IJAI (Indonesian Journal of Applied Informatics)
Vol. 9 No. 2 Tahun 2025 pISSN: 2548-3846, eISSN: 2598-5981

Penggunaan Metode *Logistic Regression* Untuk Analisis Sentimen Pembangunan Ibu Kota Nusantara Pada Media Sosial

Adi Julia Saputra[1]*, Sentot Achmadi[1], Karina Auliasari[1]
[1]Program Studi S1 Teknik Informatika, Institut Teknologi Nasional Malang, Indonesia
*Email: adiadijulia@gmail.com

Info Artikel

Kata Kunci :
analisis sentimen, regresi logistik, pembelajaran mesin

Keywords :
sentiment analysist, logistic regression, machine learning

Tanggal Artikel
Dikirim : 25 November 2024
Direvisi : 16 Januari 2025
Diterima : 10 Februari 2025

Abstrak

Indonesia, sebagai negara kepulauan terbesar, menghadapi tantangan pemerataan pembangunan, salah satunya dengan memindahkan ibu kota ke Ibu Kota Nusantara (IKN) di Kalimantan Timur. Proyek ini bertujuan untuk mengatasi masalah di Jakarta, namun ada kekhawatiran mengenai dampaknya terhadap ekonomi dan politik. Twitter menjadi *platform* utama untuk menganalisis opini masyarakat mengenai pemindahan ibu kota. Penelitian ini bertujuan untuk menganalisis sentimen masyarakat terhadap pembangunan Ibu Kota Nusantara (IKN) menggunakan metode *Logistic Regression*, yang mengklasifikasikan opini menjadi positif, negatif, atau netral. Dengan menganalisis *tweet* yang terkait, penelitian ini menemukan bahwa mayoritas sentimen adalah negatif. Model yang digunakan berhasil mengklasifikasikan sentimen dengan akurasi yang baik pada data pelatihan, meskipun hasil pada data pengujian menunjukkan tantangan lebih lanjut. Temuan ini memberikan wawasan tentang bagaimana masyarakat memandang rencana pemindahan ibu kota dan dapat membantu dalam pengambilan keputusan kebijakan.

Abstract

Indonesia, as the largest archipelagic country, faces the challenge of equitable development, one of which is the relocation of the capital city to Ibu Kota Nusantara (IKN) in East Kalimantan. This project aims to address issues in Jakarta, but there are concerns about its impact on the economy and politics. Twitter has become the main platform for analyzing public opinion regarding the capital city relocation. This study aims to analyze public sentiment toward Ibu Kota Nusantara (IKN) using the Logistic Regression method, which classifies opinions into positive, negative, or neutral. By analyzing related tweets, the study found that the majority of sentiments were negative. The model used was able to classify sentiments with good accuracy on training data, although the results on testing data showed further challenges. These findings provide insights into how the public perceives the capital relocation plan and can assist in policy decision-making.

1. PENDAHULUAN

Indonesia, sebagai negara kepulauan terbesar dengan populasi yang padat, menghadapi tantangan besar dalam pemerataan pembangunan. Salah satu langkah besar yang diambil oleh pemerintah adalah pemindahan ibu kota negara ke Ibu Kota Nusantara (IKN) di Kalimantan Timur. Kebijakan ini bertujuan untuk mengatasi masalah kemacetan, polusi, dan ketimpangan pembangunan di Jakarta, namun menimbulkan berbagai kekhawatiran di kalangan masyarakat terkait stabilitas ekonomi, sosial, dan politik proyek ini [1].

Salah satu cara untuk memahami pandangan masyarakat adalah dengan menganalisis sentimen yang muncul di media sosial, khususnya Twitter, yang menjadi *platform* populer untuk berbagi opini publik. Analisis sentimen ini dapat memberikan gambaran yang lebih jelas tentang bagaimana masyarakat merespon isu pemindahan ibu kota negara. Dengan menggunakan algoritma *machine learning* seperti *Logistic Regression*, analisis ini bertujuan untuk mengkategorikan sentimen publik menjadi tiga kategori: positif, negatif, dan netral [2].

