JURNAL TIMES
TECHNOLOGY INFORMATICS & COMPUTER SYSTEM

Volume XIV No 2, Desember 2025
*p*ISSN : 2337 – 3601
*e*ISSN : 2549 – 015X

Tersedia *online* di http://ejournal.stmik-time.ac.id

ANALISIS KOMPARATIF EFEKTIVITAS *PIPELINE DATA CLEANING* BERBASIS ATURAN DAN LEMMATISASI UNTUK KLASIFIKASI SENTIMEN

Ahmad Fauzidan Yahya Khainur[1)], Muhammad Hafiz Fathurrohman[2)], Taufiqurrohman Yuares[3)], Widianingsih[4)], Chaerur Roziki[5)]
Program Studi Informatika
Universitas Singaperbangsa Karawang
Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361
email: zidanhkainur2@gmail.com[1)], hfz581@gmail.com[2)] , taufiqyuarez@gmail.com[3)], widiasw258@gmail.com[4)], chaerur.rozikin@staff.unsika.ac.id[5)]

Abstrak
Pertumbuhan data teks tidak terstruktur menuntut metode pra-pemrosesan (*preprocessing*) yang efektif untuk analisis sentimen. Penelitian ini mengembangkan dan membandingkan dua *pipeline* automasi pembersihan data (*data cleaning*) berbasis Python menggunakan dataset *IMDB Movie Reviews* (50.000 sampel). Pipeline pertama menerapkan pendekatan Berbasis Aturan (*Rule-Based*) menggunakan ekspresi reguler (*Regex*), sedangkan pipeline kedua menerapkan pendekatan Berbasis Lemmatisasi menggunakan pustaka NLTK. Kualitas data hasil pembersihan dievaluasi menggunakan algoritma *Multinomial Naive Bayes* dan *Logistic Regression* dengan ekstraksi fitur TF-IDF (Unigram dan Bigram). Hasil eksperimen menunjukkan bahwa pendekatan Berbasis Aturan (Regex) secara signifikan lebih efisien dalam waktu komputasi (8,87 detik vs 38,43 detik) dan menghasilkan akurasi yang sedikit lebih tinggi (89,43% vs 88,93% pada *Logistic Regression*) dibandingkan pendekatan Lemmatisasi. Penelitian ini menyimpulkan bahwa untuk analisis sentimen pada dataset ulasan film berskala besar, pembersihan data sederhana berbasis pola (*pattern-based*) lebih efektif dan efisien daripada normalisasi linguistik yang kompleks, serta menegaskan pentingnya pemilihan teknik *preprocessing* yang tepat dalam siklus hidup rekayasa data.
Kata Kunci : Analisis Sentimen, Pembersihan Data, Automasi, Python, Naïve Bayes

1. Pendahuluan

Era digital modern ditandai oleh produksi dan penyebaran data dalam volume yang belum pernah terjadi sebelumnya. Sebagian besar data ini, yang berasal dari platform media sosial, ulasan produk, forum daring, dan artikel berita, berbentuk teks tidak terstruktur. Ledakan data tekstual ini menyimpan wawasan berharga mengenai opini, sentimen, dan emosi publik terhadap berbagai entitas, seperti produk, layanan, atau kebijakan [1]. Untuk mengekstrak wawasan ini secara efisien, lahirlah bidang analisis sentimen, yang juga dikenal sebagai *opinion mining*. Analisis sentimen adalah cabang dari *text mining* yang secara komputasional mengidentifikasi dan mengkategorikan opini yang diekspresikan dalam sebuah teks untuk menentukan apakah sikap penulis terhadap suatu topik bersifat positif, negatif, atau netral [2].

Aplikasi praktis dari analisis sentimen sangat luas dan berdampak signifikan di berbagai sektor. Dalam dunia bisnis, perusahaan memanfaatkannya untuk memahami umpan balik pelanggan, memantau reputasi merek secara real-time, dan melakukan riset pasar yang mendalam. Di ranah sosial dan politik, analisis sentimen digunakan untuk mengukur opini publik terhadap kebijakan pemerintah atau kandidat politik, serta melacak perkembangan isu-isu sosial yang sedang tren [3]. Dengan demikian, kemampuan untuk menganalisis sentimen secara akurat dan dalam skala besar telah menjadi sebuah kebutuhan strategis.

Keberhasilan setiap proyek berbasis *machine learning*, terutama dalam domain *Natural Language Processing* (NLP), sangat bergantung pada kualitas data yang digunakan untuk melatih model [4]. Data mentah yang dikumpulkan dari sumber-sumber daring sering kali "kotor" (*noisy*), mengandung berbagai macam inkonsistensi seperti kesalahan ketik, penggunaan singkatan non-standar, format yang tidak seragam, serta elemen-elemen irelevan lainnya seperti tag HTML, emoji, atau URL [5]. Kualitas data yang rendah ini menjadi penghalang utama dalam mencapai kinerja model yang optimal.

