



Deteksi sentimen ulasan pengguna *e-learning* menggunakan algoritma *naive bayes* (Studi Kasus: RuangGuru, Pahamify, Merdeka Mengajar)

Sentiment detection of e-learning user reviews using the naive bayes algorithm (Case Study: RuangGuru, Pahamify, Merdeka Mengajar)

Sulistiowati, Vivine Nurcahyawati*, Muhammad Alfa Fawwaz, Erwin Sutomo, Tutut Wuriyanto

*Universitas Dinamika, Jl. Raya Kedung Baruk 98 Surabaya, Indonesia

INFORMASI ARTIKEL ABSTRAK

Article History:

Submission: 10-10-2024

Revised: 24-10-2024

Accepted: 05-11-2024

Kata Kunci:

Sentimen; ulasan; E-learning; klasifikasi; akurasi

Keywords:

Sentiment; review; E-learning, classification; accuracy

* Korespondensi:

Vivine Nurcahyawati
vivine@dinamika.ac.id

Peningkatan penggunaan aplikasi *e-learning* secara signifikan telah mengubah cara belajar mandiri. Analisis sentimen pengguna *e-learning* populer di Indonesia, yaitu Pahamify, Ruangguru, dan Merdeka Mengajar, menggunakan data ulasan dari *Google Play Store*, dilakukan dalam penelitian ini. Aplikasi *e-learning* menarik banyak pelajar karena fleksibilitasnya dalam hal waktu dan tempat. Namun, pemahaman terhadap persepsi pengguna masih terbatas. Penelitian ini berfokus pada analisis sentimen pengguna dan penerapan algoritma Naive Bayes untuk memprediksi sentimen tersebut. Model ini dievaluasi menggunakan *confusion matrix*, dengan hasil akurasi 89% untuk Ruangguru, 86% untuk Pahamify, dan 92% untuk Merdeka Mengajar. Temuan ini memberikan kontribusi berupa: (1) wawasan yang berguna bagi pengembang aplikasi untuk meningkatkan layanan, (2) bukti bahwa algoritma *Naive Bayes* efektif dalam menganalisis sentimen teks berbahasa Indonesia, dan (3) dasar bagi penelitian lebih lanjut, misalnya dengan memperluas cakupan aplikasi atau menggunakan algoritma yang lebih kompleks. Mayoritas pengguna menunjukkan sentimen positif terhadap ketiga aplikasi, meskipun masih ada aspek yang perlu diperbaiki, yang bisa diidentifikasi melalui analisis lebih lanjut.

ABSTRACT

The significant increase in the use of e-learning applications has transformed the way of self-study. This research conducts a sentiment analysis of popular e-learning platforms in Indonesia, namely Pahamify, Ruangguru, and Merdeka Mengajar, utilizing review data from the Google Play Store. E-learning applications attract many students due to their flexibility in terms of time and place. However, the understanding of user perceptions remains limited. This study focuses on user sentiment analysis and the application of the Naive Bayes algorithm to predict that sentiment. The model is evaluated using a confusion matrix, with accuracy results of 89% for Ruangguru, 86% for Pahamify, and 92% for Merdeka Mengajar. These findings contribute in several ways: (1) providing valuable insights for application developers to enhance their services, (2) demonstrating the effectiveness of the Naive Bayes algorithm in analyzing sentiment in Indonesian text, and (3) establishing a basis for further research, such as expanding the scope of applications analyzed or using more complex algorithms. The majority of users express positive sentiment towards the three applications, although there are still aspects that require improvement, which can be identified through further analysis.



1. PENDAHULUAN

Banyak aplikasi saat ini mencerminkan pesatnya kemajuan internet dan teknologi. Menurut Asosiasi Penyelenggara Jasa Internet Indonesia, sekitar 196,7 juta orang di Indonesia diperkirakan akan menggunakan internet hingga akhir tahun 2020 [1]. *Google Play Store* memungkinkan pengguna untuk dengan cepat mengakses dan mengunduh aplikasi di perangkat *mobile* berbasis Android. Secara global, jumlah unduhan aplikasi meningkat sebesar 8,7% dibandingkan kuartal pertama tahun 2020, dengan *Google Play Store* memimpin sebagai *platform* terbesar dengan 3,8 miliar unduhan, menurut databoks.www.katadata.co.id [2].

Salah satu aplikasi yang tersedia di *Google Playstore* adalah aplikasi untuk mendukung pendidikan dan dapat disebut sebagai *e-learning* [3]. Aplikasi *e-learning* meningkatkan jumlah orang untuk dapat belajar mandiri secara *online* [4]. Saat ini, proses pembelajaran di institusi pendidikan terpusat pada guru. Siswa tidak terlibat secara langsung dalam pelajaran, dan penggunaan media pembelajaran yang buruk dianggap tidak efektif dalam mencapai tujuan pembelajaran karena semua kebutuhan materi harus dipenuhi oleh siswa. Dengan demikian, kebebasan dalam mengatur waktu dan lokasi menjadi salah satu alasan utama yang memotivasi pengguna untuk memilih belajar mandiri melalui *platform online* dengan menggunakan aplikasi *e-learning*. Hal ini memberi pengguna kemampuan untuk belajar kapanpun dan dimanapun mereka inginkan.

