Perbandingan SVM dan XGBoost Menggunakan SMOTE pada Aplikasi KA Bandara PT. Railink

Aulia Syalsabila Dian Yahya[1], Aries Dwi Indriyanti[2]

[1,2] Sistem Informasi, Universitas Negeri Surabaya
[1]aulia.21070@mhs.unesa.ac.id
[2]ariesdwi@unesa.ac.id

***Abstrak*— Perkembangan teknologi digital telah mendorong perusahaan transportasi seperti PT. Railink untuk menyediakan layanan berbasis aplikasi guna mempermudah pengguna dalam memesan tiket Kereta Api Bandara. Namun, berdasarkan ulasan pengguna di Google Play Store, aplikasi KA Bandara masih menghadapi berbagai keluhan, terutama terkait sistem pembayaran dan stabilitas aplikasi. Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna terhadap aplikasi KA Bandara menggunakan dua metode klasifikasi, yaitu Support Vector Machine (SVM) dan Extreme Gradient Boosting (XGBoost), serta membandingkan kinerjanya berdasarkan nilai akurasi, recall, precision, dan F1-score. Proses klasifikasi didahului dengan tahapan preprocessing teks, pembobotan menggunakan TF-IDF, dan penanganan data tidak seimbang menggunakan teknik SMOTE. Penelitian ini juga menggunakan dua pendekatan pelabelan data, yaitu manual dan lexicon-based. Hasil penelitian menunjukkan bahwa metode XGBoost memiliki performa yang paling unggul dengan nilai accuracy sebesar 89%. Penggunaan pelabelan lexicon secara signifikan meningkatkan akurasi kedua metode. Temuan ini diharapkan dapat menjadi acuan dalam mengevaluasi dan meningkatkan kualitas aplikasi KA Bandara agar lebih optimal dalam memenuhi kebutuhan pengguna.**

***Kata Kunci*— Sentimen, Aplikasi KA Bandara, SVM, XGBoost, SMOTE**

I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah membawa dampak signifikan terhadap berbagai sektor, termasuk transportasi. Transportasi publik menjadi salah satu aspek penting dalam mendukung mobilitas, aksesibilitas, dan produktivitas masyarakat [1]. Salah satu inovasi di sektor ini adalah hadirnya Kereta Api (KA) Bandara yang dioperasikan oleh PT. Railink, yang menyediakan layanan transportasi cepat dan nyaman dari dan menuju bandara besar di Indonesia, seperti Bandara Soekarno-Hatta, Kualanamu, dan Yogyakarta International Airport [2].

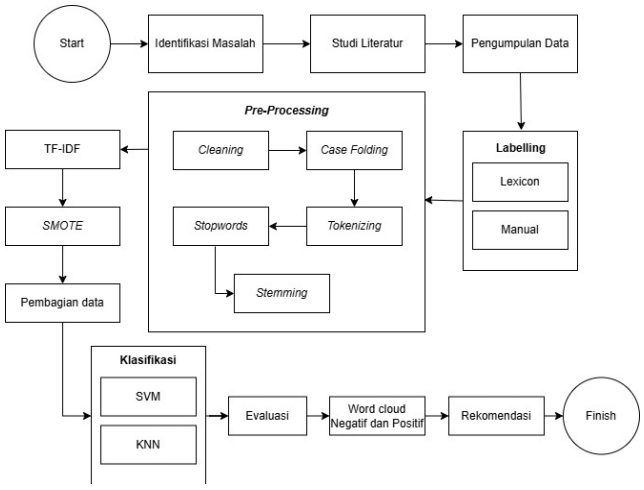
Untuk meningkatkan pengalaman pengguna, PT. Railink meluncurkan Aplikasi KA Bandara sebagai platform digital untuk pemesanan tiket, pengecekan jadwal, serta pembayaran secara daring [3]. Aplikasi ini diharapkan dapat mempermudah proses perjalanan, mengurangi antrean di stasiun, dan memberikan efisiensi waktu bagi penumpang. Namun, berdasarkan ulasan pengguna di Google Play Store, masih terdapat berbagai keluhan terkait performa aplikasi, terutama dalam aspek stabilitas sistem, kesalahan saat proses pembayaran, dan antarmuka pengguna [4]. Keluhan ini menunjukkan adanya kesenjangan antara ekspektasi pengguna dan kinerja aplikasi yang diberikan.

Salah satu pendekatan yang dapat digunakan untuk mengevaluasi persepsi dan pengalaman pengguna terhadap aplikasi adalah analisis sentimen. Analisis sentimen merupakan proses mengidentifikasi dan mengklasifikasikan opini pengguna menjadi kategori positif, negatif, atau netral berdasarkan data teks [5]. Teknik ini bermanfaat sebagai alat business intelligence untuk memahami kebutuhan dan masalah pengguna sehingga dapat digunakan sebagai dasar pengambilan keputusan dalam perbaikan produk [6].

Dalam penelitian ini, analisis sentimen dilakukan terhadap ulasan Aplikasi KA Bandara dengan memanfaatkan dua metode klasifikasi, yaitu Support Vector Machine (SVM) dan Extreme Gradient Boosting (XGBoost). SVM dipilih karena kemampuannya dalam menangani data berdimensi tinggi dan menghasilkan hyperplane optimal yang memisahkan kelas dengan baik [7]. Sedangkan XGBoost dipilih karena efisiensinya dalam proses pelatihan, kemampuannya menangani fitur dengan nilai hilang, dan stabilitas prediksi [8].

Selain itu, penelitian ini menggunakan Synthetic Minority Oversampling Technique (SMOTE) untuk menangani permasalahan ketidakseimbangan kelas pada data ulasan. SMOTE bekerja dengan membuat sampel sintetis dari kelas minoritas, sehingga dapat meningkatkan performa model dalam mengklasifikasikan data [9]. Dua metode pelabelan data digunakan, yaitu pelabelan manual dan berbasis lexicon, untuk menguji konsistensi performa model.

II. METODE PENELITIAN



Gambar 1 Metode Penelitian

Penelitian ini menggunakan Framework of Research Cross-Industry Standard Process for Data Mining (CRISP-DM), yang menggambarkan siklus penelitian dalam Data Mining.