Di sinilah peran *data cleaning* atau pembersihan data menjadi fundamental. *Data cleaning*, yang juga dikenal sebagai *text preprocessing* atau *data wrangling*, adalah proses sistematis untuk mengidentifikasi dan memperbaiki atau menghapus data yang salah, tidak lengkap, tidak relevan, atau tidak akurat dari sebuah dataset [6]. Dalam konteks analisis sentimen, kehadiran *noise* seperti kesalahan eja atau tanda baca yang tidak konsisten dapat secara langsung mendistorsi polaritas sentimen sebuah teks, yang pada akhirnya mengarah pada hasil klasifikasi yang keliru dan kesimpulan yang tidak dapat diandalkan [5]. Oleh karena itu, *data cleaning* bukanlah sekadar langkah awal yang trivial, melainkan sebuah proses kritis yang menentukan validitas dan keandalan seluruh alur kerja analisis.

Meskipun urgensinya diakui secara luas, proses *data cleaning* secara tradisional sering kali dilakukan secara manual atau semi-manual. Pendekatan ini memiliki beberapa kelemahan signifikan: memakan banyak waktu, membutuhkan sumber daya manusia yang intensif, rentan terhadap kesalahan subjektif (*human error*), dan sulit

untuk direplik asi secara konsisten pada dataset yang berbeda atau lebih besar [7]. Ketergantungan pada intervensi manual ini menciptakan sebuah kesenjangan (*gap*) antara kebutuhan akan analisis sentimen yang cepat dan berskala besar dengan praktik persiapan data yang lambat dan tidak efisien.

Kebaruan (*novelty*) dari penelitian ini terletak pada pengembangan sebuah pipeline automasi yang terstruktur dan komprehensif untuk proses *data cleaning* teks menggunakan bahasa pemrograman Python. Dengan merangkai serangkaian langkah pemrosesan teks standar ke dalam sebuah skrip yang dapat dieksekusi secara otomatis, penelitian ini menawarkan solusi yang efisien, konsisten, dan dapat direproduksi untuk mengatasi tantangan data kotor [8]. Fokus utama pada automasi tidak hanya bertujuan untuk meningkatkan efisiensi, tetapi juga untuk meningkatkan rigor ilmiah dari proses analisis sentimen. Sebuah skrip otomatis menjamin bahwa setiap data diproses dengan aturan yang sama persis, menghilangkan subjektivitas dan memungkinkan peneliti lain untuk mereplikasi dan memvalidasi hasil penelitian dengan tepat [9]. Kontribusi ini melampaui sekadar efisiensi operasional; ia memperkuat fondasi ilmiah dari penelitian NLP dengan mempromosikan transparansi dan reproduktifitas.

2. Landasan Teori

Analisis Sentimen dan NLP

Analisis sentimen, atau *opinion mining*, adalah bidang studi yang menganalisis opini, sentimen, evaluasi, dan emosi orang terhadap entitas seperti produk, layanan, dan topik [3]. Ini adalah cabang dari *text mining* dan *Natural Language Processing* (NLP) yang bertujuan untuk secara otomatis menentukan nuansa emosional di balik sebuah teks [1]. NLP sendiri merupakan sub-bidang kecerdasan buatan yang berfokus pada interaksi antara komputer dan bahasa manusia, memungkinkan mesin untuk membaca, memahami, dan menafsirkan teks [10].

Keberhasilan analisis sentimen sangat bergantung pada kualitas data. Data teks mentah dari sumber seperti media sosial seringkali tidak terstruktur dan mengandung *noise* (misalnya, kesalahan ketik, slang, tag HTML, emoji) [5]. Tanpa pembersihan yang tepat, *noise* ini dapat mendistorsi makna dan mengarah pada klasifikasi sentimen yang tidak akurat [5]. Oleh karena itu, *text preprocessing* atau *data cleaning* adalah langkah fundamental dalam NLP [11]. Proses ini melibatkan serangkaian teknik untuk menstandardisasi dan membersihkan teks, termasuk:

- Case Folding
  Mengubah semua teks menjadi huruf kecil untuk memastikan konsistensi [12].
- Noise Removal
  Menghapus karakter yang tidak relevan seperti tanda baca, angka, simbol, dan URL [13].
- Tokenization
  Memecah teks menjadi unit-unit yang lebih kecil (token), biasanya kata-kata [10].
- Stopword Removal
  Menghilangkan kata-kata umum yang tidak memiliki makna sentimen signifikan (misalnya, "dan", "di", "yang") [10].
- Stemming/Lemmatization
  Mengurangi kata ke bentuk dasarnya untuk mengelompokkan kata-kata yang memiliki makna serupa [5].

Representasi Teks (Text Representation)

Setelah teks dibersihkan, teks tersebut harus diubah menjadi format numerik yang dapat diproses oleh algoritma *machine learning*. Proses ini disebut ekstraksi fitur atau representasi teks. Terdapat beberapa metode untuk melakukan ini, salah satu yang paling umum adalah model *Bag-of-Words* (BoW) [14]. Dalam model ini, setiap dokumen direpresentasikan sebagai vektor frekuensi kata.

