

Analisis Pengaruh Random Search Pada Logistic Regression dan Support Vector Machine Dalam Klasifikasi Sentimen Aplikasi PDAM Info

Suci Awalia Ramadani¹, Heliawaty Hamrul^{2*}, Nurhikma Arifin³

^{1,2,3}Fakultas Teknik, Teknik Informatika, Universitas Sulawesi Barat, Majene, Indonesia

E-mail: ¹suciawaliah27@gmail.com, ^{2*}heliawatyhamrul@unsulbar.ac.id,

³nurhikma_arifin@unsulbar.ac.id

(* : corresponding author)

Abstrak

Transformasi digital mendorong kebutuhan untuk menghadirkan layanan publik berbasis teknologi yang cepat dan responsif melalui aplikasi, salah satunya aplikasi PDAM Info. Ulasan pengguna aplikasi merupakan informasi penting bagi pengembang untuk meningkatkan kualitas layanan. Banyaknya ulasan aplikasi menyulitkan proses analisis secara manual, sehingga diperlukan analisis sentimen berbasis teks menggunakan *machine learning*. Algoritma *machine learning* secara *default* seringkali belum optimal dalam melakukan klasifikasi, metode *Random Search* dipercaya dapat meningkatkan kinerja hasil klasifikasi. Tujuan penelitian ini untuk mengetahui hasil analisis sentimen, serta mengetahui pengaruh *Random Search* dalam klasifikasi sentimen aplikasi PDAM Info. Dataset terdiri dari 2.400 ulasan dari *Google Play Store* yang setelah *preprocessing* menjadi 1.677 data, dilabeli menggunakan metode *Lexicon-Based* dan direpresentasikan dengan TF-IDF. Klasifikasi dilakukan menggunakan *Logistic Regression* dan SVM, metode *hyperparameter tuning* yang digunakan adalah *Random Search*. Hasil menunjukkan sentimen negatif mendominasi ulasan pengguna terkait cakupan wilayah dan metode pembayaran. Penerapan *Random Search* meningkatkan kinerja kedua algoritma dengan *Accuracy* 88% dan *F1-score* 83%, serta menunjukkan perbaikan prediksi kelas positif dan netral pada data tidak seimbang. Kontribusi penelitian ini memberikan informasi terkait persepsi pengguna bagi pengembang aplikasi PDAM Info serta membuktikan bahwa *Random Search* dapat meningkatkan kinerja model *machine learning* dan memperbaiki prediksi kelas pada data tidak seimbang.

Kata kunci: analisis sentimen, logistic regression, PDAM Info, random search, support vector machine

Abstract

Digital transformation increases the demand for fast and responsive technology-based public services through mobile applications, including the PDAM Info application. User reviews provide important insights for improving service quality, but their large volume makes manual analysis inefficient, requiring text-based sentiment analysis using machine learning. Default machine learning parameters are often suboptimal; therefore, Random Search is applied to improve classification performance. This study analyzes user sentiment and examines the effect of Random Search on sentiment classification of the PDAM Info application. A total of 2,400 Google Play Store reviews were collected, resulting in 1,677 data after preprocessing, labeled using a lexicon-based approach, and represented using TF-IDF. Logistic Regression and Support Vector Machine were used for classification with Random Search for hyperparameter tuning. The results indicate that negative sentiment dominates user reviews, mainly related to service coverage and payment methods. Random Search improves classification performance, achieving 88% accuracy and 83% F1-score, particularly in predicting positive and neutral classes on imbalanced data. The contribution of this study provides insights into user perceptions for PDAM Info application developers and demonstrates that Random Search can improve machine learning model performance and enhance class prediction on imbalanced data.

Keywords: logistic regression, PDAM Info, random search, sentiment analysis, support vector machine

1. PENDAHULUAN

Perkembangan transformasi digital mendorong kebutuhan untuk menghadirkan layanan publik berbasis teknologi yang cepat dan responsif salah satunya adalah pengembangan aplikasi *mobile*. Aplikasi PDAM Info, yang dirancang oleh PT Bima Sakti Alterra pada tahun 2018 merupakan platform digital yang sudah diunduh melebihi 500 ribu kali di *Google Play Store* yang bertujuan meningkatkan kualitas layanan distribusi air bersih. Aplikasi ini memfasilitasi

pelanggan untuk mengelola layanan air secara mandiri dengan mengakses informasi tagihan, melacak pola konsumsi, melakukan pembayaran online, serta menyediakan notifikasi jatuh tempo, layanan pengaduan, dan pembaruan informasi terkini. Kemudahan ini secara signifikan meningkatkan efisiensi, transparansi, dan aksesibilitas dalam pengelolaan layanan air [1].

Persepsi dan pengalaman pengguna menjadi sangat krusial dalam dinamika perubahan pesat saat ini [2]. Umpan balik pengguna berupa ulasan dan rating pada aplikasi, merupakan sumber data esensial untuk mengukur kepuasan dan kualitas layanan [2]. Karena keragaman bahasa, nada, dan detail dalam ulasan pengguna yang menghambat analisis manual, analisis sentimen berbasis teks direkomendasikan sebagai pendekatan efektif untuk mengungkap pola sentimen yang terkandung [3]. Analisis sentimen adalah aplikasi *text mining* yang bertujuan mengidentifikasi nada emosional positif, negatif, atau netral dari opini. Secara strategis, teknik ini penting bagi *business intelligence* untuk memahami persepsi pelanggan, membantu pemerintah dalam menilai opini publik terhadap kebijakan, dan efektif dalam mengekstraksi opini dari media social [4].

Berbagai algoritma machine learning telah banyak digunakan dalam analisis sentimen, di antaranya *Logistic Regression* dan *Support Vector Machine (SVM)*. Penelitian terdahulu menunjukkan bahwa *Logistic Regression* mampu memberikan kinerja yang baik dalam analisis sentimen terhadap aplikasi TikTok melalui perbandingan algoritma *Naïve Bayes*, *Logistic Regression*, dan *Support Vector Machine* menunjukkan bahwa *Logistic Regression* menghasilkan performa terbaik dengan akurasi 84%, sementara *Support Vector Machine* mencapai akurasi 82% [5]. Penelitian lain yang membandingkan *Support Vector Machine* dan *Random Forest* pada analisis sentimen komentar YouTube terkait *boyband* BTS juga membuktikan bahwa *Support Vector Machine* lebih unggul, dengan akurasi 96% pada data latih dan 85% pada data uji, dibandingkan *Random Forest* yang memperoleh akurasi 82% pada data latih dan 80% pada data uji [6]. Temuan-temuan tersebut menunjukkan bahwa *Logistic Regression* dan *Support Vector Machine* merupakan algoritma yang efektif dan kompetitif dalam analisis sentimen berbasis teks.