A. *Business Understanding*

Tahap pertama pada penelitian ini adalah mengidentifikasi masalah. Peneliti mengidentifikasi permasalahan yang berkaitan dengan performa Aplikasi KA Bandara PT. Railink berdasarkan ulasan pengguna di Google Play Store. Ulasan tersebut menunjukkan adanya keluhan terhadap stabilitas aplikasi, kesalahan pada proses pembayaran, serta kendala dalam penggunaan fitur tertentu. Permasalahan ini dapat memengaruhi kepuasan pengguna dan citra perusahaan jika tidak segera diatasi.

Berdasarkan identifikasi masalah tersebut, peneliti merumuskan tujuan utama, yaitu membandingkan dua metode klasifikasi populer, Support Vector Machine (SVM) dan Extreme Gradient Boosting (XGBoost), dalam menganalisis sentimen ulasan aplikasi KA Bandara. Perbandingan ini akan dilakukan dengan penanganan ketidakseimbangan data menggunakan Synthetic Minority Oversampling Technique (SMOTE) agar hasil klasifikasi lebih akurat dan representatif.

B. *Data Understanding*

Pada tahap Data Understanding, peneliti mengumpulkan data ulasan aplikasi KA Bandara dari platform Google Play Store. Proses pengambilan data dilakukan dengan menggunakan teknik scraping, yaitu metode pengambilan informasi secara otomatis dari halaman web menggunakan library dari google play scraper. Dalam penelitian ini, scraping dilakukan untuk mendapatkan data teks ulasan beserta atribut pendukung seperti nama pengguna, tanggal ulasan, dan rating bintang yang diberikan. Jumlah data yang berhasil dikumpulkan adalah 670 ulasan yang mencakup periode tertentu sesuai kebutuhan penelitian.

Dalam penelitian ini, proses pelabelan ulasan dilakukan dengan dua pendekatan yaitu *labelling* manual dan *labelling* lexicon.

C. *Data Preparation*

1. *Preprocessing*

3.1.1. *Cleaning*

Cleaning data dilakukan untuk mengurangi noise yang dapat mempengaruhi perhitungan. Dalam proses cleaning data, data yang tidak lengkap dibuang [10].

3.1.2. *Case Folding*

Case Folding adalah proses penyamaan case dalam sebuah dokumen. Membantu menyederhanakan teks dan menjadi seragam serta konsisten dalam penggunaan huruf besar dan kecil [11]. Oleh karena itu peran case folding dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar [12].

3.1.3. *Tokenizing*

Tokenizing adalah proses pemotongan sebuah dokumen menjadi bagian-bagian, yang disebut dengan token. Pada saat bersamaan tokenizing juga berfungsi untuk membuang beberapa karakter tertentu yang dianggap sebagai tanda baca [12].

3.1.4. *Stopwords*

Stopword removal adalah proses penghilangan kata kata yang tidak berkontribusi banyak pada isi dokumen. Kata-kata yang termasuk ke dalam stopword dihilangkan karena memberikan pengaruh yang tidak baik dalam proses text mining seperti kata-kata “and”, “I”, “you”, “with”, “she”, “he”, dan lain-lain [12].

3.1.5. *Stemming*

Stemming merupakan suatu proses untuk menemukan kata dasar dari sebuah kata dengan menghilangkan semua imbuhan (affixes) baik yang terdiri dari awalan (prefixes), sisipan (infixes), akhiran (suffixes),dan confixes (kombinasi dari awalan dan akhiran) pada kata turunan. Stemming adalah proses yang sangat penting untuk mencari kata dasar dari sebuah kata derivatif. Inti dari proses stemming adalah menghilangkan imbuhan pada suatu kata [13].

2. TF IDF

TF-IDF adalah metode pembobotan sebuah kata/term yang memberikan bobot yang berbeda pada setiap term dalam dokumen berdasarkan frekuensi term per dokumen dan frekuensi term pada seluruh dokumen [14]. Pada proses TF-IDF ini dilakukan pembobotan pada kata atau pengubahan data teks menjadi representasi numerik.

$$w_{td} = tf_{td} \times (\log\left(\frac{N}{dft}\right) + 1)$$

Keterangan :
d : dokumen ke-d
t : kata ke-t dari kata kunci
w : bobot d sampai d terhadap kata t
tf : jumlah kata yang dicari pada sebuah dokumen
IDF : *Inverse Document Frequency*
N : jumlah total dokumen
df : jumlah dokumen yang berisi token

3. SMOTE

SMOTE merupakan teknik yang dimanfaatkan untuk mengatasi masalah *class imbalance problem* (CIP). SMOTE bekerja dengan memodifikasi dataset yang tidak seimbang dengan cara membuat data sintetik baru dari kelas minoritas dengan tujuan meningkatkan kinerja dari metode klasifikasi. Pada SMOTE kemungkinan terjadi overfitting yaitu data pada kelas minoritas yang terduplikasi [15]. Berikut rumus SMOTE:

$$synthetic = x_i + rand(0,1) \times (x_{neighbor} - x_i)$$

Keterangan:

x_i = contoh minoritas asli

$x_{neighbor}$ = tetangga terdekat yang dipilih

Rand(0,1) = nilai acak antara 0 dan 1.

D. *Modeling*

1. Klasifikasi dengan metode *Support Vector Machine* (SVM) bertujuan untuk menemukan hyperplane (batas keputusan) optimal yang memisahkan ulasan ke dalam dua kategori, yaitu ulasan positif dan ulasan negatif. Algoritma *Support Vector Machine* (SVM) kemudian digunakan untuk memprediksi data uji dengan membandingkan empat jenis kernel yang berbeda, yaitu Linier, RBF, Polynomial, dan Sigmoid.
2. Klasifikasi dengan metode *Extreame Gradient Boosting* bertujuan untuk mengklasifikasikan ulasan ke dalam dua kategori, yaitu ulasan positif dan ulasan negatif, dengan pendekatan berbasis pohon keputusan yang kuat dan efisien. Metode XGBoost bekerja dengan membentuk model ansambel dari beberapa pohon keputusan yang dilatih secara bertahap (boosting), di mana setiap pohon berikutnya dibentuk untuk memperbaiki kesalahan prediksi dari pohon sebelumnya. Dalam penelitian ini, algoritma XGBoost digunakan untuk mempelajari pola-pola sentimen dalam data pelatihan dan kemudian digunakan untuk memprediksi sentimen pada data uji.

