, and a score of F1-0.436. This indicates that Random Forest algorithms can provide good performance in classifying user sentimen, and the use of multi-node can speed up the classification process compared to single-nodes.
Keywords—Instagram Reviews, Random Forest Algorithm, Sentimen Analysis, Single-Node, Multi-Node

I. PENDAHULUAN

Google PlayStore, sebagai platform distribusi resmi aplikasi Android, memudahkan pengguna menginstal berbagai aplikasi, termasuk Instagram. Instagram, populer di kalangan milenial, menjadi platform utama untuk berbagi foto dan video dengan fitur filter digital yang membuat pengalaman visual menarik dan kreatif. Pertumbuhan pengguna Instagram menghasilkan peningkatan volume data, terutama dari ulasan pengguna [1].

Seiring berjalannya waktu, Instagram terus meningkatkan fungsionalitas dengan memperbarui aplikasinya, namun seringkali pembaruan ini tidak memuaskan sebagian pengguna. Pembaruan tersebut memicu berbagai sentimen yang dapat dianalisis melalui Natural Language Processing (NLP) untuk mengidentifikasi dan mengelompokkan sentimen [2].

Random Forest, sebagai Algoritma yang popular, telah menunjukkan kinerja yang baik dalam analisis sentimen. Pada penelitian sebelumnya yang dilakukan oleh Huda, dkk, membuktikan keunggulan Random Forest dalam analisis sentimen [3]. Pada penelitian lain yang dilakukan oleh Nugroho, dkk, membukti bahwa klasifikasi menggunakan single-node lebih efektif dibandingkan dengan klasifikasi tanpa Spark [4]. Salah satu framework terdistribusi yang dirancang dalam mengolah data dengan volume data yang besar disebut Spark. Untuk mempercepat proses pengolahan

data analisis senitimen terhadapa review pengguna dan mencapai tingkat akurasi yang tinggi, peneliti memilih mengadopsi pendekatan menggunakan multi-node.

A. Analisis Sentimen

Analisis sentimen adalah teknologi yang digunakan untuk memahami dan mengekstrak pandangan serta opini dari media terkait layanan, produk, atau organisasi. Proses ini mengelompokkan teks ke dalam kategori positif, negatif, atau netral menggunakan algoritma dan model bahasa alami, memberikan wawasan tentang reaksi masyarakat terhadap entitas tertentu [5]. Langkah-langkah utama dalam analisis sentimen mencakup pemrosesan bahasa alami, ekstraksi fitur, dan pembuatan model klasifikasi, yang membantu bisnis dan organisasi untuk melacak opini konsumen, menyelidiki umpan balik produk, dan membuat keputusan berbasis persepsi publik [3]

B. Instagram

Instagram adalah salah satu jejaring sosial paling populer di dunia, terutama di kalangan anak muda. Diluncurkan pada tahun 2010, Instagram dengan cepat berkembang menjadi platform yang mendominasi ruang jejaring sosial global. Hingga Juli 2021, Instagram melaporkan 1,07 miliar pengguna aktif, dengan lebih dari 354 juta pengguna harian. Angka ini menunjukkan tingginya keterlibatan pengguna dan memperkuat posisinya sebagai salah satu platform paling dinamis di dunia [1].

Instagram yang awalnya populer di kalangan remaja, kini digunakan oleh berbagai usia untuk berbagi konten visual, membangun identitas, mempromosikan bisnis, dan membangun hubungan sosial. Fitur seperti Feed, Stories, dan Explore memperkuat posisinya sebagai platform berpengaruh dalam budaya digital [6].

C. Text Mining

Text mining adalah ekstraksi informasi dari sumber data teks dengan tujuan menganalisis dan mengelompokkan informasi berdasarkan kata-kata untuk memahami hubungan antar data teks [7]. Tahap awal text mining adalah preprocessing, penting untuk membersihkan dan mempersiapkan teks agar optimal untuk analisis dan hasil yang lebih akurat [3].

Preprocessing diperlukan karena teks yang beragam, seperti kesalahan ketik atau data tidak relevan, dapat memengaruhi akurasi analisis [5]. Beberapa komponen text preprocessing yang akan digunakan adalah :

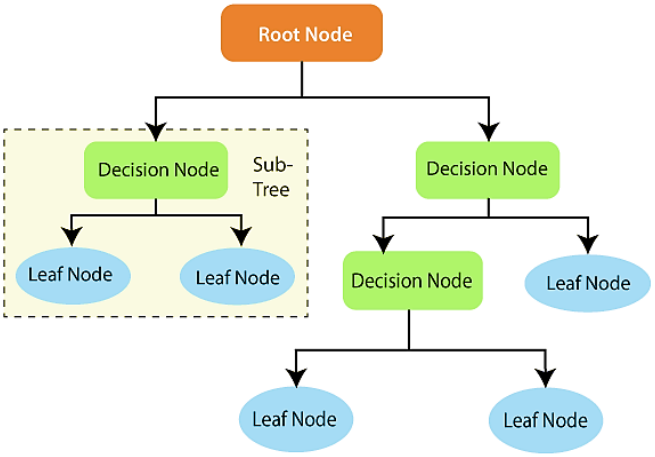
1. Case Folding : Proses mengubah seluruh huruf dalam teks menjadi huruf kecil untuk menyederhanakan dan menormalkan teks, sehingga memudahkan analisis lebih lanjut [3].
2. Stopword Removal: Menghapus kata-kata umum yang tidak memberikan informasi penting untuk analisis, seperti "dan", "atau", dan "di". Ini bertujuan mengurangi noise dalam data dan meningkatkan kualitas analisis [7].
3. Tokenize: Memecah teks menjadi unit-unit kecil, seperti kata atau frasa, yang disebut token. Tokenisasi memfasilitasi analisis lebih lanjut seperti pemodelan bahasa dan analisis sentimen [7].