Beberapa metode yang umum digunakan dalam analisis sentimen adalah *Logistic Regression*, *Naïve Bayes*, dan *Support Vector Machine* (SVM), masing-masing dengan kelebihan dan kekurangan. *Logistic Regression* unggul dalam akurasi, interpretasi hasil, dan efisiensi dibandingkan *Naïve Bayes* dan SVM. Studi yang dilakukan oleh Isna dan Sulastri (2023) menunjukkan *Logistic Regression* mencapai akurasi 84%, lebih tinggi dibandingkan SVM (82%) dan *Naïve Bayes* (79%) dalam analisis sentimen ulasan aplikasi TikTok. Temuan ini menguatkan relevansi *Logistic Regression* untuk data kompleks seperti opini publik terhadap pemindahan ibu kota [3].

Mempertimbangkan pro dan kontra terkait pemindahan dan pembangunan ibu kota baru, penelitian ini ditujukan untuk melakukan analisis terhadap sentimen publik mengenai pembangunan Ibu Kota Nusantara (IKN). Untuk mencapai memenuhi tujuan tersebut, pada studi penelitian ini menggunakan metode *Logistic Regression*, sebuah teknik analisis statistik yang efektif untuk memprediksi sebuah sentimen berdasarkan data yang ada. Dengan menganalisis opini yang telah dikumpulkan, penelitian ini berupaya memberikan gambaran yang lebih jelas tentang persepsi publik terkait pemindahan dan pembangunan ibu kota baru.

2. METODE PENELITIAN

2.1 *Logistic regression*

Logistic Regression merupakan metode atau model pembelajaran mesin (*machine learning*) yang sering digunakan untuk melakukan proses klasifikasi pada data biner, seperti menentukan apakah sebuah *email* termasuk spam atau tidak. Metode ini menggunakan fungsi logistik (sigmoid) untuk mengubah kombinasi linier menjadi nilai probabilitas antara 0 dan 1. Selain itu, Logistic Regression juga digunakan untuk melakukan analisis hubungan antara variabel dependen dengan variabel independen, sehingga dapat dimungkinkan untuk mengklasifikasikan sebuah kalimat menjadi label positif, negatif, atau netral berdasarkan vektor frekuensi dari dataset yang digunakan untuk training dan testing [4].

Fungsi sigmoid merupakan sebuah fungsi didalam matematika yang dipergunakan untuk memetakan nilai prediksi menjadi probabilitas. Fungsi ini memetakan sebuah nilai apapun menjadi nilai yang berada dalam rentang hasil 0 hingga 1. Nilai hasil *logistic regression* harus berada diantara 0 dan 1, sehingga menghasilkan kurva berbentuk "S", yang disebut dengan fungsi sigmoid atau bias disebut fungsi logistik. Dalam *logistic regression*, kita menggunakan konsep nilai ambang, yang mendefinisikan probabilitas antara 0 atau 1. Jika nilai di atas ambang batas, hasilnya cenderung 1, dan jika di bawah ambang, cenderung 0 [5].

- Fungsi *Sigmoid* :

$$sigmoid(x) = \frac{1}{1+e^{-x}} \quad (1)$$

- Persamaan *g(X)* dalam bentuk *sigmoid* :

$$g(X) = sigmoid\ (\alpha + \beta X) \quad (2)$$

Keterangan :
α : Nilai Konstanta
β : Koefisienregresi
X : Variabel Independen

Penjelasan :

Seperti yang ditunjukkan di atas, fungsi sigmoid mengonversi data variabel kontinu menjadi probabilitas, yaitu di antara 0 dan 1.