Menurut my-best.id, sepuluh aplikasi pendidikan *online* yang disarankan adalah *zenius*, *pahamify*, *kelasKita*, *ruangguru*, *udemy*, aplikasi belajar TK-PAUD, *duolingo*, *checkmate*, *merdeka mengajar*, dan *belajar Al-quran*. tiga aplikasi yang dianalisis dalam penelitian ini adalah *ruangguru*, *pahamify*, dan *merdeka mengajar*, karena berada di posisi teratas di situs *online* yang dilansir, telah memiliki ruang edar di Indonesia, dan memiliki banyak sekali ulasan. Berbagai tinjauan positif dan negatif untuk ketiga aplikasi yang diteliti diberikan, menunjukkan bahwa pengguna mungkin memiliki persepsi yang lebih baik tentang mereka dibandingkan dengan aplikasi sebelumnya. Lihat situasi dan pendapat pengguna untuk mengetahui pendapat pengguna. Karena format data opini dan *review* pengguna belum terstruktur, pengambilan data yang mengolah teks dari *review* komentar diperlukan untuk mengetahuinya. Karena banyak ulasan pengguna yang belum puas dengan kinerja atau pelayanan yang ditawarkan oleh ketiga aplikasi tersebut, maka aplikasi tersebut seharusnya berusaha memenuhi ekspektasi pengguna dan meningkatkan segala layanannya untuk tetap bersaing dengan aplikasi lainnya.

Dalam sistem ketiga aplikasi, ulasan pengguna berperan sangat penting untuk kelangsungan bisnis, karena keberlangsungan perusahaan jasa sangatlah bergantung pada kepuasan pengguna. Oleh karena itu, mempertahankan pengguna setia adalah salah satu cara untuk meningkatkan reputasi aplikasi [5]. Oleh karena itu, analisis sentimen perlu dilakukan pada ulasan pengguna di sosial media untuk menyaring opini dan mengklasifikasikannya. Hasil kualifikasi ini dapat digunakan untuk menganalisis pengguna aplikasi *e-learning*, tergantung pada aplikasi yang digunakan.

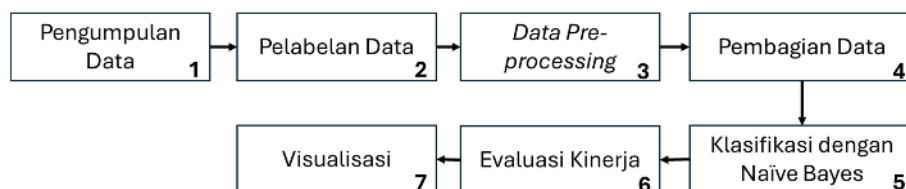
Sentiment analysis adalah proses otomatis untuk memahami, mengekstrak, dan mengolah data teks publik terkait suatu topik, opini, evaluasi, atau objek, yang diambil dari media sosial atau situs web, dengan tujuan memperoleh informasi penting dari data tersebut [6][7][8]. Pendapat publik dikategorikan ke dalam kategori positif dan negatif melalui analisis sentimen, sehingga hasil analisis tersebut dapat digunakan untuk membantu dalam memahami bagaimana pengguna berinteraksi dengan aplikasi *e-learning* berdasarkan pengalaman mereka menggunakannya [9].

Pada analisis sentimen, metode *Naive Bayes* diterapkan untuk mengelompokkan data dalam bentuk teks. Setelah itu, teks yang telah dikelompokkan digunakan untuk mengklasifikasikan kata-kata sesuai dengan kategori opini yang telah ditetapkan sebelumnya [10]. *Naive Bayes* berfungsi sebagai teknologi *pre-processing* dalam kualifikasi fitur, yang bertujuan untuk meningkatkan akurasi, efisiensi, serta skalabilitas proses klasifikasi teks [11].

Studi ini menggunakan ulasan pengguna untuk mengetahui *review* pengguna aplikasi *e-learning* terbaik yang direkomendasikan peneliti saat ini. Hasil analisis dapat digunakan untuk menyusun rencana strategis untuk senantiasa meningkatkan kualitas layanan atau performa aplikasi agar pengguna menikmatinya. Studi ini mengidentifikasi perasaan pengguna aplikasi *e-learning*, seperti yang disebutkan di atas, dengan menggunakan algoritma *Naive Bayes*. Aplikasi yang akan dipelajari adalah ruang guru, memahami, dan merdeka mengajar. *Python* dan alat *Google Colab* digunakan dalam penelitian ini.