D. Web scraping

Web scraping adalah teknik ekstraksi data otomatis dari halaman web menggunakan program atau skrip komputer untuk mengambil informasi dari HTML atau elemen lain di halaman web [8]. Ini digunakan untuk penelitian data publik, perbandingan harga bisnis, atau mengumpulkan konten dari situs web [3]. Langkah-langkah ekstraksi data dari halaman web meliputi mengunjungi situs, men-download kode HTML, menganalisis strukturnya, dan mengekstrak data yang dibutuhkan [8].

E. Decision Tree

Decision Tree (juga dikenal sebagai pohon keputusan) adalah struktur yang digunakan untuk membagi kumpulan data besar menjadi kumpulan data yang lebih kecil menggunakan seperangkat aturan keputusan. Tujuan dari metode ini adalah untuk mengatur dan menyusun data secara hierarki dengan membentuk struktur pohon hierarki. Proses ini melibatkan serangkaian keputusan berdasarkan atribut data, membentuk cabang dalam pohon keputusan [3].

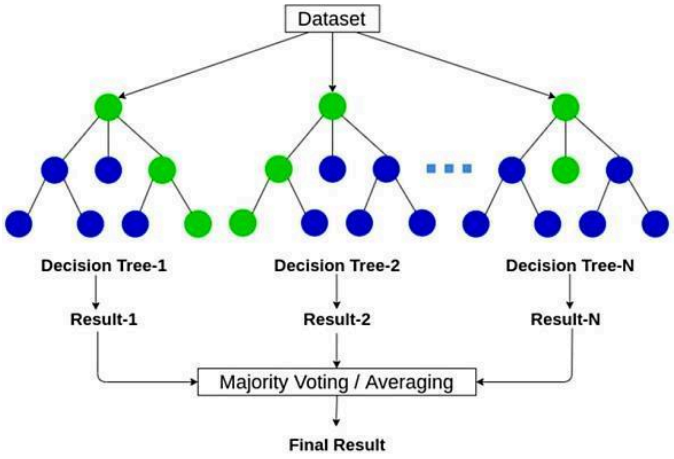


Gambar 1. Arsitektur Decision Tree

Gambar 1 menunjukkan bahwa Decision Tree membagi data menjadi kelompok lebih homogen berdasarkan atribut tertentu, dengan setiap simpul mewakili aturan dan setiap cabang mewakili kondisi. Pohon ini membantu menemukan hubungan tersembunyi antara variabel input dan target, serta membagi data ke setiap leaf tanpa kehilangan informasi [9].

F. Random Forest

Random Forest adalah metode ensemble learning yang menggunakan banyak pohon keputusan untuk regresi atau klasifikasi, cocok untuk data dan fitur yang kompleks [7].



Gambar 2. Arsitektur Random Forest

Gambar 2 menunjukkan Random Forest membangun beberapa decision tree dan menggabungkannya untuk prediksi yang lebih stabil dan akurat, menggunakan metode bagging. Klasifikasi akhir ditentukan oleh suara terbanyak dari pohon-pohon tersebut, memprediksi label teks menjadi positif, negatif, atau netral [10]. Penambahan jumlah pohon meningkatkan akurasi model, dengan "majority voting" menghasilkan prediksi lebih konsisten dan stabil. Namun, evaluasi jumlah pohon optimal diperlukan untuk keseimbangan antara akurasi dan efisiensi komputasi [7].

Metode ini membantu mengurangi risiko overfitting yang sering terjadi pada pohon keputusan tunggal. Random Forest memulai prosesnya dengan menghitung gini impurity, average gini impurity, dan information gain untuk menentukan kualitas split node [3]. Rumus mencari gini impurity ada pada persamaan 1, Rumus average gini impurity dapat dilihat pada persamaan 2, dan Rumus information gain dapat dilihat pada persamaan 3 [9].

$$Gini = 1 - \sum_{i=1}^{n} (Pi)^2 \quad (1)$$

$$AverageGiniImpurity = \frac{n}{i} \times gini \quad (2)$$

$$InformationGain = Gini\text{–}AGI \quad (3)$$

G. Confusion Matrix

Confusion matrix adalah tabel yang menggambarkan hasil klasifikasi model berdasarkan jumlah data uji yang diklasifikasikan benar dan salah. Kategori utama dalam confusion matrix meliputi True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN), yang membantu mengevaluasi kinerja model dan meningkatkan akurasi serta relevansi dalam pengklasifikasian data [3].