Salah satu skema pembobotan yang populer dalam kerangka BoW adalah Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF mengevaluasi seberapa relevan sebuah kata terhadap sebuah dokumen dalam suatu koleksi dokumen (korpus). Bobot ini meningkat seiring dengan jumlah kemunculan kata dalam dokumen (TF) tetapi diimbangi oleh frekuensi kata tersebut di seluruh korpus (IDF). Kata-kata yang sering muncul di banyak dokumen akan memiliki bobot IDF yang rendah, sehingga mengurangi pengaruh kata-kata umum dan memberikan penekanan lebih pada kata-kata yang lebih unik dan informatif untuk dokumen tertentu [14]. Representasi numerik ini kemudian menjadi masukan untuk melatih model klasifikasi.

Algoritma Machine Learning untuk Analisis Sentimen

Multinomial Naive Bayes (MNB)

Algoritma ini adalah salah satu pilihan populer untuk klasifikasi teks karena kesederhanaan dan efisiensinya [15]. MNB didasarkan pada teorema Bayes dengan asumsi "naif" bahwa fitur-fitur (dalam hal ini, kata-kata) bersifat independen satu sama lain. Meskipun asumsi ini jarang benar dalam bahasa alami, MNB seringkali memberikan kinerja yang sangat baik dalam praktik, terutama untuk *baseline model* [16]. Algoritma ini bekerja dengan menghitung probabilitas sebuah dokumen termasuk dalam kelas tertentu berdasarkan distribusi kata-kata di dalamnya.

Logistic Regression (LR)

Regresi Logistik adalah algoritma klasifikasi lain yang efektif untuk analisis sentimen. Tidak seperti Naive Bayes, LR tidak membuat asumsi independensi fitur. Algoritma ini memodelkan probabilitas sebuah teks masuk

ke dalam kelas sentimen tertentu menggunakan fungsi logistik. LR dikenal karena kinerjanya yang kuat dan kemampuannya untuk memberikan probabilitas kelas yang terkalibrasi dengan baik.

3. Metode Penelitian

Kerangka Penelitian

Penelitian ini menerapkan pendekatan kuantitatif eksperimental untuk menguji efektivitas dua strategi pembersihan data (*data cleaning*) dalam analisis sentimen. Kerangka kerja penelitian dirancang secara sistematis mulai dari akuisisi data hingga evaluasi model. Alur penelitian secara garis besar meliputi:

1. Pengumpulan Data
   Mengakuisisi dataset ulasan film berskala besar dari repositori publik.
2. Pra-pemrosesan Data (*Data Preprocessing*)
   Tahap inti penelitian yang membandingkan dua pendekatan, yaitu:
   a. **Skenario 1 (Basis Aturan/Regex):** Pembersihan data menggunakan skrip kustom berbasis *Regular Expressions* (sebelumnya disebut sebagai metode Manual).
   b. **Skenario 2 (Basis Pustaka NLP):** Pembersihan data menggunakan fungsi otomatis dari pustaka NLP standar dengan lemmatisasi (sebelumnya disebut sebagai metode Otomatis).
3. Pembagian Data
   Memisahkan dataset menjadi data latih (*training set*) dan data uji (*testing set*) dengan proporsi 80:20.
4. Ekstraksi Fitur
   Mengonversi data teks menjadi representasi vektor numerik menggunakan metode TF-IDF.
5. Pelatihan Model
   Melatih model klasifikasi *Multinomial Naive Bayes* (MNB) dan *Logistic Regression* (LR).
6. Evaluasi Kinerja
   Mengukur performa model berdasarkan metrik akurasi, presisi, *recall*, F1-score, serta efisiensi waktu komputasi.

Sumber dan Pengumpulan Data

Data yang digunakan adalah data sekunder yang diperoleh melalui teknik studi dokumentasi dari repositori publik.

Dataset

Dataset yang digunakan adalah **IMDB Dataset of 50K Movie Reviews**. Dataset ini dipilih sebagai standar *benchmark* dalam literatur analisis sentimen karena karakteristiknya yang ideal untuk pengujian klasifikasi biner.

Karakteristik Data

Dataset terdiri dari 50.000 ulasan film berbahasa Inggris. Distribusi kelas sentimen dalam dataset ini sangat seimbang (*balanced dataset*), terdiri dari 25.000 ulasan berlabel positif dan 25.000 ulasan berlabel negatif. Keseimbangan ini krusial untuk mencegah bias model terhadap kelas mayoritas selama proses pelatihan. Struktur data terdiri dari dua atribut utama: review (teks ulasan) dan sentiment (label kategori).

Implementasi Pipeline Automasi Data Cleaning

Penelitian ini mengimplementasikan dua strategi pembersihan data yang dikembangkan dalam lingkungan Python. Kedua strategi tersebut dirancang sebagai fungsi modular untuk mentransformasi data mentah menjadi format yang siap untuk pemodelan.

Pipeline 1: Pendekatan Berbasis Aturan (Rule-Based/Regex)

Pendekatan ini, yang dalam eksperimen diberi label sebagai variabel clean_manual, berfokus pada pembersihan sintaksis menggunakan modul re (Regular Expression) dan string. Langkah-langkah dalam fungsi ini meliputi:

1. **Lowercasing:** Mengonversi seluruh teks menjadi huruf kecil.
2. **Noise Removal:** Menghapus tag HTML (<.*?>), URL, *mentions*, dan tagar menggunakan pola regex.
3. **Character Filtering:** Menghapus tanda baca dan angka untuk menyisakan hanya karakter alfabetis.
4. **Whitespace Normalization:** Menghapus spasi berlebih.
5. **Stopword Removal:** Menghapus kata-kata umum yang tidak bermakna menggunakan daftar *stopwords* bahasa Inggris dari pustaka NLTK.