2. METODE

Dalam menyelesaikan penelitian ini, beberapa tahapan dilakukan secara sistematis untuk memastikan hasil yang akurat dan dapat dipertanggungjawabkan. Pertama, data yang relevan dikumpulkan dari berbagai sumber untuk mendapatkan informasi yang memadai. Setelah itu, proses pelabelan dilakukan untuk mengkategorikan data sesuai dengan tujuan penelitian. Selanjutnya, data yang telah dilabeli melewati tahap *pre-processing* untuk membersihkan dan mempersiapkan data agar siap digunakan dalam analisis. Tahap ini mencakup normalisasi, penghilangan duplikasi, serta transformasi data ke format yang lebih sesuai untuk model yang akan digunakan. Data kemudian dibagi menjadi dua subset, yakni data pelatihan dan data pengujian, untuk menghindari bias dalam evaluasi model. Proses klasifikasi dilakukan menggunakan algoritma *Naive Bayes*, yang dipilih karena kemampuannya yang efisien dalam mengelola *database* besar dan heterogen. Setelah klasifikasi selesai, kinerja model dievaluasi menggunakan metrik seperti akurasi, presisi, *recall*, dan *F1-score*. Metrik-metrik ini memberikan gambaran yang komprehensif mengenai seberapa baik model dalam mengklasifikasikan data. Akhirnya, hasil dari penelitian ini divisualisasikan dalam bentuk grafik dan tabel untuk memudahkan interpretasi dan analisis lebih lanjut. Visualisasi tersebut membantu dalam memahami pola serta anomali yang mungkin muncul dari hasil klasifikasi. Tahapan-tahapan penelitian ini dapat dilihat lebih detail pada [Gambar 1](#) yang menunjukkan alur proses dari awal hingga akhir penelitian.



Gambar 1. Tahapan penelitian.

Pengumpulan data

Data yang dianalisis dalam penelitian ini diperoleh dari ulasan pengguna pada kolom komentar tiga aplikasi yang menjadi objek penelitian di *Google Play Store*. Proses ini melibatkan pencarian data relevan berdasarkan kata kunci, serta pemahaman lebih mendalam tentang data melalui teknik data mining menggunakan *Python* dan *Google Colab*. Selanjutnya, dilakukan *import library Pandas*, yang fungsinya untuk memanipulasi data. Kemudian, *library Google Play scraper* digunakan untuk melakukan *crawling* data dari *Play Store*, yang kemudian dikonversi ke dalam format ".csv" [12]. Hasil disimpan dalam format.csv, yang memiliki nilai yang dipisahkan dengan koma. Data yang diperoleh dari *crawling* mencapai 2000 data untuk setiap objek studi kasus. Oleh karena itu, data ulasan dicatat dan dikategorikan menggunakan atribut berikut: *reviewId*, *userImage*, *userName*, *content*, *score*, *reviewCreated*, *thumbsUp*, *at*, *replyContent*, *replyAt*, *soirtOrder*, and *appId*.

Pelabelan data

Penentuan label data, atau yang dikenal sebagai klasifikasi atribut, dilakukan berdasarkan penilaian subjektif peneliti. [Tabel 1](#) memberikan contoh bagaimana proses pelabelan data dilakukan untuk mencapai tujuan tersebut. Data ini diambil dari kolom komentar pada setiap

aplikasi di *Google Play Store* dan kemudian dikategorikan. Dengan menggunakan *Google Colab*, ulasan positif dan negatif dari aplikasi *e-learning* dikumpulkan secara otomatis sesuai dengan kategorinya.

Tabel 1. Identifikasi masalah saat ini (tahap pelabelan).

Content	Kategori
Mengharapkan perbaikan segera dari aplikasi <i>force close</i> HP saya, mohon bantuannya, developer $\delta\checkmark^{\text{TM}} \delta\checkmark \gg$	Negatif
Aplikasinya sudah bagus. Saya hanya ingin tahu apakah bisa dibuat <i>full screen</i> di laptop atau komputer supaya lebih jelas, dan jika bisa, masukkan <i>dark mode</i> .	Positif

Data Preprocessing

Tahapan *pre-processing* data mencakup kegiatan ekstraksi, pembersihan, pemilihan, dan pengolahan data untuk keperluan analisis lanjutan. Tahapan ini bertujuan untuk mengurangi "sampah" dalam data teks, mengubah bentuk datanya yang tidak terstruktur menjadi bentuk yang lebih terstruktur hingga menjadi kata dasarnya. Dalam penelitian ini, seleksi data yang akan digunakan telah diselesaikan. Tahap ini melibatkan penggunaan berbagai *library* di *Google Colab* yang ditulis dalam *Python*. Proses ini mencakup pembersihan data, pemotongan bagian yang tidak relevan, penghapusan kata kunci, tokenisasi, serta penyamaran data [13].