Tabel 1. Confusion Matrix

	Positive	*Negative*
Positive	*True Positive (TP)*	*False Positive (FP)*
Negative	*False Negative (FN)*	*True Negative (TN)*

Keterangan dari tabel 1 [11]:

1. True Positive (TP) adalah jumlah data positif yang diklasifikasikan sebagai nilai positif.
2. True Positive (TN) adalah jumlah data negatif yang diklasifikasikan sebagai nilai negatif.
3. False Posistive (FP) adalah jumlah data negatif yang diklasifikasikan sebagai nilai positif.
4. False Negative (FN) adalah jumlah data positif yang diklasifikasikan sebagai nilai negatif.

Terdapat beberapa rumus untuk menghitung confusion matrix [3], diantaranya :

1. Accuracy : Mengukur sejauh mana model melakukan klasifikasi yang benar [5]. Rumus accuracy pada persamaan 4.

2. $$Accuracy = \frac{TP+FN}{Total} \quad (4)$$

3. Precision : Mengukur seberapa tepat model dalam mengklasifikasikan instans yang diprediksi sebagai positif [5]. Rumus Precision pada persamaan 5.

4. $$Precision = \frac{TP}{TP)+FP} \quad (5)$$

5. Recall : Mengukur sejauh mana model dapat mendeteksi semua instans positif [5]. Rumus Recall pada persamaan 6.

6. $$Recall = \frac{TP}{TP+FN} \quad (6)$$

7. F1-Score : Menghitung rata-rata harmonic antara precision dan Recal [12]. Rumus F1-Score pada persamaan 7.

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall \times Precision} \quad (7)$$

H. *Apache Spark*

Apache Spark adalah platform komputasi cluster berkecepatan tinggi yang unggul dalam pemrosesan data besar dengan menyimpan data dalam memori (in-memory processing), yang mempercepat akses dan mengurangi latensi. Dengan melakukan pemrosesan dalam cluster, Spark mendistribusikan beban kerja di antara banyak node atau mesin, memungkinkan tugas-tugas kompleks diselesaikan secara paralel dan meningkatkan kecepatan eksekusi, terutama untuk analisis data dan pembelajaran mesin [13].

Selain itu, Spark mendukung bahasa pemrograman seperti Java, Scala, dan Python, memberikan fleksibilitas bagi pengembang untuk bekerja dengan bahasa yang mereka kuasai. Arsitektur terdistribusi Spark memungkinkan pemrosesan paralel di seluruh node dalam cluster, memaksimalkan penggunaan sumber daya dan mempercepat eksekusi tugas skala besar. Ini membuat Spark sangat efektif untuk beban kerja yang berat dan kompleks [4].

Apache Spark dapat dibagi menjadi dua konfigurasi utama berdasarkan skala dan distribusi sumber daya:

1. Single-Node : Dalam konfigurasi single-node, Spark berjalan pada satu mesin yang berfungsi sebagai master dan pekerja secara bersamaan. Ini umumnya digunakan untuk pengembangan, pengujian, atau analisis skala kecil, memungkinkan pemrosesan dalam memori dan akses data yang cepat tanpa memerlukan cluster penuh [13].
2. Multi-Node : Pada konfigurasi multi-node, Spark berjalan pada beberapa mesin yang bekerja sama dalam cluster. Node master menjadwalkan dan mengelola tugas, sementara node pekerja melakukan pemrosesan data. Konfigurasi ini meningkatkan kinerja dengan mendistribusikan tugas dan memungkinkan pemrosesan data dalam jumlah besar melalui paralelisme [14].

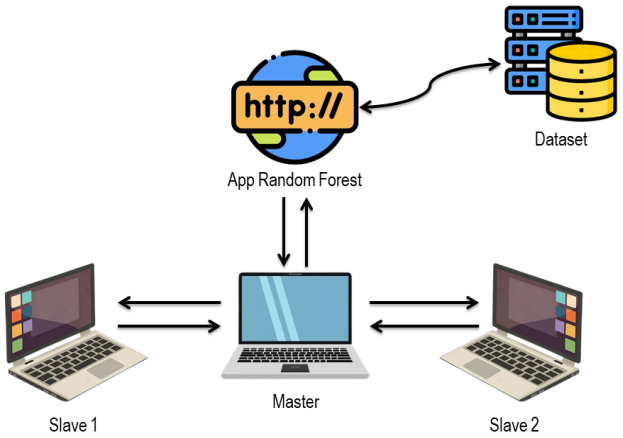
II. METODOLOGI PENELITIAN

A. Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan menggunakan dataset ulasan pengguna Instagram yang diambil dari Google PlayStore. Dataset ini dikategorikan sebagai data sekunder, karena merupakan jenis informasi yang telah dikumpulkan oleh pihak lain untuk tujuan selain dari penelitian ini, seperti analisis layanan atau umpan balik pengguna. Data ini diperoleh melalui teknik scraping, yaitu proses mengunduh dan mengekstraksi informasi dari ulasan pengguna yang tersedia di Google PlayStore.

B. Rancangan Sistem

Adapun rancangan sistem penelitian ini dapat dilihat pada Gambar 3 yang memvisualisasikan alur kerja dan komponen-komponen sistem.