Logistic Regression unggul dengan kemampuannya dalam mengidentifikasi hubungan antar fitur dan kecepatan pelatihan model saat berhadapan dengan *dataset* besar [7]. Sementara *Support Vector Machine* unggul dengan kemampuannya yang dapat memberikan akurasi tinggi karena dapat memisahkan kelas data terutama dalam ruang berdimensi tinggi [8]. Meskipun *Logistic Regression* dan *Support Vector Machine* efektif dalam mengklasifikasikan data, banyak studi terkait analisis sentimen terutama yang telah dijelaskan sebelumnya cenderung menggunakan konfigurasi *default* tanpa melibatkan metode *hyperparameter tuning*. Padahal pemilihan *hyperparameter* yang akurat yang bertindak sebagai konfigurasi pengatur algoritma merupakan aspek esensial untuk memaksimalkan kinerja dan meningkatkan kualitas hasil klasifikasi [9].

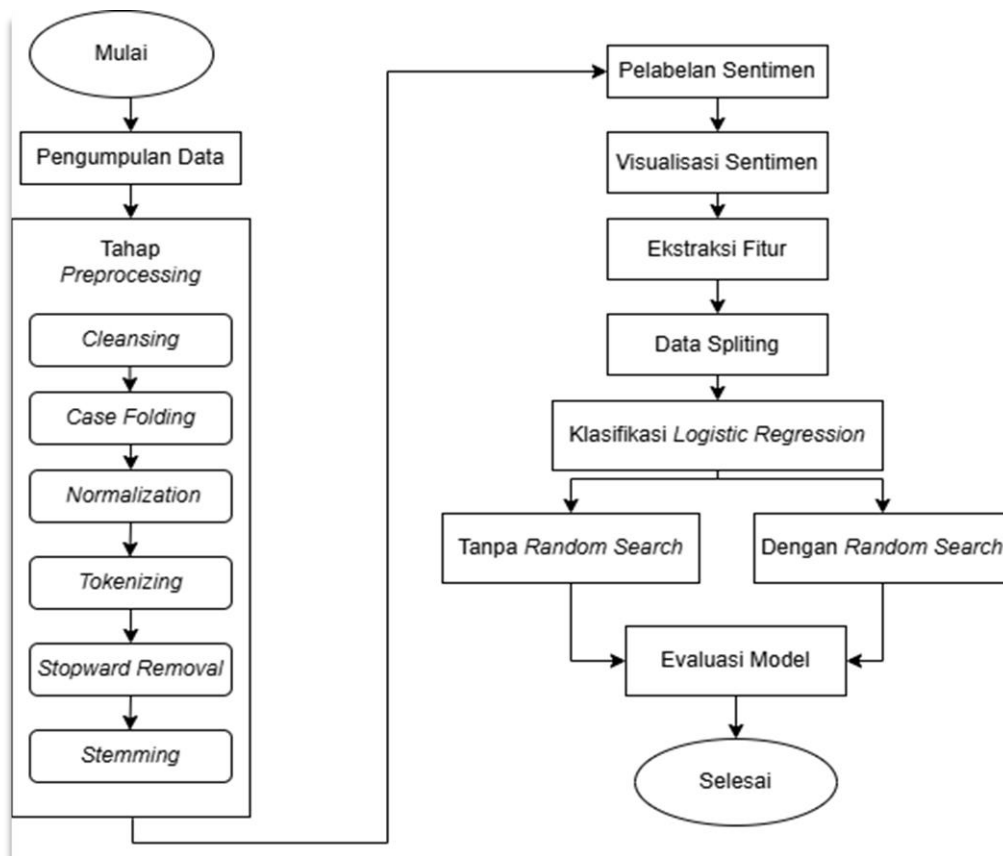
Teknik optimasi *hyperparameter* yang umum digunakan ialah *Grid Search* serta *Random Search*. *Grid Search* adalah pendekatan yang mengevaluasi setiap kombinasi parameter secara sistematis dan *Random Search* adalah pendekatan yang lebih efisien karena memilih kombinasi parameter secara acak [10]. Penelitian perbandingan *Grid Search* dan *Random Search* menunjukkan bahwa *Random Search* tidak hanya lebih efisien dari segi waktu eksekusi tetapi juga mampu mencapai akurasi tertinggi sebesar 82% dibandingkan *Grid Search* 81,50 [10]. Dengan demikian, studi ini menerapkan metode *Random Search*.

Meskipun analisis sentimen telah banyak diterapkan pada berbagai aplikasi dan layanan digital, hingga saat ini belum ditemukan secara khusus penelitian yang menganalisis sentimen ulasan pengguna pada aplikasi PDAM Info. Di sisi lain, penelitian terdahulu yang telah dijelaskan sebelumnya menunjukkan bahwa algoritma *Logistic Regression* dan *Support Vector Machine* memiliki kinerja yang baik dalam analisis sentimen, sehingga sering digunakan dalam metode klasifikasi. Namun, sebagian besar penelitian tersebut masih menerapkan kedua algoritma tersebut dengan konfigurasi parameter *default* tanpa mempertimbangkan metode *hyperparameter tuning*. Studi ini bertujuan untuk mengetahui hasil analisis sentimen aplikasi PDAM Info serta mengetahui pengaruh *hyperparameter tuning* *Random Search* pada algoritma *Logistic Regression* dan *Support Vector Machine* dalam mengklasifikasikan sentimen tanggapan pengguna aplikasi PDAM Info. Sumber data penelitian ini berasal dari ulasan pengguna aplikasi PDAM Info.

Sebelum dilakukan proses klasifikasi, data terlebih dahulu melalui tahap pra-pemrosesan yang meliputi *cleaning*, *case folding*, *normalization*, *tokenizing*, *stopword removal*, dan *stemming*. Selanjutnya, data ulasan dilabeli menggunakan metode *Lexicon-Based* dan direpresentasikan ke dalam bentuk numerik menggunakan metode TF-IDF. Proses klasifikasi kemudian dilakukan menggunakan algoritma *Logistic Regression* dan *Support Vector Machine*. Selain itu, diterapkan metode *Random Search* sebagai teknik *hyperparameter tuning* untuk meningkatkan kinerja hasil klasifikasi. Hasil penelitian menunjukkan bahwa sentimen ulasan pada aplikasi PDAM Info didominasi oleh ulasan negatif, yang umumnya berkaitan dengan keluhan mengenai cakupan wilayah layanan dan metode pembayaran. Evaluasi kinerja menunjukkan bahwa penerapan *Random Search* pada algoritma *Logistic Regression* dan *Support Vector Machine* mampu meningkatkan kinerja klasifikasi, yang ditandai dengan peningkatan nilai *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Selain itu, hasil confusion matrix menunjukkan perbaikan kemampuan model dalam memprediksi kelas positif dan netral yang sebelumnya lemah akibat ketidakseimbangan data. Hal ini menegaskan bahwa *Random Search* berkontribusi dalam menghasilkan prediksi kelas sentimen yang lebih seimbang.

2. METODE PENELITIAN

Berikut adalah alur penelitian ataupun tahap-tahap penelitian yang dipakai pada studi ini.