Dalam tahap *Cleansing*, ulasan yang telah dikumpulkan diubah menjadi format teks yang sesuai dengan kebutuhan. Langkah ini mencakup penghapusan angka, tanda baca, dan *emoticon*. Selanjutnya, pada tahap *Case Folding*, semua huruf besar diubah menjadi huruf kecil. *Stopword Removal* diterapkan untuk menghilangkan kata-kata umum yang dalam keberadaannya pada kalimat tidak memiliki makna signifikan dalam konteks kalimat atau dokumen, sehingga mengurangi jumlah kata yang disimpan dalam corpus atau sistem. Proses berikutnya adalah *Tokenizing*, yang memisahkan setiap kata dalam kalimat dengan tanda koma (,). *Stemming* kemudian dilakukan untuk mengembalikan kata-kata ke bentuk dasarnya berdasarkan aturan Bahasa Indonesia. Hasil dari penerapan proses *pre-processing* ini ditunjukkan dalam Tabel 2.

Tabel 2. Hasil data *pre-processing*.

Data mentah	Puas dilayani dengan ramah dan responnya cepat, mousenya bermutu, nanti pesan lagi.
<i>Cleansing</i>	Puas dilayani dengan ramah dan responnya cepat mousenya bermutu nanti pesan lagi
<i>Case Folding</i>	Puas dilayani dengan ramah dan responnya cepat mousenya bermutu nanti pesan lagi
<i>Stopword Removal</i>	Puas dilayani ramah responnya cepat mousenya bermutu nanti pesan lagi
<i>Tokenizing</i>	'puas' 'dilayani' 'ramah' 'responnya' 'cepat' 'mousenya' 'bermutu' 'nanti' 'pesan' 'lagi'
<i>Stemming</i>	Puas layan ramah respon cepat mouse mutu nanti pesan lagi

Pembagian data

Pada fase penelitian ini, data kemudian dibagi menjadi dua bagian: data untuk pelatihan dan data untuk pengujian [14]. Data uji digunakan untuk mengevaluasi kinerja algoritma *Naive Bayes* setelah proses pelatihan dilakukan. Dalam hal ini, terdapat *library* dengan nama *Python Sklearn* dalam *Google Colab* untuk menjalankan fungsi *train_test_split*. Sebelum digunakan sebagai *input* dalam metode *Naive Bayes*, data perlu diberikan label dan melalui tahap *pre-processing*.

Klasifikasi dengan Naive Bayes

Rumus naive bayes yang diubah digunakan untuk memenuhi kelas data yang akan diklasifikasikan sesuai persamaan 1.

$$P(Wk)|Ci = \frac{nk+1}{n+|Vocabulary|} \quad (1)$$

Dimana:

- P(Wk) : Probabilitas (peluang) kemunculan kata
- Ci : Kategori kelas
- P(Wk|Ci) : Probabilitas bahwa kata (Wk) muncul dalam dokumen yang termasuk dalam kategori kelas (Ci)
- Nk : Nilai frekuensi kata dalam kategori Ci
- N : Total jumlah dari kata dalam kategori Ci
- Vocabulary : Total jumlah dari kata

Evaluasi kinerja

Pengujian akurasi ini melibatkan empat kondisi: *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) [15]. *Confusion matrix* menjadi alat ukur untuk nilai akurasi, presisi, dan *recall*. Di bawah ini adalah rumus perhitungan detailnya:

Akurasi model dalam mengklasifikasikan dengan benar ditunjukkan oleh perhitungan berikut.:

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

Dari perhitungan menunjukkan akurasi berdasarkan hasil prediksi model dan data permintaan:

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

Kesuksesan model dalam menemukan kembali data digambarkan dengan perhitungan.

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

Visualisasi

Tahap visualisasi data menyajikan hasil akhir dalam bentuk *word cloud* dengan tujuan agar hasil analisis dapat memudahkan pemahaman peneliti dan pembaca. Dengan memanfaatkan *word cloud*, data yang divisualisasikan memberikan representasi berbentuk visual berdasarkan kata-kata yang sering muncul dari teks yang telah dianalisis [16]. Ukuran setiap kata dalam *word cloud* berhubungan langsung dengan frekuensi kemunculannya dalam kalimat atau teks. Proses visualisasi *word cloud* ini dilakukan menggunakan *library* dengan nama *word cloud* di *Python*.

3. HASIL DAN PEMBAHASAN

Pengumpulan data

Pengumpulan data dilakukan secara manual dengan menggunakan *Python* dan *Google Colab*. Proses ini melibatkan instalasi *library Python* bernama "*google-play-scraper*" yang berfungsi untuk mengekstrak informasi mengenai aplikasi, ulasan, peringkat, dan lainnya dari *Google Play Store*. Setelah itu, modul "aplikasi" di *import* dari *library* dengan nama "*google_play_scraper*" diikuti dengan *import library* *Pandas* untuk manipulasi data dan *Numpy* untuk perhitungan numerik. Selanjutnya, data ulasan diurutkan untuk mengumpulkan informasi dari *Google Play Store*. Pertama-tama, informasi ulasan dikumpulkan dari URL aplikasi yang tersedia di *Google Play Store*. Parameter "lang" ditetapkan untuk bahasa Indonesia, "country" menandakan data berasal dari pengguna di Indonesia, "sort" digunakan untuk mengurutkan data yang paling relevan, dan "count" mengumpulkan total 2000 data. Selain itu, "filter score" menunjukkan peringkat dari ulasan yang diambil. Hasil dari proses pengambilan data dan konversi ulasan menghasilkan beberapa kolom, yang kemudian dipilih untuk tahap selanjutnya. Kolom-kolom ini mencakup *userName*, skor, tanggal, dan konten. [Tabel 3](#) menyajikan hasil pengorganisasian data tersebut.