Gambar 3. Arsitektur Perancangan Sistem

Penelitian ini menggunakan tools Apache Spark versi 3.5.1, yang dapat diunduh melalui tautan https://dlcdn.apache.org/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz. Berikut adalah penjelasan Gambar 3 :

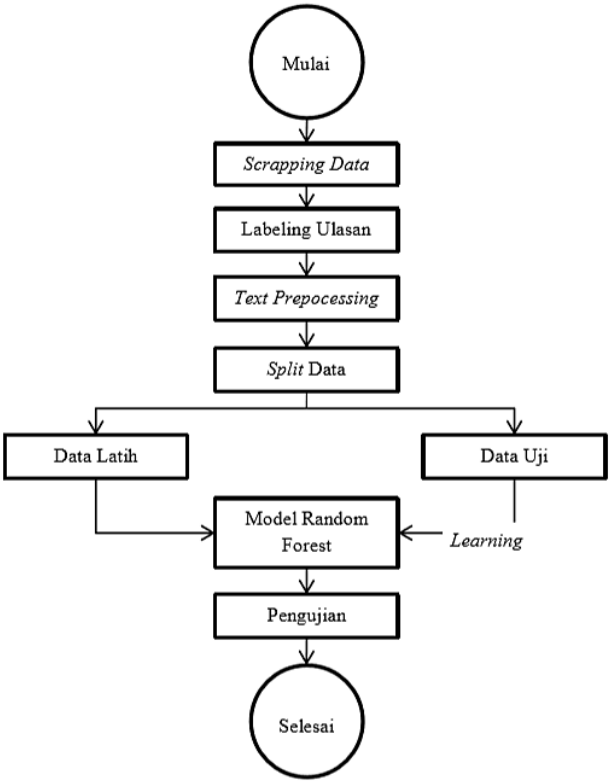
1. Dataset : Dataset yang dianalisis menggunakan Apache Spark diperoleh melalui teknik scraping dari sumber tanpa API resmi, seperti ulasan di Google PlayStore menggunakan API Google-Play-Scraper.
2. App Random Forest : Implementasi Algoritma Random Forest dalam lingkungan multi-node melibatkan pembagian tugas komputasi data besar ke beberapa node yang bekerja secara bersamaan untuk meningkatkan kecepatan dan efisiensi.
3. Master : Master adalah node utama yang berperan sebagai pengatur dan pengelola seluruh sistem. Master bertanggung jawab untuk koordinasi dan distribusi tugas komputasi ke node-node lainnya yang dikenal sebagai node pelaksana (Slaves).
4. Slave1 : Menerima tugas dari master untuk analisis sentimen, termasuk pengolahan teks, tokenisasi, ekstraksi fitur, dan klasifikasi. Setelah memproses data, Slave 1 mengirimkan hasil ke master, yang menggabungkan hasil dari semua node untuk analisis dan keputusan.
5. Slave2 : Memproses data dengan cara yang sama seperti Slave1 setelah menerima tugas dari master.

C. Metodelogi Penelitian

Pada metode penelitian ini menggunakan teknik Random Forest, yaitu metode ensemble learning yang

menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi dan mengurangi overfitting. Dataset dibagi menjadi subset menggunakan bootstrapping, dan setiap subset melatih pohon keputusan dengan pemilihan fitur secara acak.

Hasil dari pohon-pohon tersebut digabungkan melalui voting mayoritas (untuk klasifikasi) atau rata-rata (untuk regresi) untuk menghasilkan prediksi akhir. Evaluasi model dilakukan dengan matrik seperti accurasy, precision, recall, dan F1-score. Langkah-langkah metode penelitian dalam membangun arsitektur Random Forest dapat dilihat pada Gambar 4.



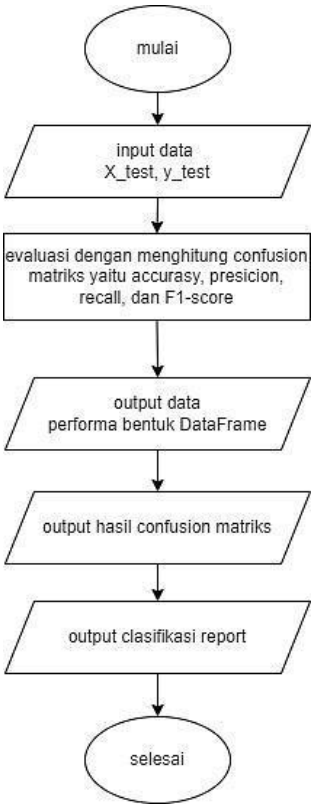
Gambar 4. Metodelogi Penelitian

Gambar 4 menggambarkan alur metodologi penelitian ulasan menggunakan model Random Forest. Tahapan metodelogi penelitian adalah sebagai berikut:

1. Mulai.
2. Scrapping Data : Mengambil data ulasan dari sumber web.
3. Labeling Ulasan : Memberikan label pada ulasan, misalnya sebagai positif atau negatif.
4. Text Preprocessing : Melakukan praproses data teks, seperti case folding, stopword removal, dan tokenize.
5. Split Data : Membagi data menjadi dua bagian: data latih dan data uji.
6. Metode Random Forest : Algoritma pembelajaran yang digunakan untuk memprediksi hasil berdasarkan data latih.
7. Pengujian : Mengukur kinerja model berdasarkan data uji.
8. Selesai.