a. *Sigmoid(x)* mendekati 1, ketika *x* mendekati ∞.
Artinya, ketika nilai *x* menjadi semakin besar (*x* mendekati tak terhingga positif), nilai *sigmoid(x)* semakin dekat ke 1. Sehingga:

$$sigmoid(x) = \frac{1}{1 + e^{-x}} = \frac{1}{1 + 0} = 1$$

b. *Sigmoid(x)* mendekati 0, ketika *x* mendekati -∞.
Artinya, ketika nilai *x* menjadi semakin kecil (negatif), nilai *sigmoid(x)* semakin dekat ke 0. Sehingga:

$$sigmoid(x) = \frac{1}{1 + e^{-x}} = \frac{1}{1 + \infty} = 0$$

c. *Sigmoid(x)* selalu diantara 0 dan 1 [6].

2.2 Analisis Sentimen

Analisis sentimen atau *sentiment analysist* merupakan sebuah proses komputasi yang melibatkan analisis teks digital yang berada di sebuah aplikasi untuk menentukan apakah kata tersebut atau kalimat yang disampaikan memiliki makna emosional tertentu. Proses ini bertujuan untuk mengidentifikasi dan mengelompokkan emosi menjadi kategori positif, negatif, atau netral. Dalam melakukan analisis sentimen, berbagai tahapan dilalui mulai dari pemrosesan teks hingga pengklasifikasian sentimen [7].

2.3 *Text Mining*

Text Mining merupakan sebuah teknik yang digunakan untuk menggali data guna memenuhi kebutuhan informasi melalui metode seperti *machine learning*, *data mining*, *natural language processing*, dan pencarian informasi. Proses ini mencakup. Proses ini mencakup praproses dokumen, pengkategorian teks, dan ekstraksi informasi untuk mengidentifikasi pola dari data teks tidak terstruktur. Berbeda dengan *data mining* yang bekerja pada data terstruktur, *Text Mining* fokus pada bahasa alami yang tidak terstruktur. Tahap utama dalam *Text Mining* adalah praproses teks [8].

2.4 *Preprocessing* Data

Text preprocessing adalah tahap untuk mengolah data berupa data teks, kata, atau kalimat yang sama sekali tidak terstruktur menjadi sebuah data yang terstruktur, sehingga mempermudah untuk melakukan proses komputasi [9]. Proses ini diperlukan untuk membantu algoritma dalam melakukan pembelajaran terhadap model [10]. Beberapa tahap yang dilakukan dalam *preprocessing data* adalah sebagai berikut :

a. *Handling Missing Value*, adalah proses yang digunakan untuk mengatasi nilai-nilai yang hilang dalam sebuah *dataset*. Nilai yang hilang tersebut bisa terjadi seperti kesalahan pengumpulan data atau kegagalan system [11].
b. Cek Duplikasi Data, merupakan proses yang digunakan untuk menghapus data yang sama atau ganda ketika melakukan pengambilan data [12].
c. *Casefolding*, merupakan fungsi yang digunakan untuk mengubah seluruh isi teks pada dokumen menjadi huruf kecil atau *lowcase*, karena tidak semua teks konsisten terhadap huruf kapital [13].
d. *Cleansing*, merupakan proses yang digunakan untuk membersihkan dokumen dan menyeleksi kata yang tidak diperlukan seperti *url*, *emoticon*, *hastag*, *mention* dan lain-lain [14].
e. Normalisasi, adalah proses yang digunakan untuk mengubah kata bahasa Indonesia yang bersifat tidak formal, kata tidak baku, kata disingkat atau kata diperpanjang dirubah menjadi bentuk kata baku yang benar [15].
f. *Tokenizing*, merupakan proses yang dilakukan untuk melakukan pemotongan sebuah dokumen menjadi bagian-bagian yang disebut sebagai token. Seperti mengubah dari sebuah data dalam bentuk kalimat menjadi sebuah kata [16].
g. *Stopwords Removal*, adalah proses penghilangan kata-kata yang tidak memiliki kontribusi pada isi dokumen. Kata tersebut dihilangkan karena memberi proses yang tidak baik dalam proses *text mining* [17].
h. *Stemming*, adalah proses yang berfungsi untuk membuat suatu kata menjadi kata dasar, dengan menghilangkan imbuhan yang ada pada kata tersebut [18].