Gambar 1. Alur Penelitian

Pada Gambar 1 menyajikan alur atau tahapan studi ini yang mencakup seluruh proses mulai dari pengumpulan data, pra-proses data, penerapan metode atau algoritma yang digunakan, hingga tahap evaluasi hasil.

2.1. Pengumpulan Data

Tahapan pengumpulan data ulasan aplikasi PDAM Info dengan memakai teknik *scraping* dengan *API google-play-scrapers*, kemudian data diolah dan dimanipulasi secara efisien menggunakan pustaka *pandas*.

2.2. Tahap *Preprocessing*

Tahap *preprocessing* merupakan bagian krusial pada proses analisis teks yang memiliki tujuan agar dapat mempersiapkan serta menyaring data mentah sehingga layak digunakan pada proses klasifikasi [11]. Adapun tahapan yang dilakukan dalam proses *preprocessing* data mencakup:

- Cleaning*, dalam tahapan ini dataset mentah disisihkan dari beberapa elemen seperti angka, simbol, dan karakter lain yang dianggap tidak bermakna besar pada analisis.
- Casefolding*, dalam tahapan ini semua huruf besar diubah menjadi huruf kecil. Supaya kata yang sebenarnya sama namun ditulis berbeda tidak dianggap berbeda dalam proses analisis selanjutnya.
- Normalization*, pada tahap ini yaitu mengubah kata yang salah eja atau berbentuk singkatan kedalam bahasa yang baku.
- Tokenizing*, dalam tahapan ini ulasan kalimat dipisah menjadi bagian-bagian kata.
- Stopword Removal*, menghapus keberadaan kata yang kurang relevan ataupun kurang informatif terhadap makna atau konteks analisis.
- Stemming*, pada tahap ini yaitu mengubah suatu kata menjadi kata dasar dengan cara menghapus imbuhan seperti awalan, sisipan, ataupun akhiran yang melekat pada kata.

2.3. Pelabelan *Lexicon Based*

Pendekatan *Lexicon Based* merupakan metode umum yang dipakai guna melakukan analisis sentimen terhadap sebuah teks. Metode ini mengandalkan kamus sentimen, yaitu daftar kata yang telah diberi label polaritas positif dan negatif [12]. Dalam praktiknya, analisis dilakukan dengan memberikan skor sentimen pada setiap teks berdasarkan frekuensi kemunculan kata-kata dari kamus tersebut. Penentuan polaritas teks positif, negatif, atau netral didasarkan pada perbandingan jumlah kata positif dengan kata negatif yang teridentifikasi [12]. Pada studi ini label sentimen yang digunakan adalah negatif, positif, dan netral.

2.4. Visualisasi Sentimen

Wordcloud menjadi representasi visual data teks yang menyajikan frekuensi munculnya suatu kata dalam sebuah dokumen teks, kata yang selalu terlihat kemunculannya dalam dokumen akan divisualisasikan menjadi ukuran huruf yang lebih besar. Sebaliknya, kata yang kemunculannya sedikit akan divisualisasikan menjadi ukuran huruf yang lebih kecil [13].

2.5. Ekstraksi Fitur

TF-IDF (*Term Frequency – Inverse Document Frequency*) menjadi satu dari beberapa metode ekstraksi fitur yang banyak dipakai pada proses pengolahan teks. Teknik ini menjadi kolaborasi antara *Term Frequency* (TF) dengan *Inverse Document Frequency* (IDF), di mana nilai akhir TF-IDF diperoleh melalui hasil perkalian keduanya, metode ini bertujuan untuk menyematkan bobot yang lebih besar pada kata yang kemunculannya paling banyak dalam suatu dokumen, namun kemunculannya jarang ada pada keseluruhan kumpulan dokumen [14]. Selain itu, TF-IDF juga dapat dimanfaatkan sebagai metode seleksi fitur karena mampu melakukan identifikasi terhadap berbagai kata yang paling relevan serta berkontribusi besar terhadap isi dokumen melalui perhitungan bobotnya, dengan memilih setiap kata yang bernilai TF-IDF paling tinggi [15]. Persamaan umum untuk menghitung nilai TF-IDF dapat dituliskan sebagai berikut [14]:

$$TF * IDF(d, t) = TF(d, t) \times \log \frac{n}{df(t)} \quad (1)$$

Keterangan :

- $TF * IDF(d, t)$: Nilai bobot TF-IDF
 $TF(d, t)$: Frekuensi kata t yang muncul dalam dokumen d.
 n : Jumlah dari seluruh kumpulan dokumen.
 $df(t)$: Jumlah dari dokumen yang memuat kata t.

2.6. Klasifikasi Logistic Regression

Logistic Regression adalah teknik analisis statistik untuk memodelkan hubungan antar variabel yang mana variabel target bersifat kategorik (nominal atau ordinal), dan metode *Maximum Likelihood Estimation* (MLE) digunakan untuk mengoptimasi nilai parameter atau koefisiennya [16]. *Logistic Regression* memanfaatkan fungsi logistik atau fungsi sigmoid untuk menghitung probabilitas terjadinya suatu peristiwa, mengubah nilai input dari rentang tak terbatas menjadi nilai probabilitas antara 0 dan 1 [17]. Berikut adalah persamaan linier yang mendasari *Logistic Regression* [18]:

$$g_i(x) = \beta_{i0} + \beta_{i1}x_1 + \beta_{i2}x_2 \dots + \beta_{ip}x_p \quad (2)$$

Untuk klasifikasi dengan *multiclass*, menggunakan jenis *Logistic Regression Multinomial* yang membentuk persamaan logit K-1, dalam kasus ini membentuk 2 persamaan logit. Setiap persamaan logit membandingkan peluang suatu kategori terhadap satu kategori pembanding, misalnya dengan kategori 0 sebagai pembanding, persamaan logit yang terbentuk adalah [18].

$$g_1(x) = \log \frac{P(Y=1|x)}{P(Y=0|x)} = \log \frac{n_1(x)}{n_0(x)} = \beta_0 + \beta_1x_1 + \beta_2x_2 \dots + \beta_px_p \quad (3)$$

$$g_2(x) = \log \frac{P(Y=2|x)}{P(Y=0|x)} = \log \frac{n_2(x)}{n_0(x)} = \beta_0 + \beta_1x_1 + \beta_2x_2 \dots + \beta_px_p \quad (4)$$

Merujuk pada kedua persamaan logit (3) dan (4), diperoleh probabilitas pada setiap kategori respon yang mencakup [18]:

$$P(Y = 1|x) = \pi_1(x) = \frac{\exp g_1(x)}{1 + \exp g_1(x) + \exp g_2(x)} \quad (5)$$

$$P(Y = 2|x) = \pi_2(x) = \frac{\exp g_2(x)}{1 + \exp g_1(x) + \exp g_2(x)} \quad (6)$$

$$P(Y = 0|x) = \pi_0(x) = \frac{1}{1 + \exp g_1(x) + \exp g_2(x)} \quad (7)$$

Untuk klasifikasi *multiclass* lebih dari dua kelas, *Logistic Regression* dapat diperluas dengan menghitung probabilitas setiap kelas dan klasifikasi akhir ditentukan oleh kelas dengan nilai probabilitas tertinggi [19].