D. Tahapan Pengujian

Pada tahapan pengujian, dilakukan dengan memasukkan data X_test dan Y_test. Model yang telah dilatih (`rf`) digunakan untuk memprediksi data testing dan hasilnya disimpan dalam variabel `rf_predicted`. Kemudian, matrik evaluasi seperti akurasi, presisi, recall, dan F1-score dihitung menggunakan fungsi dari Scikit-Learn. Dataset diuji untuk akurasi menggunakan Confusion Matrix yang mencakup Accuracy, Precision, Recall, dan F1-Score. Data dibagi menjadi data training dan data testing untuk mengukur akurasi dan kecepatan analisis sentimen. Data training sebanyak 90% dan data testing sebanyak 10%. Tahapan pengujian dapat dilihat pada Gambar 5.



Gambar 5. Tahapan Pengujian

III. HASIL DAN PEMBAHASAN

A. Hasil Scraping Data

Hasil scraping data yang dilakukan berhasil mengumpulkan data dari tahun 2018 hingga 2023 mencakup 10.000 ulasan, dimulai dari data ke-0 hingga data ke-9999. Data yang dikumpulkan mencakup beberapa variabel utama, seperti content yang berisi teks ulasan, year yang mencatat tahun ulasan, dan score yang menunjukkan penilaian pengguna. Score adalah ranting yang diberikan oleh pengguna, misal pengguna memberi rating 2 maka score-nya 2. Hasil scraping data disimpan dalam file dengan format .CSV.

Tabel 1. Hasil Scraping Data

	Content	Year	Score
0	Sering eror, selalu pembaruan terus, padahal j…	2018	2
1	Mohon durasi vidionya di tambah dan serta kone…	2018	3
2	Saya tidak bisa menggunakan fitur Ask Me a que…	2018	5
3	setiap mau mengikuti/follow org selalu error d…	2018	1
4	Ig emoticon dan superzoom sama sekali udh gk a…	2018	1
…	…	…	…
9995	Instagram saya sudah satu hari tidak bisa dig…	2023	4
9996	Sering mengalami error, tolong di perbaiki	2023	3
9997	entah kenapa sekarang Instagram saya kadang ka…	2023	4
9998	Sering di update fto2 album instagram di gale…	2023	2
9999	Kenapa Instagram saya gabisa di slide story ce…	2023	1

B. Hasil Labeling Ulasan

Hasil labeling ulasan Instagram mencakup sentimen positif, netral, dan negatif berdasarkan skor pengguna. Skor 1 dan 2 dikategorikan sebagai "Sentimen Negatif", skor 3 sebagai "Sentimen Netral", dan skor 4 serta 5 sebagai "Sentimen Positif". Pengelompokkan ini membantu dalam memahami distribusi sentimen di antara pengguna Instagram.

Tabel 2. Hasil Labeling Ulasan

	Content	Year	Score	Sentimen
0	Sering eror, selalu pembaruan terus, padahal j…	2018	2	Negatif
1	Mohon durasi vidionya di tambah dan serta kon…	2018	3	Netral
2	Saya tidak bisa menggunakan fitur Ask Me a que…	2018	5	Positif
3	setiap mau mengikuti/follow org selalu error d…	2018	1	Negatif
4	Ig emoticon dan superzoom sama sekali udh gk …	2018	1	Negatif
…	…	…	…	…
9995	Instagram saya sudah satu hari tidak bisa dig…	2023	4	Positif
9996	Sering mengalami error, tolong di perbaiki	2023	3	Netral
9997	entah kenapa sekarang Instagram saya kadang k…	2023	4	Positif
9998	Sering di update fto2 album instagram di gale…	2023	2	Negatif
9999	Kenapa Instagram saya gabisa di slide story ce…	2023	1	Negatif

C. Hasil Text Prepocessing

Hasil dari Text Prepocessing yang digunakan yaitu Case Folding, Stopword Removal, dan Tokenize. Preprocessing ini sangat penting untuk memastikan bahwa analisis teks, seperti analisis sentimen dan frekuensi kata, dapat dilakukan dengan akurasi dan efisiensi yang lebih tinggi.