E. *Evaluation*

Evaluasi dalam penelitian ini dilakukan dengan menggunakan confusion matrix untuk mengukur kinerja model berdasarkan metrik recall, precision, accuracy, dan F1-score.

1. *Recall (Sensitivity)*, yaitu perbandingan jumlah data yang mungkin dikenali dengan jumlah seluruh data yang dikenali [16]. Berikut rumus perhitungan *recall*:
$$\frac{TP}{TP + FN} \times 100\%$$
2. *Precision*, yaitu perbandingan jumlah data yang mungkin dikenali dengan jumlah data yang dikenali [16]. Berikut rumus perhitungan *Precision*:
$$\frac{TP}{TP + FP} \times 100\%$$
3. *Accuracy*, yaitu nilai yang menunjukkan tingkat akurasi system dalam mengklasifikasikan secara benar [16]. Berikut rumus perhitungan *Accuracy*:
$$\frac{TN + TP}{TN + TP + FN + FP} \times 100\%$$
4. F1-*Score*, yaitu rata-rata harmonic dari nilai *Precision* dan *Recall* [17]. Berikut rumus F1- *score*:
$$2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\%$$

III. HASIL DAN PEMBAHASAN

A. *Data Understanding*

1. Pengumpulan Data

Pada penelitian ini, peneliti mengambil data *review* pada aplikasi KA Bandara di *Google Play Store* sebanyak 670 *review*. Pada proses *scrapping* yang pertama dilakukan ialah menginstall *package google-play-scraper*, lalu mengambil modul app, sort, dan review.

Setiap ulasan pengguna dilengkapi dengan berbagai informasi, seperti ‘reviewId’ sebagai identitas unik ulasan, ‘userName’ yang menunjukkan nama pengguna aplikasi, ‘userImage’ menampilkan URL *photo profile* pengguna, ‘content’ memuat isi ulasan menurut pengalaman pengguna yang dimana berisikan keluhan atau kritik terhadap aplikasi, ‘*score*’ menampilkan bintang yang diberikan oleh pengguna, ‘thumbsUpCount’ menampilkan jumlah pengguna yang merasa relevan dengan ulasan yang diberikan pengguna lain, ‘reviewCreatedVersion’ dan ‘appVersion’ yang menunjukkan versi aplikasi Ketika ulasan ditulis, ‘replyContent’ menampilkan respon dari pengembang terhadap ulasan yang diberikan pengguna, ‘repliedAt’ menunjukkan waktu pengembang membalas review dari pengguna.

Dari seluruh atribut yang ada, peneliti hanya mengambil atribut ‘content’ untuk dilakukan pelabelan data menggunakan lexicon yang digunakan untuk mengetahui komentar tersebut masuk kedalam kategori positif, negatif, atau netral.

2. *Labelling Data*

1.2.1. Manual

Pada penelitian ini, proses pelabelan dilakukan secara manual berdasarkan sistem perangkingan skor yang diberikan oleh pengguna aplikasi KA Bandara. Skor yang diberikan berkisar antara 1 hingga 5, yang mencerminkan tingkat kepuasan pengguna terhadap layanan yang diterima. Untuk mengklasifikasikan sentimen dari ulasan, dilakukan pengelompokan skor ke dalam dua kategori sentimen, yaitu negatif dan positif**.** Skor dengan nilai 1 hingga 2 dikategorikan sebagai sentimen negatif**,** karena merepresentasikan ketidakpuasan atau keluhan dari pengguna. Sementara itu, skor dengan nilai 3 hingga 5 dimasukkan ke dalam kategori sentimen positif, karena dinilai menunjukkan tingkat kepuasan yang lebih tinggi.

Jumlah masing-masing score yang diperoleh yaitu 356 untuk score 1, 73 untuk score 2, 47 untuk score 3, 37 untuk score 4, 157 untuk score 5.

Melalui proses pelabelan manual ini, diperoleh hasil bahwa dari total 670 data ulasan**,** terdapat 427 data dengan sentimen negatif dan 240 data dengan sentimen positif**.**

1.2.2. Lexicon

Proses pelabelan dengan metode berbasis lexicon digunakan untuk melakukan pelabelan sentiment secara otomatis. Proses analisis dilakukan dengan memanfaatkan dua kamus kata, yakni positive.tsv dan negative.tsv, yang masing-masing berisi daftar kata berkonotasi positif dan negatif. Kedua kamus tersebut diolah menggunakan pandas dan dikonversi ke dalam bentuk set guna mempercepat proses pencocokan kata, dengan seluruh kata terlebih dahulu diseragamkan dalam huruf kecil.

Sebelum klasifikasi sentimen dilakukan, teks ulasan diproses melalui fungsi preprocess_text() untuk mengubah teks menjadi huruf kecil serta menghapus karakter non-alfabet dan angka, sehingga hasil tokenisasi

lebih konsisten. Selanjutnya, fungsi determine_sentiment() digunakan untuk menghitung jumlah kata positif dan negatif pada setiap ulasan. Skor sentimen diperoleh dari selisih jumlah kata positif dan negatif, dengan aturan: skor > 0 dikategorikan sebagai sentimen positif, sedangkan skor ≤ 0 dikategorikan sebagai sentimen negatif.

Hasil pelabelan otomatis menunjukkan bahwa dari 670 ulasan yang dianalisis, sebanyak 550 ulasan tergolong negatif dan 120 ulasan tergolong positif.

B. *Data Pre-Paration*

Judul harus dalam font biasa berukuran 20 pt. Nama pengarang harus dalam font biasa berukuran 11 pt. Jumlah kata judul maksimal 12 kata.

1. *Pre-processing*

Pada tahap pre-processing, dilakukan serangkaian proses meliputi *cleaning, case folding, tokenizing, stopword removal,* dan *stemming* untuk menyiapkan data mentah menjadi lebih terstruktur dan relevan. Adapun tahapannya ditunukkan sebagai berikut:

a. Data Cleaning berfungsi menghilangkan elemen tidak penting seperti tanda baca, angka, karakter khusus, emotikon, serta elemen HTML yang dapat menimbulkan noise pada analisis. Proses ini menggunakan modul *Regular Expression* (re) untuk pencarian/penggantian pola teks dan modul string untuk konstanta bawaan, sehingga data menjadi lebih bersih, konsisten, dan siap untuk transformasi selanjutnya.