2.7. Klasifikasi Support Vector Machine

Support Vector Machine merupakan salah satu algoritma klasifikasi machine learning yang bersifat *supervised learning*. Proses klasifikasi menggunakan Support Vector Machine (SVM) dilakukan dengan menentukan sebuah hyperplane atau batas keputusan yang mampu memisahkan data dari kelas yang berbeda [20]. Hyperplane optimal diperoleh dengan memaksimalkan nilai margin, yaitu jarak antara hyperplane dengan data terdekat dari masing-masing kelas yang dikenal sebagai support vector [20]. Apabila data dapat dipisahkan secara sempurna menggunakan hyperplane linear, maka metode yang digunakan disebut sebagai SVM linier, sebaliknya ketika data tidak dapat dipisahkan secara linear dengan baik, SVM menerapkan fungsi kernel untuk mentransformasikan data ke dalam ruang fitur berdimensi lebih tinggi [20]. Berdasarkan pendekatan tersebut, fungsi keputusan SVM linier dapat dirumuskan sebagaimana ditunjukkan pada Persamaan (8) [20].

$$f(x) = \text{sign}(w \cdot x + b) \quad (8)$$

Pada persamaan tersebut, $f(x)$ menyatakan fungsi prediksi, w merupakan vektor normal yang membentuk hyperplane, x adalah vektor fitur masukan, dan b berperan sebagai bias atau intersep

2.8. Hyperparameter Tuning Random Search

Hyperparameter merupakan konfigurasi model yang nilainya harus ditetapkan sebelum proses pelatihan dimulai. Pemilihan *hyperparameter* yang akurat sangat krusial karena secara fundamental mengatur cara kerja algoritma *machine learning*, sehingga memiliki pengaruh determinan terhadap kinerja dan kualitas hasil klasifikasi model [9]. Untuk mengidentifikasi konfigurasi optimal ini, studi ini mengimplementasikan metode *Random Search*. Metode *Random Search* adalah sebuah teknik untuk mengoptimalkan kinerja model dengan mencari kombinasi *hyperparameter* terbaik dari suatu algoritma pembelajaran, *random search* bekerja dengan cara mengambil sampel acak sejumlah kombinasi *hyperparameter* dari rentang atau distribusi nilai yang telah ditentukan [8].

2.9. Evaluasi Model

Evaluasi kinerja model menjadi upaya guna melakukan penilaian terhadap kemampuan model dalam menjalankan tugas klasifikasi dengan baik [21]. Untuk mengevaluasi kinerja suatu model metrik umum yang dipakai mencakup *accuracy*, *precision*, *recall*, serta *f1-score* [21].

a. *Accuracy* melakukan pengukuran terhadap proporsi prediksi yang benar dari seluruh total prediksi [22].

$$\text{Accuracy} = \frac{TP+TN+TNet}{TP+FN+FNet+FP+TN+FNet+FP+FN+TNet} \quad (9)$$

b. *Precision* menggambarkan kemampuan model dalam menghasilkan prediksi positif yang sesuai dengan kelas sebenarnya [22].

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (10)$$

c. *Recall* melakukan pengukuran terhadap seberapa besar proporsi hasil positif yang sukses dikenali dengan benar dari seluruh data yang sejatinya positif [22].

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (11)$$

d. *F1-Score* yaitu nilai rata-rata harmonis yang mempertimbangkan nilai *Precision* dan *Recall* secara bersamaan [22].

$$F1\text{-Score} = 2 \times \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \quad (12)$$

3. HASIL DAN PEMBAHASAN

3.1. Pengumpulan Data

Sumber data penelitian ini diperoleh dari ulasan pengguna pada aplikasi PDAM Info yang dikumpulkan melalui proses scraping dengan memanfaatkan pustaka *google-play-scrapers*. Proses scraping berhasil mengumpulkan sebanyak 2.400 data ulasan yang berasal dari rentang waktu September 2018 hingga September 2025, dengan menerapkan parameter *Sort.NEWEST* dan menggunakan ID aplikasi *id.co.limasakti.pdaminfo* sebagai objek pengambilan data.

3.2. Tahap Preprocessing

Tahapan yang dilakukan pada *preprocessing* (praproses) data, yaitu *cleaning*, *casefolding*, *tokenizing*, *normalization*, *stopword removal*, dan *stemming* :

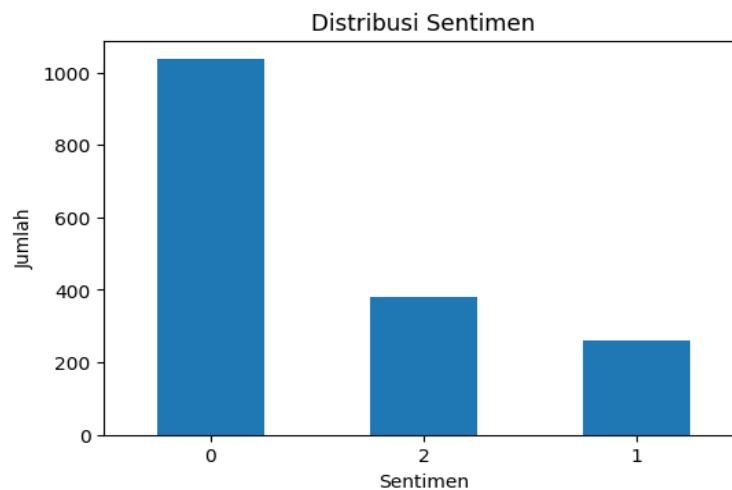
Tabel 1. Tahap *Preprocessing*

<i>Preprocessing</i>	Sebelum	Sesudah
<i>Cleaning</i>	PDAM Tirta Kalwedo semakin didepan 🇮🇩	PDAM Tirta Kalwedo semakin didepan
<i>Casefolding</i>	PDAM CIANJUR ADA GAK	pdam cianjur ada gak
<i>Normalization</i>	tolong diupdate apknya untuk unggah foto bukti pembayarannya tidak bisa	tolong diupdate aplikasi nya untuk unggah foto bukti pembayarannya tidak bisa
<i>Tokenizing</i>	tolong tambahkan buat kota samarinda	['tolong', 'tambahkan', 'buat', 'kota', 'samarinda']
<i>Stopword</i>	['so', 'ada', 'aplikasi', 'ini', 'jadi', 'gampang', 'mo', 'ba', 'bayar', 'tagihan', 'air', 'terima', 'kasih', 'neh', 'sangat', 'terbantu']	['aplikasi', 'gampang', 'bayar', 'tagihan', 'air', 'terima', 'kasih', 'terbantu']
<i>Stemming</i>	['bermanfaat']	['manfaat']

Tabel 1 menunjukkan hasil *preprocessing* data, data yang awalnya berjumlah 2.400 menjadi 1.677 data karena telah dilakukan pembersihan data sebelum diproses ketahap selanjutnya. Langkah ini penting dilakukan karena merupakan bagian krusial pada proses analisis teks yang memiliki tujuan agar dapat mempersiapkan serta menyaring data mentah sehingga layak digunakan pada proses klasifikasi [11].