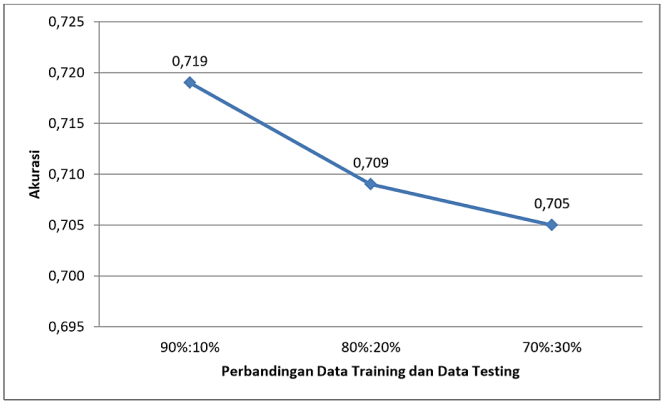
1. Case folding adalah proses pemrosesan teks untuk menyamakan huruf kapital menjadi huruf kecil, menghindari perbedaan pengenalan kata akibat kapitalisasi yang tidak konsisten. Misalnya, "Ask" dan "ask" akan diubah menjadi "ask," memastikan kata-kata serupa dikenali sebagai entitas yang sama. Proses ini meningkatkan akurasi model dengan menghilangkan variabilitas kapitalisasi dan mempersiapkan teks untuk langkah-langkah berikutnya seperti tokenisasi. Contohnya, "Saya tidak bisa menggunakan fitur Ask Me a que…" setelah case folding menjadi "saya tidak bisa menggunakan fitur ask me a que…".
2. Stopword removal adalah proses menghapus kata-kata umum yang tidak memberikan informasi signifikan, seperti "dan," "atau," dan "di," untuk menyederhanakan teks. Dengan menghapus stopwords, teks menjadi lebih fokus pada kata-kata penting, memudahkan analisis. Contohnya, dari teks asli "Saya tidak bisa menggunakan fitur Ask Me a que…" setelah stopword removal menjadi "menggunakan fitur ask me a question versi…", di mana kata-kata seperti "Saya," "tidak," dan "bisa" dihapus.
3. Tokenize adalah proses membagi teks menjadi unit-unit kecil seperti kata atau frasa, yang mempermudah analisis lebih lanjut. Misalnya, untuk teks "Sering eror, selalu pembaruan terus, padahal j…", hasil tokenisasi adalah ['eror', 'pembaruan', 'jaringan', 'bagus'], sementara untuk teks "Mohon durasi vidionya di tambah dan serta kon…", hasilnya adalah ['mohon', 'durasi', 'vidio', 'koneksi']. Meskipun tokenisasi umumnya efektif, adanya token yang tidak lengkap, seperti ['org'], dapat memengaruhi kualitas analisis dan interpretasi data. Oleh karena itu, perbaikan dalam tokenisasi sangat penting untuk meningkatkan akurasi model analisis teks.

Tabel 3. Hasil Text Preprocessing

	Content	Hasil Text Preprocessing
0	Sering eror, selalu pembaruan terus, padahal j…	[‘eror’, ‘pembaruan’, ‘jaringan’, ‘bagus’, ‘ba…
1	Mohon durasi vidionya di tambah dan serta kon…	[‘mohon’, ‘durasi’, vidionya’, ‘koneksi’, ‘ja…
2	Saya tidak bisa menggunakan fitur ask me a que…	[‘menggunakan’, ‘fitur’, ‘ask’, ‘me’, ‘a’, ‘qu…
3	setiap mau mengikuti/follow org selalu error d…	[‘mengikuti’, ‘follow’, ‘org’, ‘error’, ‘aplikasi’…
4	Ig emoticon dan superzoom sama sekali udh gk …	[‘ig’, ‘emoticon’, ‘superzoom’, ‘udh’, ‘gk’…
…	…	…
9995	Instagram saya sudah satu hari tidak bisa dig…	[‘instagram’, ‘memuat’, ‘ulang’, ‘tampilan’…
9996	Sering mengalami error, tolong di perbaiki	‘mengalami’, ‘error’, ‘tolong’, ‘perbaiki’]
9997	entah kenapa sekarang instagram saya kadang k…	[‘instagram’, ‘kadang’, ‘kalo’, ‘liat’, ‘story’…
9998	Sering di update fto2 album instagram di gale…	[‘update’, ‘foto’, ‘album’, ‘instagram’, ‘gale…
9999	Kenapa instagram saya gabisa di slide story ce…	[‘instagram’, ‘gabisa’, ‘slide’, ‘story’, ‘ce…

D. Hasil Split Data

Pembagian data dilakukan dengan menghitung jumlah data latih dan data uji dari total 10.000 data untuk menemukan rasio yang memberikan akurasi tertinggi. Tiga variasi rasio yang digunakan adalah 70:30, 80:20, dan 90:10, yang mengatur proporsi data untuk pelatihan dan pengujian model. Rasio ini mengalokasikan sebagian besar data untuk pelatihan dan sisanya untuk pengujian.



Gambar 6. Hasil Perbandingan Rasio

Berdasarkan hasil perbandingan rasio yang ditunjukkan pada Gambar 6, akurasi tertinggi diperoleh dengan menggunakan 90% data latih dan 10% data uji, mencapai nilai 0,719. Ini menunjukkan bahwa model memberikan hasil yang paling akurat ketika sebagian besar data digunakan untuk pelatihan dan hanya sedikit untuk pengujian. Sebaliknya, rasio 70% data latih dan 30% data uji menghasilkan akurasi terendah, yaitu 0,705. Rasio 80% data latih dan 20% data uji memberikan akurasi 0,709, yang lebih baik dibandingkan rasio 70:30 tetapi tidak setinggi rasio 90:10. Oleh karena itu, rasio 90% data latih dan 10% data uji dipilih sebagai konfigurasi optimal untuk penelitian ini karena memberikan performa terbaik dalam hal akurasi model.

Tabel 4. Hasil Split Data

Pembagian Split Data	
Training	9.000
Testing	1.000

E. Hasil Pengujian Single Node

Proses pengujian singel-node dilakukan pada satu perangkat, yaitu Laptop HP 245 G7 Notebook Processor AMD Ryzen 3 3250U RAM 12 GB. Tabel 4.6 menyajikan hasil Confusion Matrix dari pengujian single-node yang dilakukan pada 50 Iterasi. Confusion Matrix adalah alat evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dengan membandingkan prediksi model terhadap data yang sebenarnya. Tabel ini mencakup Accuracy, Precision, Recall, dan F1-Score.