Tabel 1 Hasil Proses Cleaning

Sebelum	Sesudah
Kecewa sekali pakai aplikasi ini. Payment sdh berhasil, saldo Livin sdh terpotong tapi muncul notif menunggu pembayaran. Trus ditunggu2 Ticketnya ga berhasil keluar. Memang cuman 5000 harga ticketnya tapi kali berapa orang yg complain udh brp itu, banyak jg.	Kecewa sekali pakai aplikasi ini Payment sdh berhasil saldo Livin sdh terpotong tapi muncul notif menunggu pembayaran Trus ditunggu Ticketnya ga berhasil keluar Memang cuman harga ticketnya tapi kali berapa orang yg complain udh brp itu banyak jg

b. Case Folding dilakukan untuk menyeragamkan penulisan huruf dengan mengubah seluruh karakter menjadi huruf kecil (*lowercase*). Proses ini mencegah perbedaan pemaknaan kata akibat penggunaan huruf kapital, misalnya “Bagus” dan “bagus”. Implementasi dilakukan dengan fungsi casefolding() yang mengonversi teks menggunakan method .lower() dan diterapkan pada seluruh data ulasan melalui fungsi apply() dari pustaka pandas.

Tabel 2 Hasil Proses Case Folding

Sebelum	Sesudah
Kecewa sekali pakai aplikasi ini. Payment sdh berhasil, saldo Livin sdh terpotong tapi muncul notif menunggu pembayaran.Trus ditunggu2 Ticketnya ga berhasil keluar. Memang cuman 5000 harga ticketnya tapi kali berapa orang yg complain udh brp itu, banyak jg.	kecewa sekali pakai aplikasi ini payment sdh berhasil saldo livin sdh terpotong tapi muncul notif menunggu pembayaran trus ditunggu ticketnya ga berhasil keluar memang cuman harga ticketnya tapi kali berapa orang yg complain udh brp itu banyak jg

c. Tokenizing bertujuan memecah kalimat menjadi unit kata yang lebih kecil sebagai dasar ekstraksi fitur dan representasi numerik. Pada penelitian ini digunakan fungsi word_tokenize dari pustaka nltk, dengan dependensi punkt_tab yang diunduh terlebih dahulu. Proses tokenisasi diterapkan melalui fungsi word_tokenize_wrapper() pada setiap baris data di kolom content setelah tahap case folding.

Tabel 3 Hasil Proses Tokenizing

Sebelum	Sesudah
Kecewa sekali pakai aplikasi ini. Payment sdh berhasil, saldo Livin sdh terpotong tapi muncul notif menunggu pembayaran.Trus ditunggu2 Ticketnya ga berhasil keluar. Memang cuman 5000 harga ticketnya tapi kali berapa orang yg complain udh brp itu, banyak jg.	‘kecewa’, ‘sekali’, ‘pakai’, ‘aplikasi’, ‘ini’, ‘payment’, ‘sdh’, ‘berhasil’, ‘saldo’, ‘livin’, ‘sdh’, ‘terpotong’, ‘tapi’, ‘muncul’, ‘notif’, ‘menunggu’, ‘pembayaran’, ‘trus’, ‘ditunggu’, ‘ticketnya’, ‘ga’, ‘berhasil’, ‘keluar’, ‘memang’, ‘cuman’, ‘harga’, ‘ticketnya’, ‘tapi’, ‘kali’, ‘berapa’, ‘orang’, ‘yg’, ‘complain’, ‘udh’, ‘brp’, ‘itu’, ‘banyak’, ‘jg’

d. Stopword removal dilakukan untuk menghapus kata-kata umum yang tidak bermakna penting dalam analisis sentimen, seperti “dan”, “yang”, atau “itu”. Proses ini membantu mengurangi dimensi data dan menekankan kata yang lebih informatif. Pada penelitian ini, stopword removal diterapkan menggunakan pustaka nltk dalam bahasa pemrograman Python.

e. Stemming bertujuan mengembalikan kata ke bentuk dasarnya (*root word*), sehingga variasi kata dapat diseragamkan. Dalam penelitian ini, proses stemming dilakukan menggunakan

pustaka Sastrawi pada Python, yang secara khusus dirancang untuk Bahasa Indonesia.

Tabel 4 Hasil Stemming

Sebelum	Sesudah
Kecewa sekali pakai aplikasi ini. Payment sdh berhasil, saldo Livin sdh terpotong tapi muncul notif menunggu pembayaran. Trus ditunggu2 Ticketnya ga berhasil keluar. Memang cuman 5000 harga ticketnya tapi kali berapa orang yg complain udh brp itu, banyak jg.	['kecewa', 'pakai', 'aplikasi', 'payment', 'hasil', 'saldo', 'livin', 'potong', 'muncul', 'notif', 'tunggu', 'bayar', 'trus', 'tunggu', 'ticketnya', 'hasil', 'harga', 'ticketnya', 'orang', 'complain', 'brp']