3.3. Pelabelan Sentimen

Pada tahap ini data ulasan akan dilabeli menggunakan pelabelan otomatis *Lexicon Based*, label teks akan dikelompokkan menjadi 3 kategori sentimen yaitu negatif yang ditandai dengan angka 0, positif yang ditandai dengan angka 2, dan netral yang dilabeli dengan angka 1.



Gambar 2. Distribusi Sentimen

Gambar 2 menyajikan distribusi sentimen dari data ulasan yang menunjukkan sebaran jumlah ulasan diantara tiga kategori sentimen. Berdasarkan grafik batang tersebut, sentimen negatif mendominasi data secara signifikan dengan jumlah ulasan mencapai sekitar 1.036. Sementara itu, sentimen positif dengan jumlah sekitar 379 ulasan, dan sentimen netral memiliki sekitar 262 ulasan.


```

... TF-IDF selesai dibuat: (1677, 411)
Jumlah fitur: 410
Idx      admin      adu      air      ... wilayah kota
-----
0        0.0          0.0          0.0 ...      0.0
1        0.0          0.0          0.0 ...      0.0
2        0.0          0.0          0.0 ...      0.0
3        0.0          0.0          0.0 ...      0.0
4        0.0          0.0          0.0 ...      0.0
5        0.0          0.0          0.0 ...      0.0
6        0.0          0.0          0.0 ...      0.0
7        0.0          0.0          0.238435 ... 0.0
8        0.0          0.0          0.0 ...      0.0
9        0.0          0.0          0.0 ...      0.0
...
1676    0.0          0.0          0.0 ...      0.0
    
```

Gambar 6. Hasil TF-IDF

Gambar 6 menunjukkan hasil ekstraksi fitur menggunakan TF-IDF, di mana setiap baris merepresentasikan dokumen ulasan dan setiap kolom merupakan kata yang dijadikan fitur. Nilai pada sel menunjukkan nilai TF-IDF yang mencerminkan seberapa penting suatu kata dalam dokumen tertentu, sementara banyaknya nilai nol menandakan kata bersifat jarang karena sebagian besar kata tidak muncul di setiap dokumen.

3.5. Pembagian Data

Pada tahap ini, data hasil ekstraksi TF-IDF diklasifikasikan ke dalam dua bagian, yakni 80% data latih serta 20% data uji dari keseluruhan 1.677 sampel data. Pembagian ini menghasilkan 1.341 data latih terdiri dari 828 sentimen negatif, 210 netral, dan 303 positif yang dig'unakan untuk melatih model, serta 336 data uji meliputi 208 negatif, 52 netral, dan 76 positif yang digunakan untuk mengevaluasi kinerja dan akurasi prediktif model.

3.6. Implementasi Logistic Regression dan Support Vector Machine

Pada tahap ini, model klasifikasi dibangun dengan memanfaatkan pustaka *scikit-learn* untuk algoritma *Logistic Regression* dan *Support Vector Machine* (SVM). Dalam implementasi menggunakan *scikit-learn*, parameter *default* yang digunakan pada algoritma *Logistic Regression* meliputi parameter $C = 1.0$, $Penalty = l2$, $Solver = lbfgs$, dan $Max\ iter = 600$. Sementara itu, algoritma *Support Vector Machine* menggunakan parameter *default* yang meliputi $C = 1.0$, $Gamma = scale$, dan $Kernel = rbf$.

3.7. Evaluasi

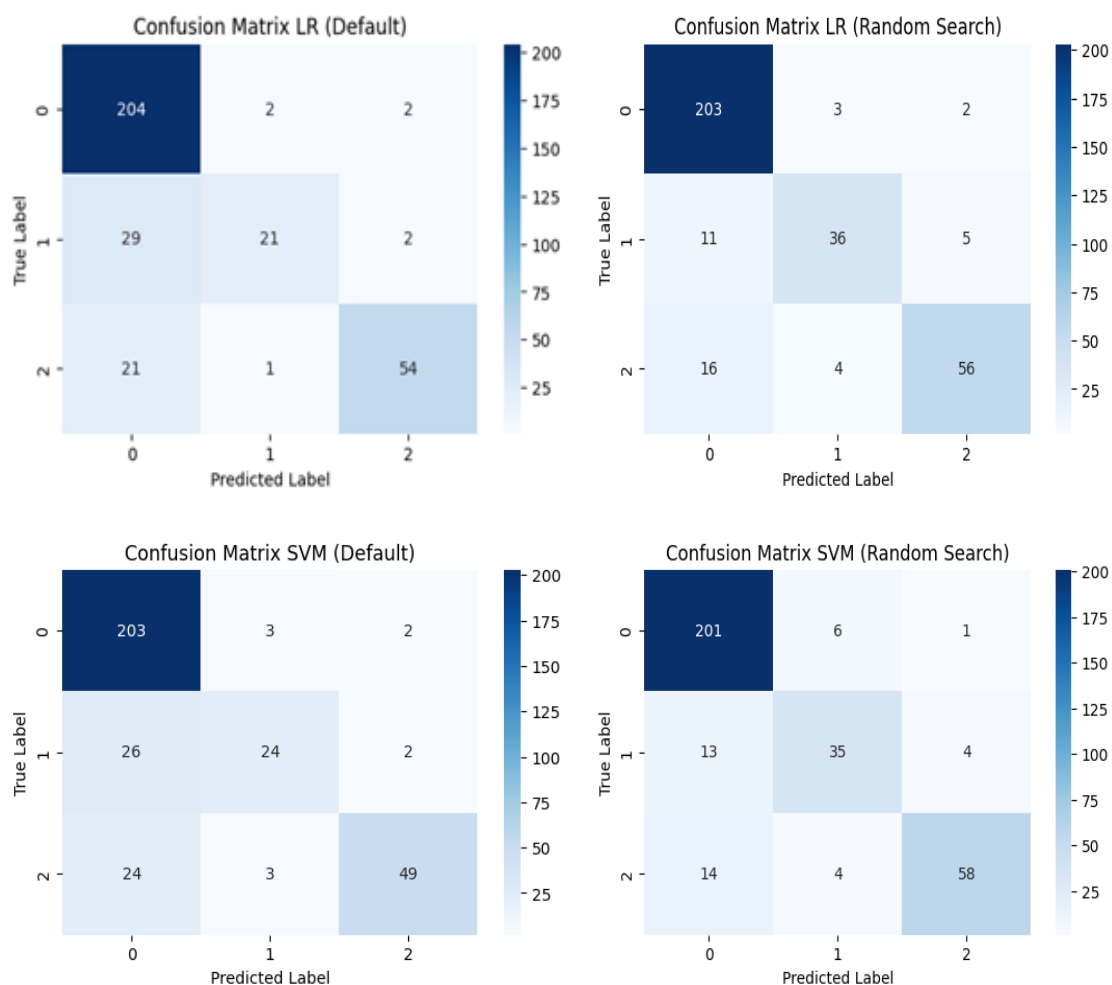
Setelah tahap klasifikasi, tahapan terakhir adalah evaluasi model *machine learning*. Tahap evaluasi dilakukan guna melakukan pengujian terhadap model *machine learning* dengan cara mengukur nilai *accuracy*, *precision*, *recall*, serta *f1-score*. Tahap ini dilaksanakan dengan 2 pengujian yaitu tanpa *Random Search* dan dengan *Random Search* sebagai metode *hyperparameter tuning*. Parameter *Logistic Regression* dan *Support Vector Machine* yang akan ditentukan diantaranya adalah:

Tabel 2. Parameter *Logistic Regression*

Algoritma Klasifikasi	Parameter yang diinputkan	Parameter terbaik
<i>Logistic Regression</i>	$C = \text{loguniform}(0.01, 100)$, $Penalty = ['l2', 'l1']$, $Solver = ['lbfgs', 'newton-cg']$, $Max-iter = [100, 200, 400, 600, 800, 1000]$	$C = 1.7077$, $Penalty = l2$, $Solver = \text{newton-cg}$, $Max-iter = 600$
<i>Support Vector Machine</i>	$C = [0.1, 1, 10, 100]$, $Gamma = [\text{scale}, \text{auto}, 0.01, 0.1, 1]$, $Kernel = [\text{linier}, \text{rbf}, \text{poly}]$	$C = 1$, $Gamma = \text{scale}$, $Kernel = \text{linier}$

Tabel 2 menunjukkan rentang parameter yang diinputkan untuk proses *tuning* pada *Logistic Regression* dan *Support Vector*. Parameter *Logistic Regression* yang ditentukan mencakup *C*, *Penalty*, *Solver*, dan *Max-iter*. *C* yaitu parameter yang mengontrol kekuatan regularisasi, semakin rendah nilai *C* maka kekuatan regularisasi semakin kuat serta semakin tinggi nilai *C* maka kekuatan regularisasinya semakin lemah. *Penalty* yaitu jenis regularisasi. *Solver* yaitu metode atau algoritma untuk melakukan optimasi parameter atau koefisien *Logistic Regression* serta *Max-Iter* yaitu maksimal iterasi yang dilakukan oleh model. Pemilihan parameter ini didasarkan pada studi [5] [22] yang juga memilih parameter ini untuk dituning. Sementara untuk *Support Vector Machine* parameter yang ditentukan mencakup *C*, *Gamma*, dan *Kernel*. *C* mengatur seberapa ketat SVM terhadap kesalahan klasifikasi *C* kecil lebih toleran, *C* besar lebih ketat, *Gamma* mengatur jangkauan pengaruh data terhadap proses pelatihan model, *Kernel* adalah fungsi untuk memetakan data agar dapat dipisahkan, seperti linear untuk data *linear* dan RBF untuk data *non-linear*. Pemilihan parameter ini juga didasarkan pada studi [24] yang memilih parameter ini untuk dituning. Setelah dilakukan hyperparameter tuning pada kedua algoritma tersebut, parameter terbaik yang dihasil oleh model *Logistic Regression* yaitu $C = 1.7077$, $Penalty = 12$, $Solver = newton-cg$, $Max\ iter = 600$, sementara itu parameter terbaik yang dihasilkan oleh model *Support Vector Machine* yaitu $C = 1$, $Gamma = scale$, $Kernel = Linear$

Berikut adalah hasil pengujian tanpa *Random Search* dan dengan *Random Search* yang telah dilakukan :



Gambar 8. *Confusion Matrix* Tanpa dan Dengan *Random Search*

Gambar 8 menampilkan *confusion matrix* hasil klasifikasi sentimen menggunakan algoritma *Logistic Regression* dan *Support Vector Machine (SVM)*, baik sebelum maupun sesudah dilakukan optimasi hyperparameter menggunakan *Random Search*. Pada pengaturan default, baik *Logistic Regression* maupun *Support Vector Machine* masih menunjukkan kesalahan klasifikasi yang cukup dominan pada kelas sentimen netral dengan label 1, yang sering diprediksi sebagai sentimen negatif dengan label 0. Hal ini mengindikasikan bahwa parameter bawaan model belum sepenuhnya optimal dalam membedakan karakteristik kelas netral. Sementara itu, pada pengaturan *Random Search*, distribusi prediksi antar kelas menjadi lebih proporsional, khususnya pada kelas sentimen netral dan kelas sentimen positif dengan label 2. Meskipun kesalahan klasifikasi masih terjadi, *Random Search* membantu model dalam menyesuaikan batas keputusan sehingga prediksi kelas tidak terlalu bias ke kelas negatif. Berikut adalah contoh perhitungan *Accuracy*, *Precision*, *Recall*, dan *F1-Score*.

a. *Accuracy*

Keseluruhan

$$= \frac{204 + 21 + 54}{204 + 2 + 2 + 29 + 21 + 2 + 21 + 1 + 54} \quad (13)$$

$$= \frac{279}{336} = 0,8303 = 83\%$$

b. *Precision*

Kelas 0 (Negatif)

$$= \frac{204}{(204 + (29 + 21))} = \frac{204}{(204 + 50)} = \frac{204}{254} = 0,803 = 83\% \quad (14)$$

Kelas 1 (Netral)

$$= \frac{21}{(21 + (2 + 1))} = \frac{21}{(21 + 3)} = \frac{21}{24} = 0,875 = 88\% \quad (15)$$

Kelas 2 (Positif)

$$= \frac{54}{(54 + (2 + 2))} = \frac{54}{(54 + 4)} = \frac{54}{58} = 0,931 = 91\% \quad (16)$$

Keseluruhan

$$= \frac{98 + 40 + 71}{3} = 69,66 = 70\% \quad (17)$$

c. *Recall*

Kelas 0 (Negatif)

$$= \frac{204}{(204 + (2 + 2))} = \frac{204}{(204 + 4)} = \frac{204}{208} = 0,980 = 98\% \quad (18)$$

Kelas 1 (Netral)

$$= \frac{21}{(21 + (29 + 2))} = \frac{21}{(21 + 31)} = \frac{21}{52} = 0,403 = 40\% \quad (19)$$

Kelas 2 (Positif)

$$= \frac{54}{(54 + (21 + 1))} = \frac{54}{(54 + 22)} = \frac{54}{76} = 0,710 = 71\% \quad (20)$$

Keseluruhan

$$= \frac{98 + 40 + 71}{3} = 69,66 = 70\% \quad (21)$$

d. *F1-Score*

Kelas 0 (Negatif)

$$= 2 \times \frac{98 \times 80}{98 + 80} = 2 \times \frac{7.840}{178} = 2 \times 44,04 = 88,08 = 88\% \quad (22)$$

Kelas 1 (Netral)

$$= 2 \times \frac{40 \times 88}{40 + 88} = 2 \times \frac{3.520}{128} = 2 \times 27,5 = 55\% \quad (23)$$

Kelas 2 (Positif)

$$= 2 \times \frac{71 \times 93}{71 + 93} = 2 \times \frac{6.603}{164} = 2 \times 40,26 = 80,52 = 81\% \quad (24)$$

Keseluruhan

$$= \frac{88 + 55 + 81}{3} = 74,66 = 75\% \quad (25)$$

Persamaan-persamaan diatas adalah contoh perhitungan *Accuracy*, *Presicion*, *Recall*, dan *F1-Score* pada evaluasi Logistiic Regression Secara *default*.