Evaluasi performa model pada pengujian single-node menunjukkan nilai rata-rata untuk matrik Accuracy, Precision, Recall, dan F1-Score. Accuracy rata-rata adalah 0,592, menunjukkan bahwa model benar dalam sekitar 59,2% prediksi. Precision rata-rata adalah 0,556, mengindikasikan bahwa sekitar 55,6% dari prediksi positif model adalah akurat.

Hasil confusion matrix menunjukkan matrik kinerja seperti Accuracy, Precision, Recall, dan F1-Score di bawah 80%, kemungkinan karena ketidakakuratan dalam proses tokenize saat preprocessing. Token yang tidak lengkap dapat menyebabkan kesalahan interpretasi data, menurunkan kualitas analisis, dan berimbas pada kinerja model yang kurang optimal.

Tabel 5. Hasil Pengujian Single-Node

Iterasi	**Accuracy**	**Precision**	***Recall***	**F1-Score**
1	0,592	0,604	0,592	0,446
2	0,568	0,471	0,568	0,416
3	0,601	0,601	0,601	0,454
4	0,590	0,528	0,590	0,440
…	…	…	…	…
46	0,592	0,350	0,592	0,400
47	0,592	0,505	0,592	0,445
48	0,587	0,611	0,587	0,440
49	0,596	0,617	0,596	0,446
50	0,581	0,618	0,581	0,432
Rata-Rata	**0,592**	**0,556**	**0,573**	**0,439**

F. Hasil Pengujian Multi Node

Pengujian multi-node dilakukan menggunakan beberapa perangkat, dimana Laptop HP 245 G7 Notebook Processor

AMD Ryzen 3 3250U RAM 12 GB sebagai master, Laptop Acer Aspire 4739 dengan RAM 2 GB dan prosesor Intel Core i3 sebagai Slave1, serta Laptop Acer ES1-432-C3V7 dengan RAM 6 GB dan prosesor Intel Inside sebagai Slave2. Proses pengujian ini bertujuan untuk mengevaluasi performa dan fungsionalitas masing-masing node dalam sistem terdistribusi sebelum diintegrasikan dalam skala yang lebih besar.

Berdasarkan data pengujian, model memiliki rata-rata Accuracy sebesar 0,586, Precision 0,614, dan Recall 0,586. Ini menunjukkan bahwa model secara umum akurat dan cukup baik dalam identifikasi positif, dengan Precision yang baik dalam memprediksi positif dan Recall yang solid dalam menangkap instance positif. Namun, nilai F1-Score rata-rata 0,435 menunjukkan adanya ruang untuk perbaikan dalam keseimbangan antara Precision dan Recall.

Hasil confusion matrix menunjukkan bahwa model di sistem multi-node juga mengalami penurunan performa serupa dengan sistem single-node, dengan matrik seperti Accuracy, Precision, Recall, dan F1-Score di bawah 80%. Penurunan ini mungkin disebabkan oleh tokenisasi yang tidak lengkap, yang mengakibatkan hilangnya informasi penting dan mengurangi konteks serta makna data. Meskipun sistem multi-node menawarkan kapasitas pemrosesan yang lebih baik, kualitas data tetap penting. Kesalahan dalam tokenisasi mempengaruhi data pelatihan dan evaluasi, mengakibatkan penurunan akurasi dan performa model secara keseluruhan.

Tabel 6. Hasil Pengujian Multi-Node

Iterasi	Accuracy	Precision	*Recall*	F1-Score
1	0,583	0,619	0,581	0,429
2	0,576	0,617	0,576	0,426
3	0,584	0,618	0,584	0,432
4	0,585	0,600	0,585	0,435
...	…	…	…	…
46	0,591	0,613	0,591	0,442
47	0,601	0,617	0,599	0,484
48	0,588	0,618	0,588	0,437
49	0,574	0,610	0,573	0,420
50	0,597	0,606	0,594	0,457
Rata-Rata	**0,586**	**0,614**	**0,586**	**0,435**

G. Hasil Perbandingan Waktu Eksekusi Single Node dan Multi Node

Berdasarkan data eksekusi, konfigurasi multi-node menunjukkan efisiensi waktu yang lebih baik dibandingkan single-node. Pada single-node, rata-rata waktu eksekusi adalah 58 detik per iterasi, dengan variasi antara 51 hingga 60 detik. Sebaliknya, multi-node memiliki rata-rata waktu eksekusi 40 detik per iterasi, dengan variasi antara 39 hingga 48 detik. Penggunaan beberapa node mempercepat proses dan meningkatkan efisiensi, serta meningkatkan skalabilitas sistem. Namun, pengelolaan multi-node memerlukan perhatian pada konfigurasi dan monitoring untuk memastikan kehandalan dan optimalisasi sumber daya.