2. TF-IDF

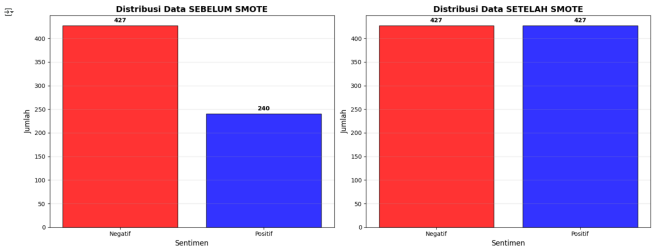
Metode representasi teks yang digunakan dalam penelitian ini adalah TF-IDF (Term Frequency–Inverse Document Frequency), yaitu teknik pembobotan yang menekankan kata penting dengan mempertimbangkan frekuensi kemunculan kata dalam dokumen dan tingkat keunikannya di seluruh korpus. Proses dilakukan dalam dua tahap: pertama, menggunakan CountVectorizer dengan parameter ngram_range=(1,2) dan max_features=320 untuk menghasilkan unigram dan bigram dengan fitur paling signifikan. Kedua, hasilnya diubah menjadi bobot TF-IDF menggunakan TfidfVectorizer dari pustaka sklearn, sehingga diperoleh representasi numerik yang mencerminkan kepentingan setiap kata pada masing-masing dokumen. Berikut hasil pembobotan TF-IDF menggunakan 'max_features=320' :

```
Coords        Values
(0, 129)      0.4212428229893626
(0, 207)      0.18682142616246938
(0, 12)       0.10262305769588684
(0, 212)      0.219656684945705
(0, 34)       0.388407564206127
(0, 250)      0.19699736254213326
(0, 289)      0.219656684945705
(0, 191)      0.18913581927860074
(0, 184)      0.23130517837628137
(0, 215)      0.13013067709492707
(0, 302)      0.219656684945705
(0, 97)       0.22506348536432746
(0, 202)      0.18913581927860074
(0, 71)       0.18682142616246938
(0, 267)      0.24772280776789918
(0, 305)      0.21488754898466353
(0, 203)      0.2386875343168755
(0, 292)      0.22506348536432746
(1, 294)      0.45382613088562745
(1, 199)      0.4653070491302833
(1, 259)      0.42369837179700104
(1, 69)       0.3369736590032404
(1, 306)      0.4653070491302833
(1, 132)      0.2606702627060777
(2, 12)       0.23099659178994697
:     :
(638, 12)     0.4620061812572543
(638, 115)    0.8868767042154672
(639, 86)     1.0
(640, 293)    0.6461055257250746
(640, 284)    0.5816811221661382
(640, 95)     0.4941606234242743
```

Gambar 2 Hasil TF-IDF

3. SMOTE

Dalam analisis sentimen, distribusi kelas yang tidak seimbang dapat membuat model lebih cenderung mengenali kelas mayoritas. Pada data ulasan KA Bandara, terdapat 427 ulasan negatif dan 240 ulasan positif. Ketidakseimbangan ini diatasi dengan SMOTE (Synthetic Minority Oversampling Technique), yang menghasilkan data sintetik untuk kelas minoritas melalui interpolasi, bukan duplikasi. Sebelum penerapan SMOTE, data teks terlebih dahulu direpresentasikan dengan TF-IDF dan dikonversi ke format array agar kompatibel dengan fungsi fit_resample(). Setelah SMOTE, distribusi data menjadi seimbang dengan masing-masing 427 ulasan positif dan negatif, sehingga model dapat belajar lebih adil tanpa bias terhadap kelas mayoritas.



Gambar 3 Perbandingan Penerapan SMOTE

C. *Evalution*

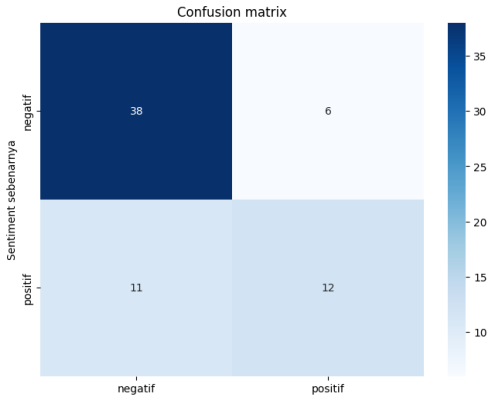
1. Analisis Hasil Pengujian menggunakan *Labelling* Manual

3.1.1. Algoritma *Support Vector Machine*

Model dengan kernel RBF mencapai akurasi 74,63%, dengan presisi dan recall lebih tinggi pada kelas Negatif (0,78 dan 0,86) dibanding kelas Positif (0,67 dan 0,52). Nilai F1-Score rata-rata tertimbang sebesar 0,74 menunjukkan kecenderungan model lebih baik dalam mengenali sentimen negatif.

Tabel 5 Evaluasi menggunakan kernel RBF pada Labelling manual

Kernal RBF	
Akurasi	0.75
Precision	0.77
Recall	0.75
F1-Score	0.75



Gambar 4 Confusion matrix RBF pada Labelling manual

Berdasarkan confusion matrix, sebanyak 38 data negatif berhasil diprediksi dengan benar dan 6 data

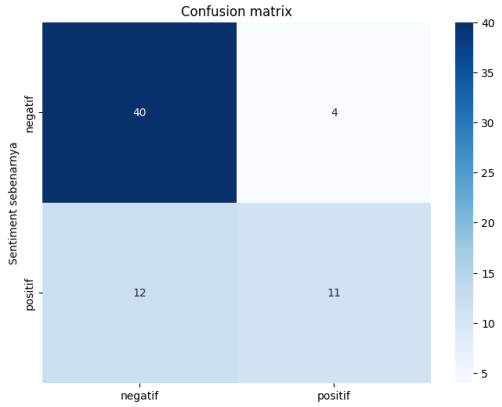
negatif salah diklasifikasikan sebagai positif. Sementara itu, pada kelas positif hanya 12 data terprediksi benar, sedangkan 11 lainnya salah diklasifikasikan sebagai negatif.

Kernel Linear menghasilkan akurasi 76,12% dengan presisi tinggi pada kelas Positif (0,81), namun recall rendah (0,48). Sebaliknya, performa pada kelas Negatif lebih baik (recall 0,91, F1-Score 0,83). Nilai F1-Score rata-rata tertimbang sebesar 0,75.

Tabel 6 Evaluasi menggunakan kernel linier pada Labelling Manual

Kernal Linier	
Akurasi	0.76
Precision	0.81
Recall	0.76
F1-Score	0.78

Hasil confusion matrix menunjukkan bahwa 40 data negatif berhasil diprediksi dengan benar dan 4 data negatif salah diklasifikasikan sebagai positif, sedangkan pada kelas positif hanya 11 data terklasifikasi benar dan 12 data lainnya salah diprediksi sebagai negatif.



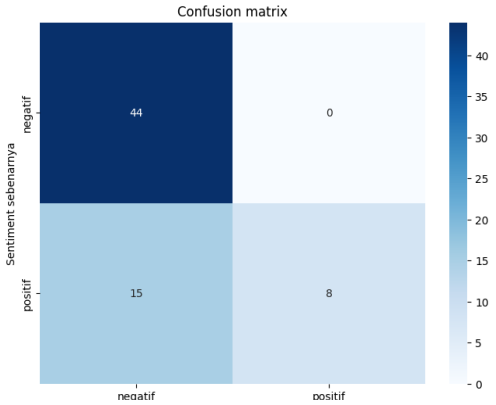
Gambar 5 Confusion matrix Linier pada Labelling manual

Pada kernel Polynomial, akurasi meningkat menjadi 77,61% dengan presisi tinggi (0,92), namun recall relatif tidak seimbang antar kelas. Nilai rata-rata tertimbang F1-Score tercatat 0,74.