Pipeline 2: Pendekatan Berbasis Lemmatisasi (NLP-Library)

Pendekatan ini, yang diberi label clean_auto, memperluas pipeline pertama dengan menambahkan normalisasi morfologi. Fungsi ini memanfaatkan WordNetLemmatizer dari pustaka NLTK. Proses utamanya adalah:

1. **Advanced Pattern Removal:** Menggunakan regex terkompilasi untuk penghapusan *noise* (HTML, URL, simbol) dalam satu langkah eksekusi.
2. **Lemmatization:** Mengubah kata ke bentuk dasarnya (lemma) dengan mempertimbangkan konteks morfologi kata dalam database WordNet.

3. **Token Filtering:** Melakukan tokenisasi dan penghapusan *stopwords* secara simultan menggunakan *list comprehension* untuk optimasi memori.

Ekstraksi Fitur

Data teks yang telah dibersihkan dari kedua pipeline diubah menjadi representasi numerik menggunakan TfidfVectorizer dari pustaka Scikit-learn. Konfigurasi parameter yang digunakan adalah max_features=5000 (mengambil 5.000 kata teratas) dan ngram_range=(1, 2) untuk menangkap konteks unigram dan bigram.

Pemodelan dan Skenario Pengujian

Setelah data teks dibersihkan, data tersebut perlu diubah menjadi format numerik yang dapat dipahami oleh algoritma *machine learning*.

Ekstraksi Fitur

Teknik Term Frequency-Inverse Document Frequency (TF-IDF) digunakan untuk proses ini. TF-IDF adalah metode pembobotan statistik yang merefleksikan seberapa penting sebuah kata dalam sebuah dokumen relatif terhadap keseluruhan korpus. Metode ini mengubah kumpulan teks menjadi matriks vektor fitur numerik, di mana setiap baris mewakili sebuah dokumen dan setiap kolom mewakili sebuah kata unik dalam kosakata.

Pelatihan Model

Matriks TF-IDF dari data latih, beserta label sentimen yang sesuai, digunakan untuk melatih model klasifikasi Naïve Bayes (MultinomialNB dari Scikit-learn). Selama fase ini, model mempelajari probabilitas kemunculan setiap kata dalam kelas sentimen positif dan negatif.

Evaluasi Model

Model yang telah dilatih kemudian digunakan untuk membuat prediksi sentimen pada data uji, yang belum pernah dilihat sebelumnya. Prediksi yang dihasilkan oleh model dibandingkan dengan label sentimen yang sebenarnya (ground truth) dari data uji untuk mengevaluasi seberapa baik kinerja model.

Metrik Evaluasi

Untuk mengukur kinerja model klasifikasi secara kuantitatif, digunakan empat metrik standar yang dihitung dari *confusion matrix* (matriks konfusi). Matriks ini membandingkan hasil prediksi dengan kelas aktual dan terdiri dari empat komponen: *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN).

Akurasi

Mengukur proporsi prediksi yang benar secara keseluruhan (baik positif maupun negatif) dari total prediksi. Dihitung dengan rumus:

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.1}$$

Presisi

Mengukur proporsi dari prediksi positif yang ternyata benar-benar positif. Metrik ini relevan ketika biaya dari *False Positive* tinggi. Dihitung dengan rumus:

$$Presisi = \frac{TP}{TP + FP} \tag{3.2}$$

Recall (Sensitivity)

Mengukur proporsi dari kasus positif aktual yang berhasil diidentifikasi dengan benar oleh model. Metrik ini penting ketika biaya dari *False Negative* tinggi. Dihitung dengan rumus:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

F1-Score

Merupakan rata-rata harmonik dari Presisi dan *Recall*. Metrik ini memberikan skor tunggal yang menyeimbangkan kedua metrik tersebut, sangat berguna terutama jika terdapat ketidakseimbangan kelas dalam dataset. Dihitung dengan rumus:

$$F1\text{-}Score = 2 \times \frac{(Presisi \times Recall)}{(Presisi + Recall)} \tag{3.4}$$

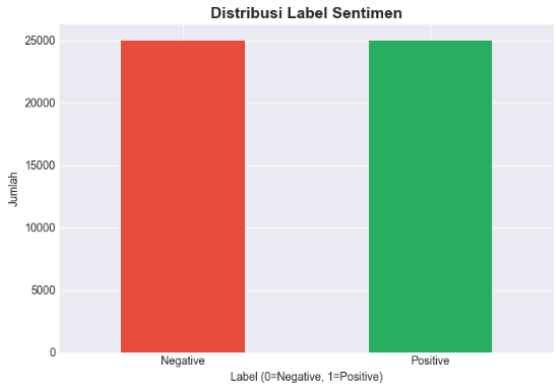
4. Hasil Penelitian

Hasil Eksperimen

Eksplorasi dan Visualisasi Data Awal

Dataset yang digunakan dalam penelitian ini adalah IMDB Dataset of 50K Movie Reviews, bukan 1001 ulasan seperti yang mungkin disebutkan dalam draf awal. Dataset ini terdiri dari 50.000 ulasan film dalam bahasa Inggris. Analisis distribusi sentimen awal menunjukkan bahwa dataset ini seimbang, terdiri dari 25.000 ulasan

positif (50.0%) dan 25.000 ulasan negatif (50.0%). Keseimbangan kelas ini penting untuk menghindari bias model terhadap kelas mayoritas. Visualisasi distribusi label sentimen dapat dilihat pada Gambar 1