Tabel 3. Hasil pemilihan kolom.

	User Name	Score	AT	Content
0	Bilqia Amiroh T.A	4	2023-12-05 12:52:17	Kadang terjadi banyak sekali lah, yang disuruh...
1	Shafinna Az Zahra	4	2023-12-02 13:12:43	Aplikasinya suka nge-lag atau nge-hang seperti...
2	Ken Ahmas	2	2024-01-08 15:38:32	Rapalt ya, sebelumnya aku udah kasih 5 bintang...
3	Aqila Azka	5	2024-01-10 13:49:14	Aplikasinya sangat berguna bagi para pelajar u...
4	Alifiyan Fadlurrahman	3	2024-01-09 06:07:21	tambahkan fitur tap tahan lama di layar untuk...

Dibuat data *frame* baru bernama "my_df", yang terdiri dari kolom "sorted_df". Kemudian, "my_df" hanya akan digunakan dua kolom, yaitu kolom "isi" dan "nilai." Tabel 4 menunjukkan hasil proses. Berdasarkan cuplikan data pada Tabel 4, terlihat adanya beragam sentimen pengguna terhadap produk atau layanan yang sedang dinilai. Ada ulasan positif yang memuji kualitas produk (misalnya, "enak games pet yang dulu dibandingkan sekarang"), namun juga ada ulasan negatif yang mengeluhkan masalah teknis (misalnya, "Aplikasi tiba-tiba *crash*/tidak bisa masuk ke apk") atau pelayanan (misalnya, "kenapa yah baru baru ini pengiriman j&t super..."). Kolom *content* yang selanjutnya akan dilakukan analisis sentimen.

Tabel 4. Hasil proses pemisahan kolom.

	Content	Score
11	Sudah berkali lagi belanja disini, pesanan sel...	1
387	Enakan games per yang dulu dibandingkan sekarang.	4
451	Aplikasi tiba2 <i>crach</i> /tidak bisa masuk ke apk t...	5
88	Seneng makenya. Dan memudahkan. Tapi keamanan...	1
553	kenapa yah baru baru ini pengiriman j&t super...	2

Pelabelan data

Tools Google Colab digunakan untuk melabelkan data secara otomatis dengan menggunakan perintah *if* dan *elif* dengan nilai skor yang telah ditentukan sebelumnya. Ada "pelabelan (*score*)" yang digunakan untuk melabelkan nilai "*score*". Nilai "*score*" yang kurang dari tiga dianggap negatif, sedangkan nilai "*score*" yang sama dengan 4 dan 5 dianggap positif. Fungsi pelabelan digunakan pada kolom "*score*" dan hasilnya akan disimpan pada kolom "label". Hasil proses pelabelan dapat data ditunjukkan pada Tabel 5.

Tabel 5. Pelabelan data.

	Content	Score	Label
11	Sudah berkali lagi belanja disini, pesanan sel...	1	Negatif
387	Enakan games per yang dulu dibandingkan sekarang.	4	Positif
451	Aplikasi tiba2 <i>crach</i> /tidak bisa masuk ke apk t...	5	Positif
88	Seneng makenya. Dan memudahkan. Tapi keamanan...	1	Negatif
553	kenapa yah baru baru ini pengiriman j&t super...	2	Negatif

Data preprocessing

Setelah data disimpan, langkah berikutnya adalah membaca dari *file* CSV. Proses ini memanfaatkan fungsi "*read_csv*" untuk membaca *file* tersebut dan mengkonversinya menjadi data *frame*, yang kemudian disimpan dalam variabel "df." Setelah itu, beberapa tahap dalam proses *preprocessing* teks akan diterapkan. Proses *cleansing* dilakukan dengan tujuan untuk menghapus angka, tanda baca, dan URL menggunakan ekspresi reguler. Pada langkah selanjutnya, tahap *Case Folding* mengubah setiap huruf kapital dalam ulasan menjadi huruf kecil dengan menggunakan fungsi *Lower* (). Kata-kata dalam ulasan akan dibandingkan dengan daftar

stopwords yang ada dalam *library corpus/stopwords*. Untuk menyaring atau membersihkan data, *library NLTK* diimpor, yang memungkinkan fokus pada kata-kata yang memiliki makna penting sambil menghapus kata-kata yang tidak diperlukan, seperti preposisi dan konjungsi. Untuk memudahkan pemrosesan dan analisis teks, tahap *tokenizing* memisahkan kalimat menjadi kata atau token berdasarkan spasi. Proses pemisahan kata menjadi token ini dapat dilakukan dengan fungsi dari *library*. Langkah *stemming* dilakukan dengan mengubah kata-kata ke bentuk dasarnya menggunakan *library Sastrawi*, yang merupakan *library stemming* untuk *Python* yang dikhususkan bagi bahasa Indonesia. **Tabel 2** juga menampilkan perubahan bentuk data tersebut.