Tabel 7 Evaluasi menggunakan kernel Polynomial pada Labelling Manual

Kernal Polynomial	
Akurasi	0.78
Precision	0.92
Recall	0.78
F1-Score	0.81

Hasil confusion matrix menunjukkan bahwa 44 data negatif berhasil diklasifikasikan dengan benar, sedangkan dari 23 data positif hanya 8 data terprediksi benar dan 15 lainnya salah diklasifikasikan sebagai negatif.



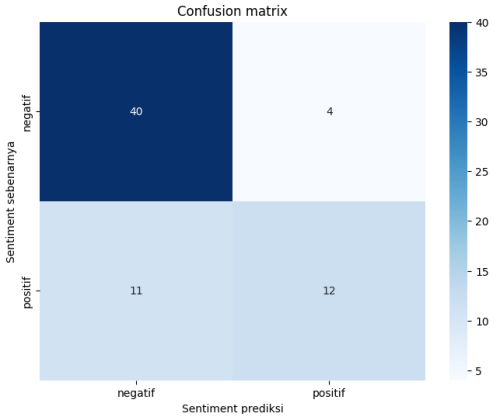
Gambar 6 Confusion matrix Polynomial pada Labelling Manual

Kernel Sigmoid juga memperoleh akurasi 77,61%, dengan performa seimbang namun tetap lebih unggul pada kelas Negatif (F1-Score 0,84) dibanding kelas Positif (0,62). Nilai F1-Score rata-rata tertimbang adalah 0,79.

Tabel 8 Evaluasi menggunakan kernel Sigmoid pada Labelling Manual

Kernal Sigmoid	
Akurasi	0.78
Precision	0.82
Recall	0.78
F1-Score	0.79

Hasil confusion matrix menunjukkan bahwa 40 data negatif berhasil diprediksi dengan benar dan 4 data negatif salah diklasifikasikan sebagai positif, sedangkan pada kelas positif terdapat 12 data terprediksi benar dan 11 data salah diklasifikasikan sebagai negatif.



Gambar 7 Confusion matrix sigmoid pada Labelling manual

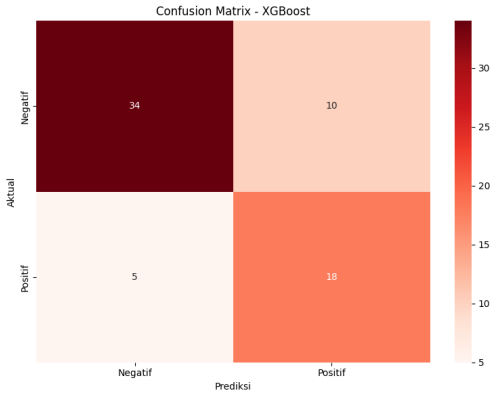
3.1.2. Algoritma *Extreme Gradient Boosting*

Algoritma XGBoost Classifier menunjukkan performa yang seimbang dengan akurasi 77,61%, precision 0,79, recall 0,78, dan F1-Score 0,78, yang mencerminkan keseimbangan antara ketepatan dan kemampuan mengenali data aktual.

Tabel 9 Evaluasi XGBoost pada Labelling Manual

XGBoost	
Akurasi	0.78
Precision	0.82
Recall	0.78
F1-Score	0.79

Hasil confusion matrix menunjukkan bahwa 34 data negatif diklasifikasikan dengan benar sementara 10 salah diprediksi sebagai positif, sedangkan pada kelas positif terdapat 18 prediksi benar dan 5 salah diklasifikasikan sebagai negatif.



Gambar 8 Confusion matrix XGBoost pada Labelling Manual

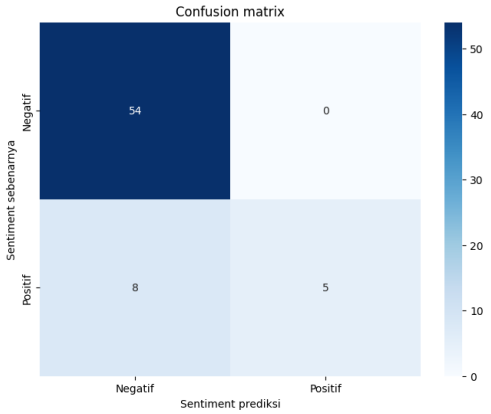
2. Analisis Hasil Pengujian menggunakan *Labelling* Lexicon

3.2.1. Algoritma *Support Vector Machine*

Kernel RBF memperoleh akurasi 88,06% dengan precision 0,95, recall 0,88, dan F1-Score 0,90. Meskipun kinerjanya sangat baik pada kelas Negatif (recall 1,00), performa pada kelas Positif masih terbatas dengan recall 0,38.

Tabel 10 Evaluasi kernel RBF pada Labelling Lexicon

Kernel RBF	
Akurasi	0.88
Precision	0.95
Recall	0.88
F1-Score	0.90



Gambar 9 Confusion matrix RBF pada Labelling Lexicon

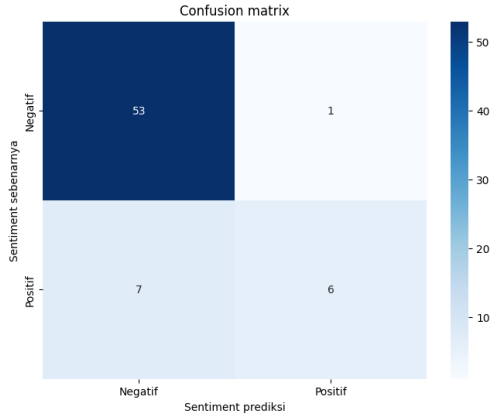
Hasil confusion matrix menunjukkan bahwa seluruh 54 data negatif berhasil diklasifikasikan dengan benar, sedangkan dari 13 data positif hanya 5 teridentifikasi dengan tepat dan 8 lainnya salah diprediksi sebagai negatif.