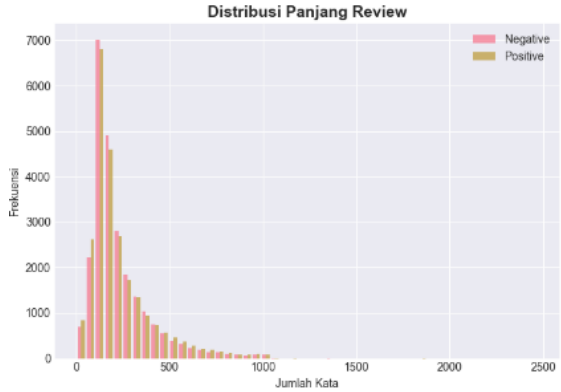


Gambar 1. Visualisasi distribusi label sentimen

Statistik deskriptif teks ulasan menunjukkan variasi panjang yang signifikan:

- Rata-rata kata per ulasan: 231.16
- Rata-rata karakter per ulasan: 1309.43
- Ulasan terpendek: 4 kata
- Ulasan terpanjang: 2470 kata

Distribusi panjang ulasan (jumlah kata) juga divisualisasikan menggunakan histogram (Gambar 2), yang menunjukkan mayoritas ulasan memiliki panjang antara 100 hingga 500 kata.



Gambar 1. Visualisasi distribusi panjang review

Hasil Pipeline Data Cleaning

Efisiensi komputasi dari kedua pendekatan diukur berdasarkan waktu eksekusi pada total 50.000 data ulasan. Hasil pengukuran tercatat sebagai berikut:

- **Waktu Pipeline Berbasis Aturan (Regex):** 8,87 detik.
- **Waktu Pipeline Berbasis Lemmatisasi:** 38,43 detik.

Hasil ini menunjukkan bahwa pipeline berbasis Lemmatisasi memakan waktu sekitar 333% lebih lama dibandingkan pendekatan Regex sederhana. Peningkatan waktu ini disebabkan oleh kompleksitas komputasi algoritma lemmatisasi yang harus melakukan pencarian (*lookup*) ke korpus WordNet untuk setiap token, berbeda dengan operasi regex yang bekerja pada level pencocokan pola string dasar.

Ekstraksi Fitur

Setelah pembersihan, teks dikonversi menjadi representasi numerik menggunakan TF-IDF. TfidfVectorizer dari Scikit-learn digunakan dengan parameter max_features=5000 (membatasi kosakata hingga 5000 fitur teratas) dan ngram_range=(1, 2) (mempertimbangkan unigram dan bigram). Hasilnya adalah matriks fitur TF-IDF dengan dimensi (40000, 5000) untuk data latih.