Pembagian data

Setelah proses *pre-processing* teks selesai, data akan dibagi menjadi dua bagian, yaitu data latih dan data uji, dengan rasio 80:20 dari total data. Data uji bertujuan untuk mengevaluasi kinerja model atau algoritma yang diterapkan, sementara data latih digunakan untuk melatih model atau algoritma tersebut. Pembagian data ini dilakukan menggunakan fungsi yang tersedia dalam *library*. Pembagian data latih dan uji tersebut dilakukan oleh fungsi *train_test_split* dalam *library Sklearn (scikit-learn)*. Fungsi tersebut akan menghasilkan 4 kelompok data yaitu 2 kelompok data *training* (1 atribut dan 1 label) dan 2 kelompok data *testing* (1 atribut dan 1 label). Data tersebut akan digunakan untuk melatih ketangguhan model dalam melakukan analisis sentimen. Nilai *test_size* ditentukan 20% yang menandakan jumlah dari data *testing* yang digunakan, dengan mode acak atau *random_state*. Nilai acak yang digunakan adalah 0.

Klasifikasi dengan *Naive Bayes*

Dengan menggunakan algoritma *Naive Bayes*, proses klasifikasi ini dibagi menjadi dua tahap: pelatihan dan uji. Pertama, proses pelatihan data digunakan untuk mengajarkan algoritma untuk mengenali dan mempelajari data. Setelah itu, prediksi dibuat pada data pengujian, dan pengujian dilakukan dengan mempertimbangkan pelatihan yang mungkin terjadi pada dataset. Untuk melakukan klasifikasi *Naive Bayes*, *library Sklearn (scikit-learn)* mengimport *MultinomialNB* dari modul *naive_bayes*. Model klasifikasi objek "nb" akan dilatih menggunakan data latihan, dan model "*MultinomialNB()*" akan mempelajari pola dari data latihan. Pola dalam data latih dipelajari dengan menggunakan fungsi *fit()*. Fungsi *fit* menerima atribut data latihan *tfidf_train* dan label data latihan *y_train*. Label data uji *tfidf_test* diprediksi dengan menggunakan *predict()* dari model "nb". Pada langkah selanjutnya, hasil prediksi akan disimpan dalam variabel *y_pred*, yang akan digunakan untuk melakukan evaluasi kinerja model.

Prinsip perhitungan dalam *Naive Bayes* adalah menggunakan probabilitas. Nilai probabilitas per kata seperti terlihat pada **Tabel 6**. Setelah melalui tahap data *pre-processing* maka kata yang tersisa akan dihitung nilai probabilitasnya untuk menentukan kecenderungan masuk kedalam kelas positif atau negatif seperti pada perhitungan dibawah. Contoh kata yang digunakan adalah "layanan sistem bosan kecewa lancar".

Tabel 6. Contoh perhitungan nilai probabilitas tiap kata.

No	Kata	$N_k Negatif$	$N_k Positif$	$P(W_k C_i)$ negatif	$P(W_k C_i)$ positif
1	kecewa	1	0	0.029	0.012
2	sistem	3	3	0.059	0.051
2	lancar	1	1	0.029	0.025
3	layanan	1	0	0.029	0.012
4	bosan	0	2	0.014	0.038

Nilai Positif:

$$= P(\text{pos}) \times P(\text{layanan}|\text{pos}) \times P(\text{sistem}|\text{pos}) \times P(\text{bosan}|\text{pos}) \times P(\text{kecewa}|\text{pos}) \times P(\text{lancar}|\text{pos}).$$

$$= 0.6 * 0.012 * 0.051 * 0.038 * 0.012 * 0.025$$

$$= 0.000000004186$$

Nilai Negatif:

$$= P(\text{neg}) \times P(\text{layanan}|\text{neg}) \times P(\text{sistem}|\text{neg}) \times P(\text{bosan}|\text{neg}) \times P(\text{kecewa}|\text{neg}) \times P(\text{lancar}|\text{neg}).$$

$$= 0.4 * 0.029 * 0.059 * 0.014 * 0.029 * 0.029$$

$$= 0.000000008058$$

Proses klasifikasi sentimen di atas didasarkan pada hasil nilai probabilitas, yang memiliki nilai tertinggi, yaitu 0.000000008058 sehingga ulasan “layanan sistem bosan kecewa lancar” tersebut diklasifikasikan kedalam kelas ulasan “Negatif”.