Tabel 3. Evaluasi model

Evaluasi Model	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Tanpa Random Search (LR)	83%	87%	70%	75%
Dengan Random Search (LR)	88%	87%	80%	83%
Tanpa Random Search (SVM)	82%	84%	69%	74%
Dengan Random Search (SVM)	88%	86%	80%	83%

Tabel 3 menunjukkan hasil evaluasi model, *Logistic Regression* memperoleh kinerja yang lebih unggul dibandingkan *Support Vector Machine* pada penelitian ini, baik sebelum maupun sesudah optimasi parameter. *Logistic Regression* tanpa *Random Search* memperoleh akurasi sebesar 83% dan *f1-score* 75%, lebih tinggi dibandingkan *Support Vector Machine* yang hanya mencapai akurasi 82% dan *f1-score* 74%. Setelah dilakukan optimasi parameter menggunakan *Random Search*, kinerja *Logistic Regression* dan *Support Vector Machine* meningkat dengan *accuracy* mencapai 88%, dan *f1-score* mencapai 83%. hasil ini menegaskan bahwa optimasi parameter menggunakan *Random Search* pada algoritma *Logistic Regression* dan *Support Vector Machine* mampu meningkatkan kinerja klasifikasi. Peningkatan tersebut ditunjukkan oleh naiknya nilai *Accuracy*, *Precision*, *Recall*, dan *F1-score* pada kedua algoritma setelah diterapkan *Random Search*, serta perbaikan hasil prediksi model yang tercermin pada *confusion matrix*. Sebelum optimasi, model cenderung lemah dalam memprediksi kelas positif dan netral akibat ketidakseimbangan jumlah data antar kelas. Namun, setelah diterapkan *Random Search*, kemampuan model dalam memprediksi kelas positif dan netral mengalami peningkatan, yang menunjukkan bahwa *Random Search* berkontribusi dalam menghasilkan prediksi kelas sentimen yang lebih seimbang.

4. KESIMPULAN DAN SARAN

Merujuk pada temuan yang telah diselenggarakan sebelumnya, hasil analisis sentimen ulasan aplikasi PDAM Info menunjukkan ulasan negatif mendominasi dengan rata-rata keluhan terkait cakupan wilayah dan metode bayar. Hasil evaluasi model menunjukkan bahwa kinerja model *Logistic Regression* sedikit lebih unggul dibandingkan dengan *Support Vector Machine* baik tanpa *Random Search* maupun dengan *Random Search*. *Logistic Regression* tanpa *Random Search* memperoleh akurasi sebesar 83% dan *f1-score* 75%, lebih tinggi dibandingkan *Support Vector Machine* yang hanya mencapai *accuracy* 82% dan *f1-score* 74%. Setelah dilakukan

optimasi parameter menggunakan *Random Search*, kinerja kedua algoritma meningkat dengan *accuracy* mencapai 88% dan *f1-score* mencapai 83%, kemudian sebelum optimasi *Random Search* model lebih baik dalam memprediksi sentimen negatif dan sulit memprediksi sentimen positif dan netral karena jumlah data antara kelas yang tidak seimbang, setelah optimasi model jauh lebih baik memprediksi sentimen positif dan netral. Hasil ini menegaskan bahwa optimasi parameter berperan penting dalam meningkatkan kinerja klasifikasi pada *Logistic Regression* dan *Support Vector Machine* dan dapat menghasilkan prediksi kelas yang lebih seimbang.

Temuan ini menyarankan pada pengembang aplikasi PDAM Info dari hasil analisis sentimen, rata-rata komentar negatif atau keluhan dari pengguna adalah untuk menambahkan kerja sama dari cabang PDAM di seluruh Indonesia dan menambahkan metode pembayaran online, agar pengguna lebih mudah mengakses layanan airnya hanya dalam satu aplikasi saja. Kemudian temuan ini juga masih terdapat kekurangan karena metode pelabelan yang digunakan berbasis kamus *Lexicon*, sehingga terdapat beberapa label sentimen yang tidak tepat, untuk penelitian selanjutnya diharapkan menggunakan metode atau model yang dapat menangkap bahasa Indonesia dengan lebih baik seperti *IndoBERT*, serta memperbaiki tahap prapemrosesan data, khususnya penghapusan *stopword* dan mempertimbangkan perbandingan metode *hyperparameter tuning* lainnya, seperti *Bayesian Optimization* untuk eksplorasi peningkatan kinerja model lebih lanjut.

DAFTAR PUSTAKA

- [1] T. Tri Indriyani, “Bebas Ribet Bayar dan Urus Tagihan Air, Cukup Pakai PDAM Info!,” PDAM Info, 2024. [Online]. Tersedia: <https://pdaminfo.pdampintar.id/blog/lainnya/bebas-ribet-bayar-dan-urus-tagihan-air-dengan-pdam-info>
- [2] E. Indrayuni and A. Nurhadi, “Analisis Sentimen Aplikasi Tiktok Shop Seller Center Menggunakan Naive Bayes, SVM dan Logistic Regression,” *Inti Nusa Mandiri*, vol. 20, pp. 26–34, 2025, doi: 10.33480/inti.v20i1.6851.
- [3] K. Khotimah, M. Martanto, A. R. Dikananda, and A. Rifa’i, “Analisis Sentimen Ulasan Aplikasi Pintu Di Google Play Store Menggunakan Algoritma Naive Bayes,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 1, pp. 963, 963-971, 2025, doi: 10.23960/jitet.v13i1.5789.
- [4] F. Zakwan, “Perbandingan Algoritma Klasifikasi Naive Bayes,” Skripsi, Universitas Islam Negeri Ar-Raniry Banda Aceh, Bandar Lampung, 2023.
- [5] I. R. Ainunnisa and S. Sulastri, “Analisis Sentimen Aplikasi Tiktok dengan Metode Support Vector Machine (SVM), Logistic Regression dan Naive Bayes,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 6, no. 3, pp. 423–430, 2023, doi: 10.32493/jtsi.v6i3.31076.
- [6] A. N. Syafia, M. F. Hidayattullah, and W. Sutеды, “Studi Komparasi Algoritma SVM dan Random Forest Pada Analisis Sentimen Komentar Youtube BTS,” vol. 8, no. 3, pp. 207-212, 2023, doi: 10.30591/jpit.v8i3.5064.
- [7] A. Wahid Mu’ammар, K. Adi Nugroho, T. Safitri, and F. Setyo Utomo, “Optimasi Logistic Regression dan Random Forest untuk Deteksi Berita Hoax Berbasis TF-IDF,” *Jurnal Pendidikan dan Teknologi Indonesia (JPTI)*, vol. 4, no. 8, pp. 381–392, 2024, doi: 10.52436/1.jpti.602.
- [8] J. E. Br Sinulingga and H. C. K. Sitorus, “Analisis Sentimen Opini Masyarakat terhadap Film Horor Indonesia Menggunakan Metode SVM dan TF-IDF,” *Jurnal Manajemen Informatika (JAMIKA)*, vol. 14, no. 1, pp. 42–53, 2024, doi: 10.34010/jamika.v14i1.11946.
- [9] T. A. E. Putri, T. Widiharih, and R. Santoso, “Penerapan Tuning Hyperparameter Randomsearchcv Pada Adaptive Boosting Untuk Prediksi Kelangsungan Hidup Pasien Gagal Jantung,” *Jurnal Gaussian*, vol. 11, no. 3, pp. 397–406, 2023, doi: 10.14710/j.gauss.11.3.397-406.
- [10] A. D. Rachmatsyah, T. Sugihartono, and K. Irfan, “Perbandingan Teknik Optimasi Grid Search dan Randomized Search dalam Meningkatkan Akurasi Metode Klasifikasi SVM