Kinerja Model Klasifikasi

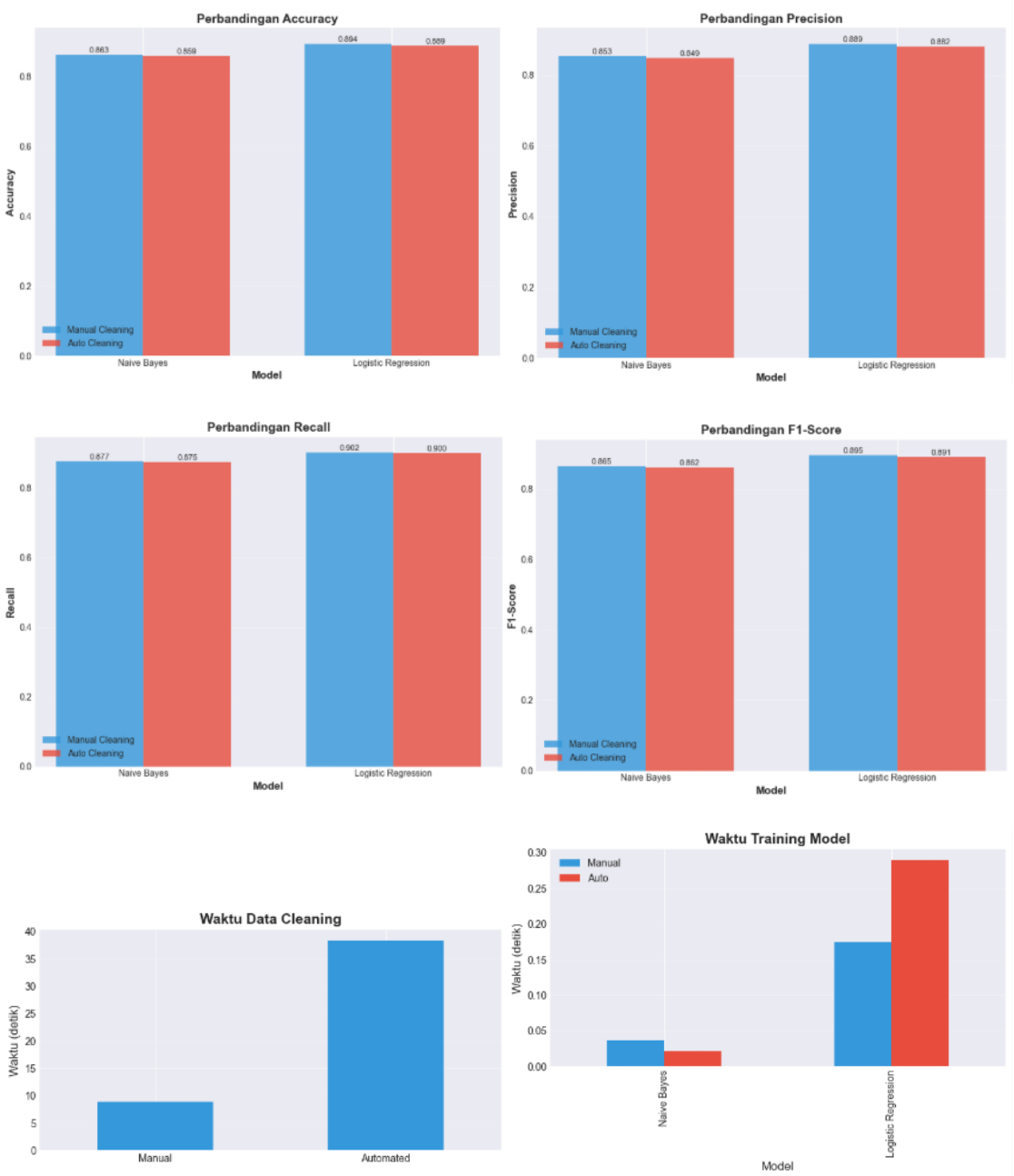
Dua algoritma klasifikasi, Multinomial Naive Bayes (MNB) dan Logistic Regression (LR), dilatih dan diuji menggunakan data hasil cleaning manual dan otomatis. Dataset dibagi menjadi 80% data latih (40.000 sampel) dan 20% data uji (10.000 sampel) secara stratifikasi. Kinerja model dievaluasi menggunakan metrik Akurasi, Presisi, Recall, dan F1-Score. Hasil evaluasi pada data uji dirangkum dalam **Tabel 1**. Validasi silang 5-fold (*5-Fold Cross-Validation*) juga dilakukan pada data latih untuk mengukur robustisitas model, yang rinciannya disajikan dalam **Tabel 2**.

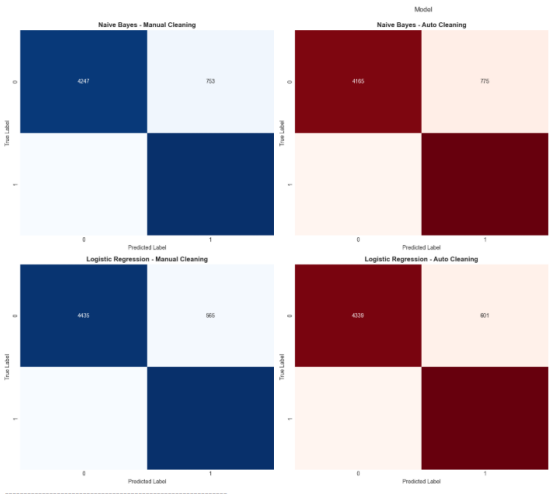
Tabel 1 Hasil Kinerja Model Klasifikasi

Model	Metode Cleaning	Akurasi	Presisi	Recall	F1-Score
Naive Bayes	Manual	0.8633	0.8535	0.8772	0.8652
	Otomatis	0.8589	0.8489	0.8746	0.8616
Logistic Regression	Manual	0.8943	0.8886	0.9016	0.8951
	Otomatis	0.8893	0.8817	0.9001	0.8908

Tabel 2 Hasil Kinerja Model Klasifikasi (Lanjutan)

Model	Metode Cleaning	Waktu Latih (s)	Skor CV 5-Fold (Mean ± Std)
Naive Bayes	Manual	0.04	0.8583 ± 0.0028
	Otomatis	0.02	0.8588 ± 0.0024
Logistic Regression	Manual	0.17	0.8889 ± 0.0034
	Otomatis	0.29	0.8858 ± 0.0014





Uji signifikansi statistik menggunakan Paired T-Test dilakukan untuk membandingkan skor akurasi antara metode cleaning manual dan otomatis pada kedua model. Hasilnya menunjukkan T-statistic = 14.4888 dan P-value = 0.0439.

Pembahasan

Hasil eksperimen memberikan beberapa wawasan penting mengenai perbandingan antara pipeline data cleaning manual dan otomatis dalam konteks analisis sentimen pada dataset IMDB.