Evaluasi kinerja

Hasil dari *Confusion Matrix* berfungsi sebagai indikator untuk mengevaluasi performa atau kinerja model, yang mencakup *matrix accuracy*, *precision*, dan *recall*. Hasil *Confusion Matrix* terpaparkan pada [Tabel 7](#).

Tabel 7. Hasil confusion matrix.

Sumber Data	Accuracy	Precision	Recall	F1-Score
Ruang Guru	0.89	0.62	0.53	0.57
Pahamify	0.86	0.70	0.65	0.68
Merdeka Mengajar	0.92	0.66	0.4	0.5

Hasil untuk Ruangguru, model *Naive Bayes* menghasilkan akurasi 0.89, yang cukup tinggi. Namun, nilai *precision* sebesar 0.62 menunjukkan bahwa model cenderung menghasilkan prediksi positif yang tidak selalu akurat. *Recall* sebesar 0.53 menunjukkan bahwa hanya 53% dari sentimen positif sebenarnya yang terprediksi dengan benar oleh model. *F1-Score* sebesar 0.57, yang merupakan keseimbangan antara *precision* dan *recall*, menunjukkan bahwa meskipun model cukup akurat, kinerja keseluruhannya masih bisa ditingkatkan terutama dalam menangkap lebih banyak data positif secara tepat. Sedangkan pada *Pahamify*, meskipun akurasinya sedikit lebih rendah dibandingkan Ruangguru (0.86), *Pahamify* memiliki *precision* yang lebih tinggi, yaitu 0.70, yang berarti model lebih tepat dalam memprediksi sentimen positif. *Recall* sebesar 0.65 menunjukkan bahwa model telah cukup baik dalam mengidentifikasi sentimen positif yang benar, lebih tinggi dari Ruangguru. *F1-Score* sebesar 0.68 menunjukkan bahwa secara keseluruhan, model ini lebih seimbang dalam memprediksi sentimen dibandingkan dengan Ruangguru. Hasil berikutnya adalah *Merdeka Mengajar*. Aplikasi ini memiliki akurasi tertinggi (0.92), tetapi nilai *precision* (0.66) dan terutama *recall* (0.4) lebih rendah dibandingkan aplikasi lain. Hal ini menunjukkan bahwa meskipun model sangat akurat dalam prediksi keseluruhan, hanya 40% dari sentimen positif yang benar-benar teridentifikasi, yang menyebabkan *F1-Score* juga rendah (0.5). Ini menunjukkan bahwa meskipun akurasi tinggi, model kurang optimal dalam mendeteksi semua sentimen positif yang ada di data. Pada hasil penelitian ini didapatkan bahwa penggunaan kata tidak baku dalam penulisan *review* menyebabkan berkurangnya nilai akurasi. Sehingga sebagai peluang penelitian kedepan bahwa penggunaan dan penambahan *library* bahasa gaul atau kata-kata *slang* memiliki potensi untuk meningkatkan nilai akurasi model, membuka peluang pengembangan lebih lanjut untuk analisis sentimen yang lebih akurat pada aplikasi *e-learning*.

Visualisasi

Pada tahap akhir visualisasi ini, hasil analisis ditampilkan melalui visualisasi di *word cloud*. Tujuannya adalah untuk menggambarkan kata-kata yang sering muncul di setiap kelas data yang dianalisis. Banyak kata yang sering muncul dalam data yang dianalisis digambarkan oleh berbagai ukuran font. Hasil visualisasi dengan *word cloud* ditunjukkan pada [Gambar 2](#).



Gambar 2. Hasil visualisasi dengan *word cloud*.

4. SIMPULAN

Berdasarkan hasil penelitian, aplikasi *e-learning* Ruangguru, *Pahamify*, dan Merdeka Mengajar menunjukkan tingkat akurasi yang berbeda dalam prediksi sentimen menggunakan model *Naive Bayes*. Ruangguru memiliki akurasi sebesar 89% dengan kata-kata paling sering ditemukan seperti "ruangguru", "ruang", "guru", dan "ajar", menunjukkan fokus pada branding dan materi pengajaran. *Pahamify* mencapai akurasi sebesar 86%, dengan kata-kata umum seperti "ajar", "aplikasi", "bagus", dan "pahamify", yang menggambarkan kualitas pengajaran serta kepuasan pengguna terhadap aplikasi. Sementara itu, Merdeka Mengajar memperoleh akurasi tertinggi, yaitu 92%, dengan kata-kata yang sering muncul seperti "bantu", "aplikasi", "guru", dan "ajar", yang menekankan pada peran aplikasi dalam membantu guru mengajar.