- Pada Sentimen Ulasan Pengguna Aplikasi JKN Mobile,” *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 8, no. 1, pp. 13–22, 2025, doi: 10.36080/skanika.v8i1.3328.
- [11] S. W. Ritonga, Y., M. Fikry, and E. P. Cynthia, “Klasifikasi Sentimen Masyarakat di Twitter terhadap Ganjar Pranowo dengan Metode Naïve Bayes Classifier,” *Building of Informatics, Technology and Science (BITS)*, vol. 5, no. 1, 2023, doi: 10.47065/bits.v5i1.3535.
- [12] H. Putri, E. Rensa, N. Nur, and F. Wajidi, “Analisis Sentimen Komentar YouTube terhadap Ganjar Pranowo Pasca Pemilu 2024 dengan Pendekatan Long Short-Term Memory,” *Journal of Computer and Information System (J-CIS)*, vol. 7, no. 2, pp. 89–98, 2024, doi: 10.31605/jcis.v7i2.
- [13] I. Tri Julianto, “Analisis Sentimen Terhadap Sistem Informasi Akademik Mahasiswa Institut Teknologi Garut,” *Jurnal Algoritma Institut Teknologi Garut*, vol. 19, pp. 458–465, 2022, doi: 10.33364/algoritma/v.19-1.1112.
- [14] C. E. Joergensen Munthe, N. Astuti Hasibuan, and H. Hutabarat, “RESOLUSI: Rekayasa Teknik Informatika dan Informasi Penerapan Algoritma Text Mining Dan TF-RF Dalam Menentukan Promo Produk Pada Marketplace,” *RESOLUSI: Rekayasa Teknik Informatika dan Informasi*, vol. 2, no. 3, pp. 110–115, 2022, doi: 10.30865/resolusi.v2i3.309.
- [15] I. Gusti, A. Ngurah, R. Semadi, M. Samsudin, and K. Dharmendra, “Perbandingan Metode Seleksi Fitur Pada Analisis Sentimen (Studi Kasus Opini PILKADA DKI 2017),” *Journal of informatics*, vol. 8, no. 1, pp. 11–18, 2023, doi: 10.51211/itbi.v8i1.2408.
- [16] R. Yusuf, K. Bahumatra, N. Komaria, E. A. Aqma, and L. Cahyani, “Analisis Sentimen Terhadap Aplikasi Google Meet Berdasarkan Komentar Pengguna Menggunakan Metode Logistic Regression,” *Jurnal Ilmiah Edutic: Pendidikan dan Informatika*, vol. 11, no. 1, pp. 53–64, 2024, doi: 10.21107/edutic.v11i1.28113.
- [17] B. B. Tangkere, “Analisis Performa Logistic Regression dan Support Vector Classification untuk Klasifikasi Email Phising,” *Jurnal Ekonomi Manajemen Sistem Informasi (JEMSI)*, vol. 5, pp. 442–450, 2024, doi: 10.38035/jemsi.v5i4.
- [18] R. Prabowo, H. Sujaini, and T. Rismawan, “Analisis Sentimen Pengguna Twitter Terhadap Kasus COVID-19 di Indonesia Menggunakan Metode Regresi Logistik Multinomial,” *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 11, no. 2, pp. 366–370, 2023, doi: 10.26418/justin.v11i2.57449.
- [19] S. A. Utirahman, A. Mulawati, and M. Pratama, “KLIK: Kajian Ilmiah Informatika dan Komputer Analisis Perbandingan KNN, SVM, Decision Tree dan Regresi Logistik Untuk Klasifikasi Obesitas Multi Kelas,” in *Media Online*, 2024, pp. 3137–3146. doi: 10.30865/klik.v4i6.1871.
- [20] J. Ipmawati, S. Saifulloh, and K. Kusnawi, “Analisis Sentimen Tempat Wisata Berdasarkan Ulasan pada Google Maps Menggunakan Algoritma Support Vector Machine,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 1, pp. 247–256, 2024, doi: 10.57152/malcom.v4i1.1066.
- [21] N. C. Ramadani, “Analisis Sentimen Untuk Mengukur Ulasan Pengguna Aplikasi Mobile Legend Menggunakan Algoritma Naive Bayes, SVM, Random Fores, Decision Tree, dan Logistic Regression,” *JSI: Jurnal Sistem Informasi (E-Journal)*, vol. 16, no. 1, pp. 123–138, 2024, doi: <https://doi.org/10.18495/jsi.v16i1.152>.
- [22] A. K. Sari, *et al.*, “Analisis Sentimen Twitter Menggunakan Machine Learning untuk Identifikasi Konten Negatif,” *Adopsi Teknologi dan Sistem Informasi (ATASI)*, vol. 3, no. 1, pp. 64–73, 2024, doi: 10.30872/atasi.v3i1.1373.
- [23] A. Tirta, P. Subandono, and D. Ariatmanto, “Optimalisasi Seleksi Fitur dalam Analisis Sentimen Bank Saqu: Studi Perbandingan SVM dan Random Forest Menggunakan Information Gain dan Chi-Square,” *Sistemasi: Jurnal Sistem Informasi*, vol. 14, pp. 1205–1219, 2025, doi: 10.32520/stmsi.v14i3.5106.

- [24] F. Dewi, N. C. H. Wibowo, M. R. Handayani, and K. Umam, "Evaluasi Hyperparamter Tuning Pada Support Vector Machine (SVM) Dalam Klasifikasi Ulasan Hotel Di Tripadvisor," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 3, pp. 2584–2593, 2025, doi: 10.29100/jipi.v10i3.7774.