Tabel 7. Hasil Perbandingan Waktu Eksekusi Single-Node Dan Multi Node

Iterasi	Waktu Eksekusi (detik)	
	Single-node	*Multi-node*
1	51	41
2	54	39
3	54	40
4	53	40
...	…	…
46	52	43
47	60	48
48	60	44
49	59	39
50	53	43
Rata-Rata	**58**	**40**

IV. KESIMPULAN

Berdasarkan penelitian terhadap klasifikasi review pengguna Instagram, akurasi rata-rata pada single-node mencapai 59,2%, sedikit lebih tinggi dibandingkan multi-node yang memiliki akurasi 58,6%. Meski begitu, multi-node

unggul dalam presisi (61,4% vs. 55,6%) dan recall (58,6% vs. 57,3%). Perbedaan F1-score antara keduanya tidak signifikan, yakni 0,439 untuk single-node dan 0,432 untuk multi-node. Ini menunjukkan bahwa meskipun multi-node lebih unggul dalam presisi dan recall, performa keseluruhan tetap seimbang. Dalam hal kecepatan, waktu eksekusi rata-rata pada single-node adalah 58 detik, sedangkan multi-node hanya membutuhkan 40 detik. Penggunaan multi-node terbukti lebih efisien dalam memproses data besar, memotong waktu eksekusi hampir setengah dibandingkan single-node.

Saran untuk penelitian selanjutnya adalah menggunakan dataset yang sama dengan menambahkan lebih dari dua slave untuk mengevaluasi pengaruh penambahan slave terhadap kecepatan kinerja sistem multi-node. Selain itu, disarankan mempertimbangkan teknik preprocessing tambahan seperti stemming, lemmatization, dan normalization untuk meningkatkan akurasi model yang dihasilkan.

REFERENSI

H. Santoso, R. A. Putri, dan S. Sahbandi, "Deteksi Komentar Cyberbullying pada Media Sosial Instagram Menggunakan Algoritma Random Forest," J. Manaj. Inform. JAMIKA, vol. 13, no. 1, hlm. 62–72, 2023, doi: 10.34010/jamika.v13i1.9303.

A. Andreyestha dan A. Subekti, "Analisa Sentiment Pada Ulasan Film Dengan Optimasi Ensemble Learning," J. Inform., vol. 7, no. 1, hlm. 15–23, 2020, doi: 10.31311/ji.v7i1.6171.

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," J. Ilm. Inform., vol. 11, no. 02, hlm. 150–158, 2023, doi: 10.33884/jif.v11i02.7952.

R. A. Nugroho, I. Cholissodin, dan Indriati, "Implementasi Naïve Bayes Classifier untuk Klasifikasi Emosi Tweet Berbahasa Indonesia pada Spark," J. Pengemb. Teknol. Inf. Dan Ilmu Komput., vol. 5, no. 1, hlm. 301–310, 2021.

N. Ika, P. Kalingara, O. N. Pratiwi, dan H. D. Anggana, "Analisis Sentimen Review Customer Terhadap Layanan Ekspedisi Jne Dan J & T Express Menggunakan Metode Naïve Bayes Sentiment Analysis Review Customer of Jne and J & T Express Expedition Services Using Naïve Bayes Method," E-Proceeding Eng., vol. 8, no. 5, hlm. 9035–9048, 2021.

I. D. Aryani dan D. Murtiariyati, "Instagram Sebagai Media Promosi Dalam Meningkatkan Jumlah Penjualan Pada A.D.A Souvenir Project," J. Ris. Akunt. Dan Bisnis Indones. STIE Widya Wiwaha, vol. 2, no. 2, hlm. 466–477, Jun 2022.

F. A. Larasati, D. E. Ratnawati, dan B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," … Teknol. Inf. Dan …, vol. 6, no. 9, hlm. 4305–4313, 2022.

C. G. Indrayanto, D. E. Ratnawati, dan B. Rahayudi, "Analisis Sentimen Data Ulasan Pengguna Aplikasi MyPertamina di Indonesia pada Google Play Store menggunakan Metode Random Forest," J. Pengemb. Teknol. Inf. Dan Ilmu Komput., vol. 7, no. 3, hlm. 1131–1139, 2023.

S. H. Sitorus dan U. Ristian, "Penerapan Metode Decision Tree Untuk Mengklasifikasi Mutu Buah Jeruk Berdsarkan Fitur Warna Dan Ukuran," J. Komput. Dan Apl., vol. 09, no. 01, hlm. 76–86, 2021.

T. F. Basar, D. E. Ratnawati, dan I. Arwani, "Analisis Sentimen Pengguna Twitter terhadap Pembayaran Cashless menggunakan Shopeepay dengan Algoritma Random Forest".

I. Afdhal, R. Kurniawan, I. Iskandar, R. Salambue, E. Budianita, dan F. Syafria, "Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia," J. Nas. Komputasi Dan Teknol. Inf., vol. 5, no. 1, hlm. 122–130, 2022.

I. R. Prabaswara dan R. Saputra, "Analisis Data Sosial Media Twitter Menggunakan Hadoop dan Spark," IT J. Res. Dev., vol. 4, no. 2, Mar 2020, doi: 10.25299/itjrd.2020.vol4(2).4099.

S. Oliviandi, A. B. Osmond, dan R. Latuconsina, "Implementasi Apache Spark Pada Big Data Berbasis Hadoop Sistributed File System," E-Proceeding Eng., vol. 5, no. 1, hlm. 1005–1012, 2019.

C. Wibawa, S. Wirawan, M. Mustikasari, dan D. T. Anggraeni, "Komparasi Kecepatan Hadoop MapReduce dan Apache Spark Dalam Mengolah Data Teks," J. Ilm. Matrik, vol. 24, no. 1, hlm. 10–20, Apr 2022, doi: 10.33557/jurnalmatrik.v24i1.1649.