REFERENSI

- [1] S. Hamza, "Pelatihan Penggunaan Web Site Desa Kecamatan Sahu Timur Kabupaten Halmahera Barat," *J. Biosainstek*, vol. 4, no. 2, pp. 61–66, 2022, doi: 10.52046/biosainstek.v4i2.1158.
- [2] S. Annisa, R. Manaf, A. Fitrianto, A. Mohamad, and S. #3, "Perbandingan Algoritma Pohon dengan Beberapa Skenario Pelabelan untuk Analisis Sentimen pada Aplikasi Milik Pemerintah/BUMN," *JEPIN (Jurnal Edukasi dan Penelit. Inform.)*, vol. 10, no. 1, pp. 24–32, 2024, doi: <http://dx.doi.org/10.26418/jp.v10i1.73512>.
- [3] D. Yulistiyanti, N. Farkhatin, and D. Mustari, "Penggunaan Aplikasi Sebagai Media E-Learning Remaja Di Karang Taruna," *J. Empower.*, vol. 2, no. 1, p. 89, 2021, doi: 10.35194/je.v2i1.1265.
- [4] A. Isroqmi, R. Rohana, and E. Septiati, "Pemanfaatan E-learning Moodle Sebagai Laboratorium Matematika Virtual di Universitas PGRI Palembang," *Indiktika J. Inov. Pendidik. Mat.*, vol. 5, no. 2, pp. 244–254, 2023, doi: 10.31851/indiktika.v5i2.11653.
- [5] S. Sundari, A. Ahmadi, and F. Wulansari, "Strategi Mempertahankan Loyalitas Konsumen Pada Usaha Mr Laundry Di Kota Pontianak," *ARMADA J. Penelit. Multidisiplin*, vol. 1, no. 9, pp. 1117–1132, 2023, doi: 10.55681/armada.v1i9.880.
- [6] L. Ardiani, H. Sujaini, and T. Tursina, "Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 2, p. 183, 2020, doi: 10.26418/justin.v8i2.36776.
- [7] D. Musfiroh, U. Khaira, P. E. P. Utomo, and T. Suratno, "Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 1, no. 1, pp. 24–33, 2021, doi: 10.57152/malcom.v1i1.20.
- [8] Sriani, A. H. Lubis, and L. P. A. Lubis, "Sentiment analysis on twitter about the death penalty using the support vector machine method," *TEKNOSAINS J. Sains, Teknol. dan Inform.*, vol. 11, no. 2, pp. 312–321, 2024, doi: 10.37373/tekno.v11i2.1096.
- [9] Tania Puspa Rahayu Sanjaya, Ahmad Fauzi, and Anis Fitri Nur Masruriyah, "Analisis sentimen ulasan pada e-commerce shopee menggunakan algoritma naive bayes dan support vector machine," *INFOTECH J. Inform. Teknol.*, vol. 4, no. 1, pp. 16–26, 2023, doi: 10.37373/infotech.v4i1.422.
- [10] D. Darwis, N. Siskawati, and Z. Abidin, "Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional," *J. Tekno Kompak*, vol. 15, no. 1, p. 131, 2021, doi: 10.33365/jtk.v15i1.744.
- [11] M. E. Purbaya, D. P. Rakhmadani, Maliana Puspa Arum, and Luthfi Zian Nasifah, "Implementation of n-gram Methodology to Analyze Sentiment Reviews for Indonesian Chips Purchases in Shopee E-Marketplace," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 3, pp. 609–617, 2023, doi: 10.29207/resti.v7i3.4726.
- [12] F. Pirmansyah and T. Wahyudi, "Implementasi Data Mining Menggunakan Algoritma C4.5 Untuk Prediksi Evaluasi Anggota Satuan Pengamanan Studi Kasus Pt. Yimm Pulogadung," *J. Indones. Manaj. Inform. dan Komun.*, vol. 4, no. 3, pp. 1566–1580, 2023, doi:

10.35870/jimik.v4i3.370.

- [13] W. Latuny, V. O. Lawalata, D. B. Pailin, and R. Ohoirenan, "Sentiment Analysis of Consumers for Determining the Packaging Features of Eucalyptus Oil Products," *J. Ilm. Tek. Ind.*, vol. 20, no. 1, pp. 71–80, 2021, doi: 10.23917/jiti.v20i1.13461.
- [14] H. Hayati and M. R. Alifi, "Analisis Sentimen Pada Tweet Terkait Vaksin Covid-19 Menggunakan Metode Support Vector Machine," *JTT (Jurnal Teknol. Ter.)*, vol. 7, no. 2, p. 110, 2021, doi: 10.31884/jtt.v7i2.349.
- [15] I. Maulana, N. Rahaningsih, and T. Suprapti, "Analisis Penggunaan Model Yolov8 (You Only Look Once) Terhadap Deteksi Citra Senjata Berbahaya," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 6, pp. 3621–3627, 2024, doi: 10.36040/jati.v7i6.8271.
- [16] J. A. Wibowo, V. C. Mawardi, and T. Sutrisno, "Visualisasi Word Cloud Hasil Analisis Sentimen Berbasis Fitur Layanan Aplikasi Gojek Dengan Support Vector Machine," *J. Serina Sains, Tek. dan Kedokt.*, vol. 2, no. 1, pp. 61–70, 2024, doi: 10.24912/jsstk.v2i1.32058.