Analisis Performa Model

Evaluasi model menggunakan *Logistic Regression* menunjukkan bahwa data yang diproses dengan pendekatan Berbasis Aturan (Regex) menghasilkan akurasi 89,43%, sedikit lebih unggul dibandingkan pendekatan Lemmatisasi yang mencapai 88,93%. Uji statistik *Paired T-Test* mengonfirmasi bahwa perbedaan ini signifikan secara statistik (p-value = 0.0439 < 0.05). Fenomena ini mengindikasikan bahwa pada dataset ulasan film IMDB, bentuk asli kata (*inflected form*) mungkin mengandung sinyal sentimen yang penting yang hilang saat kata dinormalisasi menjadi lemma. Misalnya, kata "boring" (kata sifat) dan "bored" (kata kerja lampau) memiliki nuansa yang berbeda dalam ulasan, namun lemmatisasi mungkin menyatukannya menjadi "bore", sehingga mengurangi daya pembeda fitur bagi model.

Trade-off Efisiensi dan Kompleksitas

Temuan ini menantang asumsi umum bahwa pemrosesan linguistik yang lebih dalam (seperti lemmatisasi) selalu menghasilkan model yang lebih baik. Dalam konteks rekayasa data untuk sistem *real-time* atau skala besar, pendekatan berbasis aturan (*rule-based*) terbukti menjadi pilihan yang lebih pragmatis karena menawarkan kecepatan eksekusi 4 kali lebih cepat dengan kinerja model yang justru lebih superior pada kasus klasifikasi biner ini.

Konsistensi dan Reproduktifitas

Meskipun hasil validasi silang (CV) tidak menunjukkan keunggulan konsistensi yang jelas untuk pipeline otomatis di semua kasus (standar deviasi CV sedikit lebih rendah untuk MNB-Otomatis tetapi sedikit lebih tinggi untuk LR-Otomatis dibandingkan manual), sifat terprogram dari pipeline otomatis secara fundamental menawarkan tingkat reproduktifitas dan konsistensi yang lebih tinggi daripada pendekatan manual. Skrip otomatis memastikan bahwa setiap data diproses dengan aturan yang identik setiap saat, menghilangkan subjektivitas dan variabilitas yang mungkin timbul dari intervensi manual.

Implikasi

Hasil ini menyoroti adanya *trade-off* antara kualitas pembersihan, kecepatan eksekusi, dan konsistensi. Meskipun cleaning manual memberikan sedikit keunggulan performa model dalam eksperimen ini dan lebih cepat dalam eksekusi, pipeline otomatis menawarkan jaminan reproduktifitas. Pilihan antara kedua pendekatan bergantung pada prioritas spesifik proyek: untuk akurasi maksimal pada dataset spesifik dengan waktu eksekusi preprocessing yang lebih cepat, pendekatan manual (atau semi-manual yang disesuaikan) mungkin lebih baik. Namun, untuk skalabilitas, konsistensi lintas proyek, dan pengurangan potensi human error dalam pipeline produksi, pendekatan otomatis tetap menjadi pilihan strategis, meskipun perlu optimasi lebih lanjut untuk meningkatkan kecepatan eksekusinya (misalnya, menggunakan pustaka lemmatisasi yang lebih cepat atau teknik paralelisasi).

5. Kesimpulan

Penelitian ini telah melakukan analisis komparatif mendalam terhadap efektivitas dua strategi *data cleaning* dalam alur kerja analisis sentimen menggunakan dataset ulasan film IMDB. Berdasarkan serangkaian eksperimen yang

membandingkan **Pipeline Berbasis Aturan (*Rule-Based*)** yang memanfaatkan regex dengan **Pipeline Berbasis Lemmatisasi** yang memanfaatkan pustaka NLTK, dapat ditarik beberapa kesimpulan utama.

Efisiensi Komputasi

Temuan paling signifikan dari penelitian ini adalah adanya kesenjangan efisiensi waktu yang drastis antara kedua pendekatan. Pipeline Berbasis Aturan terbukti jauh lebih unggul dengan waktu eksekusi rata-rata **8,87 detik**, dibandingkan dengan Pipeline Berbasis Lemmatisasi yang membutuhkan waktu **38,43 detik**. Hal ini menunjukkan bahwa penerapan normalisasi morfologi (lemmatisasi) menambah beban komputasi sekitar **333%** lebih tinggi tanpa memberikan keuntungan kecepatan pada fase *preprocessing*.

Kinerja Model Klasifikasi

Berlawanan dengan asumsi umum bahwa pemrosesan linguistik yang lebih kompleks akan selalu meningkatkan akurasi, hasil eksperimen menunjukkan bahwa pendekatan sederhana Berbasis Aturan justru menghasilkan kinerja yang lebih kompetitif.

- Model **Logistic Regression (LR)** dengan data hasil pembersihan Berbasis Aturan mencapai akurasi tertinggi sebesar **89,43%**, unggul dibandingkan data hasil Lemmatisasi (88,93%).
- Uji statistik *Paired T-Test* mengonfirmasi bahwa perbedaan akurasi ini, meskipun terlihat kecil (~0,5%), adalah **signifikan secara statistik ($p < 0.05$)**.
- Hal ini mengindikasikan bahwa untuk dataset ulasan film berbahasa Inggris yang cenderung informal, bentuk asli kata (*inflected words*) mungkin mengandung nuansa sentimen spesifik yang hilang ketika kata dikembalikan ke bentuk dasarnya melalui lemmatisasi.