Kernel Linear juga mencapai akurasi 88,06%, precision 0,92, recall 0,88, dan F1-Score 0,89. Model ini sangat andal pada kelas Negatif (recall 0,98), namun recall kelas Positif relatif rendah (0,46).

Tabel 11 Evaluasi kernel Linear pada Labelling Lexicon

Kernel Linier	
Akurasi	0.88
Precision	0.92
Recall	0.88
F1-Score	0.89

Hasil confusion matrix menunjukkan bahwa dari 54 data negatif, 53 terklasifikasi dengan benar dan 1 salah sebagai positif, sedangkan dari 13 data positif hanya 6 dikenali dengan tepat dan 7 lainnya keliru diprediksi sebagai negatif.



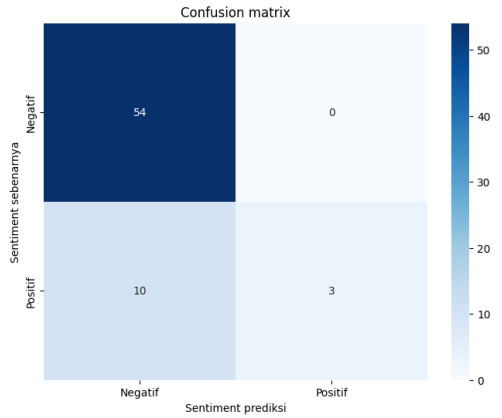
Gambar 10. Confusion matrix Linier pada Labelling Lexicon

Polynomial menghasilkan akurasi 85,07%, precision 0,96, recall 0,85, dan F1-Score 0,89. Model ini sangat kuat dalam mengenali kelas Negatif (recall 1,00), tetapi recall kelas Positif hanya 0,23, yang menunjukkan ketidakseimbangan antar kelas.

Tabel 12 Evaluasi kernel Polynomial pada Labelling Lexicon

Kernel Polynomial	
Akurasi	0.85
Precision	0.96
Recall	0.85
F1-Score	0.89

Hasil confusion matrix menunjukkan seluruh 54 data negatif berhasil diprediksi dengan benar, sedangkan hanya 3 dari 13 data positif yang terklasifikasi tepat. Ketimpangan ini berdampak pada turunnya macro average recall menjadi 0,62 dan weighted average F1-score sebesar 0,81.



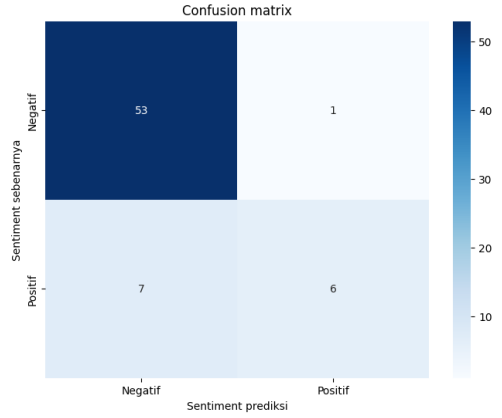
Gambar 11. Confusion matrix Polynomial pada Labelling Lexicon

Kernel Sigmoid mencapai akurasi 88,06%, precision 0,93, recall 0,88, dan F1-Score 0,89. Sama seperti kernel lainnya, model ini menunjukkan hasil yang sangat baik pada kelas Negatif, namun recall untuk kelas Positif masih rendah (0,46).

Tabel 13 Evaluasi kernel Sigmoid pada Labelling Lexicon

Kernel Sigmoid	
Akurasi	0.88
Precision	0.93
Recall	0.88
F1-Score	0.89

Hasil confusion matrix menunjukkan bahwa 53 dari 54 data negatif terklasifikasi dengan benar, sedangkan hanya 6 dari 13 data positif yang dikenali, dengan mayoritas kesalahan terjadi pada kelas positif yang diprediksi sebagai negatif.



Gambar 12. Confusion matrix Sigmoid pada Labelling Lexicon

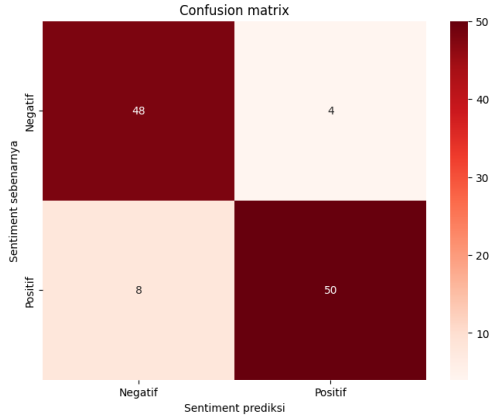
3.2.2. Algoritma *Extreme Gradient Boosting*

Pengujian menggunakan XGBoost menghasilkan akurasi 89%, dengan 98 dari 110 data uji berhasil diklasifikasikan dengan benar. Untuk kelas Negatif, model mencapai precision 0,86, recall 0,92, dan F1-Score 0,89. Sementara pada kelas Positif, precision sebesar 0,93, recall 0,86, dan F1-Score 0,89. Hasil ini menunjukkan bahwa XGBoost memiliki performa seimbang dan andal dalam membedakan kedua kelas sentimen.

Tabel 14 Evaluasi XGBoost pada Labelling Lexicon

XGBoost	
Akurasi	0.89
Precision	0.89
Recall	0.89
F1-Score	0.89

Confusion matrix menunjukkan bahwa dari 52 data negatif, 48 terklasifikasi benar dan 4 salah diprediksi positif, sedangkan dari 58 data positif, 50 dikenali dengan tepat dan 8 salah diprediksi negatif. Distribusi kesalahan ini relatif seimbang, sehingga tidak tampak adanya bias dominan terhadap salah satu kelas.



Gambar 13. Confusion matrix XGBoost pada Labelling Lexicon

3. Perbandingan Algoritma *Support Vector Machine* (SVM) dan *Extreme Gradient Boosting* (GXBoost)

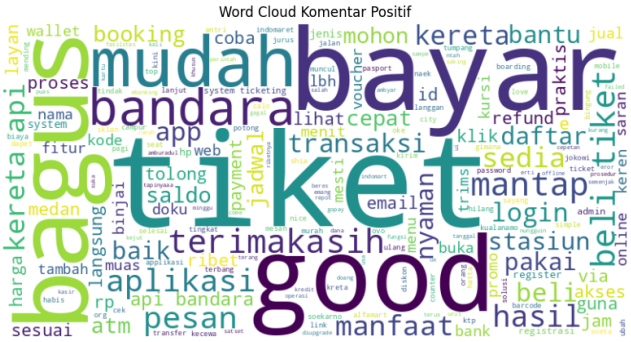
Hasil perbandingan menunjukkan bahwa SVM dan XGBoost memperoleh performa terbaik ketika menggunakan labelling berbasis lexicon. Pendekatan ini mampu menghasilkan pembagian sentimen yang lebih akurat, sehingga model belajar dan mengklasifikasikan data dengan lebih optimal. Berikut hasil perbandingan dari labelling terbaik yang telah diuji:

Tabel 15 Perbandingan Algoritma

Algoritma		**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVM	RBF	0.88	0.95	0.88	0.90
	Linier	0.88	0.92	0.88	0.89
	Poly	0.85	0.96	0.85	0.89
	Sigmoid	0.88	0.92	0.88	0.89
XGBoost		0.89	0.89	0.89	0.89

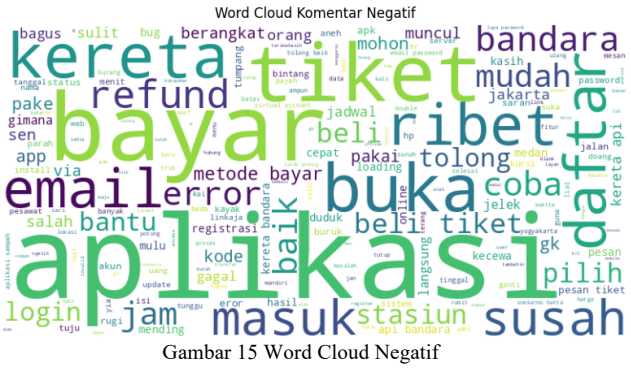
4. Word Cloud

Hasil analisis selanjutnya divisualisasikan menggunakan *word cloud* untuk menampilkan kata-kata yang paling sering muncul dalam dataset. Gambar berikut menunjukkan word cloud untuk komentar positif:



Gambar 14 Word Cloud Positif

Berikut hasil *word cloud* komentar negatif:



Gambar 15 Word Cloud Negatif

IV. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Analisis sentimen dengan SVM dan XGBoost diawali dengan pengumpulan data ulasan dari Google Play Store, dilanjutkan preprocessing (cleaning, case folding, tokenizing, stopword removal, stemming), pembobotan TF-IDF, serta penanganan ketidakseimbangan data menggunakan SMOTE.
2. Model SVM diuji dengan empat kernel (Linear, RBF, Polynomial, Sigmoid), sedangkan XGBoost dioptimalkan melalui parameter seperti max depth, learning rate, gamma, dan base score. Evaluasi kedua model menggunakan accuracy, precision, recall, dan F1-score.
3. Hasil evaluasi menunjukkan XGBoost memiliki akurasi tertinggi (0,89), unggul tipis dari SVM (0,88 untuk RBF, Linear, dan Sigmoid; 0,85 untuk Polynomial). Hal ini menegaskan keunggulan XGBoost dalam menangani pola kompleks dengan lebih stabil.
4. Analisis ulasan menunjukkan dominasi sentimen negatif dengan rekomendasi utama berupa perbaikan sistem pembayaran serta peningkatan stabilitas aplikasi.

REFERENSI

[1] M. D. Setiawan dan R. Andriani, "Perkembangan Transportasi Publik di Indonesia," *Jurnal Transportasi Indonesia*, vol. 12, no. 2, pp. 45–53, 2020.

[2] PT Railink, "Profil Perusahaan PT Railink," [Online]. Available: https://www.railink.co.id. [Accessed: Aug. 8, 2025].

[3] S. Amalia, A. H. Prasetyo, dan F. A. Maulana, "Analisis Sentimen pada Ulasan Aplikasi Menggunakan Metode Klasifikasi," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 1, pp. 23–31, 2021.

[4] Google Play Store, "Aplikasi KA Bandara," [Online]. Available: https://play.google.com/store/apps/details?id=id.co.railink. [Accessed: Aug. 8, 2025].

[5] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[6] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, dan M. Stede, "Lexicon-Based Methods for Sentiment Analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[7] C. Cortes dan V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] T. Chen dan C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, dan W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[10] Kamila, I., Khairunnisa, U., & Mustakim. (2019). Perbandingan Algoritma K-Means dan K-Medoids untuk Pengelompokan. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, *5*(1), 119–125.

[11] Puspitasari, R., & Dwi Indriyanti, A. (2024). *ANALISIS SENTIMEN OPINI PUBLIK TERHADAP KEBIJAKAN BARU SKRIPSI PADA MEDIA SOSIAL TWITTER MENGGUNAKAN METODE NAÏVE BAYES*.

[12] Alita, D., & Isnain, A. R. (2020). Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier. *Jurnal Komputasi*, *8*(2). https://doi.org/10.23960/komputasi.v8i2.2615

[13] Guterres, A., Gunawan, & Santoso, J. (2019). Stemming Bahasa Tetun Menggunakan Pendekatan Rule Based. *Teknika*, *8*(2), 142–147. https://doi.org/10.34148/teknika.v8i2.224

[14] Cahyani, D. E., & Patasik, I. (2021). Performance comparison of tf-idf and word2vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics*, *10*(5). https://doi.org/10.11591/eei.v10i5.3157

[15] Cahyaningtyas, C., Nataliani, Y., & Widiasari, I. R. (2021). Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis SMOTE. *AITI*, *18*(2), 173–184. https://doi.org/10.24246/aiti.v18i2.173-184

[16] Sholawati, M., Auliasari, K., & Ariwibisono, FX. (2022). PENGEMBANGAN APLIKASI PENGENALAN BAHASA ISYARAT ABJAD SIBI MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN). *JATI (Jurnal Mahasiswa Teknik Informatika)*, *6*(1). https://doi.org/10.36040/jati.v6i1.4507

[17] Fadiyah Basar, T., Ratnawati, D. E., & Arwani, I. (2022). *Analisis Sentimen Pengguna Twitter terhadap Pembayaran Cashless menggunakan Shopeepay dengan Algoritma Random Forest* (Vol. 6, Issue 3). http://j-ptiik.ub.ac.id