Implikasi Praktis

Penelitian ini menyimpulkan bahwa dalam konteks analisis sentimen pada *Big Data* atau sistem waktu nyata (*real-time*) di mana latensi adalah faktor kritis, Pipeline Berbasis Aturan merupakan pilihan yang lebih strategis. Metode ini menawarkan keseimbangan terbaik antara kecepatan eksekusi yang tinggi dan akurasi model yang superior. Sementara itu, penggunaan lemmatisasi, meskipun menjamin konsistensi kosakata yang lebih baik, perlu dipertimbangkan ulang penggunaannya jika sumber daya komputasi terbatas atau jika dataset memiliki karakteristik informal yang kuat.

Saran Penelitian Selanjutnya

Untuk penelitian mendatang, disarankan untuk menguji kedua pipeline ini pada dataset dengan domain bahasa yang lebih formal (seperti jurnal hukum atau medis) di mana lemmatisasi mungkin memberikan dampak positif yang lebih besar. Selain itu, eksplorasi teknik *stemming* sebagai jalan tengah antara regex murni dan lemmatisasi penuh juga dapat menjadi arah penelitian yang menarik untuk menyeimbangkan efisiensi dan normalisasi kata.

6. Daftar Pustaka

[1] "Text Mining in Big Data Analytics," *MDPI Journals*, [Daring]. Tersedia pada: https://www.mdpi.com/2504-2289/4/1/1

[2] I. Idris, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Mengunakan Algoritma Support Vector Machine (SVM)," *Jambura J. Electr. Electron. Eng.*, [Daring]. Tersedia pada: https://ejurnal.ung.ac.id/index.php/jjeee/article/view/16830

[3] I. Clemence, "Day 47: Sentiment Analysis Using Python Libraries," 2025. [Daring]. Tersedia pada: https://ianclemence.medium.com/day-47-sentiment-analysis-using-python-libraries-a2447c6154d8

[4] A. Al-Sallab, "A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance," 2021.

[5] A. Upadhye, "A Comprehensive Survey of Text Data Cleaning Techniques: Challenges, Methods, and Best Practices," *J. Sci. Eng. Res.*, vol. 7, no. 8, hal. 205–210, 2020.

[6] M. A. Waskom, "The Importance of Data Cleaning in Machine Learning: Best Practices and Techniques," 2024. [Daring]. Tersedia pada: https://www.researchgate.net/publication/385173406_The_Importance_of_Data_Cleaning_in_Machine_Learning_Best_Practices_and_Techniques

[7] "A Comparative Study on Data Cleaning Approaches in Sentiment Analysis." [Daring]. Tersedia pada: https://www.researchgate.net/publication/342156562_A_Comparative_Study_on_Data_Cleaning_Approaches_in_Sentiment_Analysis

[8] M. A. Ogunlese dan et al., "An Automated Python Script for Data Cleaning and Labeling using Machine Learning Technique," *Informatica*, vol. 47, hal. 219–232, 2023.

[9] "Automated Rule-Based Data Cleaning Using NLP." [Daring]. Tersedia pada: https://www.researchgate.net/publication/365834184_Automated_Rule-Based_Data_Cleaning_Using_NLP

[10] Codefinity, "A Comprehensive Guide to Sentiment Analysis with Python." [Daring]. Tersedia pada: https://codefinity.com/blog/A-Comprehensive-Guide-to-Sentiment-Analysis-with-Python

[11] DataCamp, "NLTK Sentiment Analysis Tutorial: Text Mining & Analysis in Python." [Daring]. Tersedia pada: https://www.datacamp.com/tutorial/text-analytics-beginners-nltk

[12] S. Gupta, "Text Cleaning in Python: Effective Data Cleaning Tutorial," 2023. [Daring]. Tersedia pada:

https://docs.kanaries.net/topics/Python/text-cleaning-python
[13] Intel, “Four Data Cleaning Techniques to Improve Large Language Model (LLM) Performance,” 2024. [Daring]. Tersedia pada: https://medium.com/intel-tech/four-data-cleaning-techniques-to-improve-large-language-model-llm-performance-77bee9003625
[14] “Sentiment Analysis of Indonesian Society Toward the Launch of iPhone 16 Using Naive Bayes, Random Forest, and KNN Algorithms,” *J. Kom.*, [Daring]. Tersedia pada: https://penerbitadm.pubmedia.id/index.php/KOMITEK/article/view/2219
[15] Y. A. Rahman dan dkk., “Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Threads Instagram di Playstore Menggunakan Algoritma Naive Bayes,” *JITET (Jurnal Inform. dan Tek. Elektro Ter.*, vol. 11, no. 2, hal. 1–7, 2023.
[16] R. Rahmadani, A. Rahim, dan R. Rudiman, “Analisis Sentimen Ulasan ‘Ojol the Game’ di Google Play Store Menggunakan Algoritma Naive Bayes dan Model Ekstraksi Fitur TF-IDF untuk Meningkatkan Kualitas Game,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024, [Daring]. Tersedia pada: https://jurnal.unimed.ac.id/2012/index.php/JITET/article/view/41554