2.5 Parameter *Logistic Regression*

Parameter yang digunakan untuk mengatur proses pelatihan dan menentukan cara model mempelajari data dibagi menjadi dua jenis, yaitu :

a. *Solver Liblinear*, pada *Logistic Regression* fungsinya adalah untuk mengoptimalkan parameter model, khususnya dalam kasus klasifikasi biner dan kecilnya jumlah data atau fitur. *Solver* ini bekerja dengan metode optimisasi berbasis gradien dan cocok untuk dataset dengan jumlah sampel yang relatif kecil hingga menengah [19].
b. *Class Weight Balanced*, parameter *class weight* digunakan untuk membantu mengatasi masalah dengan *dataset* yang tidak seimbang dengan memberikan bobot yang berbeda pada kelas berdasarkan frekuensinya. Pada data yang tidak seimbang, model mungkin lebih mengutamakan kelas mayoritas, sehingga *class weight* dapat digunakan untuk memberikan perhatian lebih pada kelas minoritas [20].

2.6 Teknik Validasi *Train Test Split*

Metode *train test split* digunakan untuk membagi data menjadi data pelatihan dan data pengujian. Data dibagi menjadi fitur (X) dan label (y), kemudian dibagi lebih lanjut menjadi X_*train*, X_*test*, y_*train*, dan y_*test*. X_*train* dan y_*train* digunakan untuk melatih model, sedangkan X_*test* dan y_*test* digunakan untuk menguji apakah model dapat memprediksi label dengan benar. Secara umum, 20% data digunakan untuk pengujian dan 80% untuk pelatihan [21].

2.7 Pembobotan TF-IDF

Teknik pembobotan TF-IDF atau teknik pembobotan kata adalah langkah untuk mengubah sebuah kata menjadi bentuk numerik atau vektor. Sementara itu, TF (*term frequency*) digunakan untuk menentukan suatu frekuensi didalam sebuah kata pada dokumen atau *dataset*. Pembobotan pada setiap kata dalam *F1 Score* adalah perbandingan rata-rata antara presisi dan *recall* yang telah dilakukan pembobotan kata [22].

a. Term Frequency (TF)

Term Fequency (TF) adalah proses metode untuk melakukan perhitungan terhadap frekuensi kemunculan sebuah kata pada dokumen. Jika sebuah kata memiliki rentang kemunculan lebih sering, maka akan semakin besar pula bobot yang akan diberikan. Perhitungan TF dapat dirumuskan sebagaimana ditunjukkan dalam persamaan (3) [23].

$$\mathrm{TF} = 1 + \log\left(F_{t,d}\right) \ , \ F_{t,d} > 0 \qquad (3)$$

Keterangan :

TF	=	*Term Frequency*
$F_{t,d}$	=	*Frequency Term* pada dokumen

b. Inverse Document Frequency (IDF)

IDF merupakan perhitungan yang menggambarkan tingkat penyebaran sebuah kata atau data dalam suatu kumplan dokumen. IDF menunjukkan seberapa sering kata tersebut muncul dalam keseluruhan dokumen. Rumus perhitungan IDF dapat dirumuskan sebagaimana ditunjukkan dalam persamaan (4) [24].

$$\mathrm{IDF} = \log\left({}^{D}/_{df}\right) \qquad (4)$$

Keterangan:

IDF	=	*Inverse Document Frequency*
D	=	Jumlah Dokumen Dalam Analisis
df	=	Jumlah Dokumen yang Mengandung Kata

c. TF-IDF

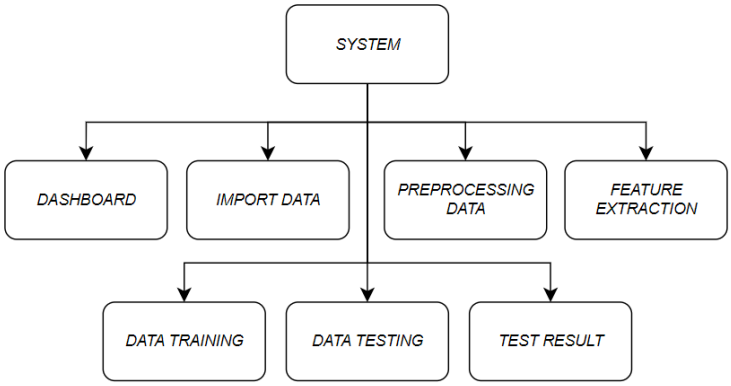
Setelah menentukan TF dan IDF, dilakukan perhitungan pada nilai TF-IDF yaitu dengan cara mengkalikan hasil TF dan IDF. Perkalian ini menunjukkan seberapa penting suatu kata dalam dokumen tertentu dibandingkan dengan keseluruhan koleksi dokumen. Rumus perhitungan TF-IDF dapat dirumuskan sebagaimana ditunjukkan dalam persamaan (5).

$$W = tf \text{ x } idf \quad (5)$$

Keterangan :
W = Bobot Perhitungan TF-IDF
tf = Hasil nilai TF
idf = Hasil nilai IDF

2.8 Diagram Struktur Menu

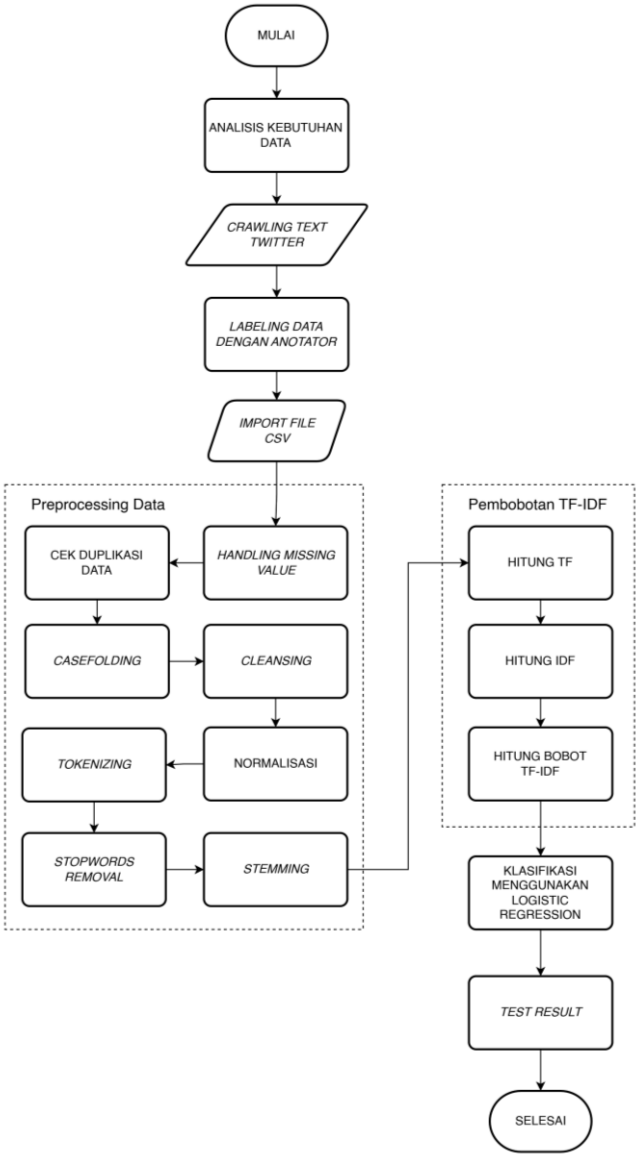
Struktur menu dalam sebuah aplikasi dirancang untuk merepresentasikan alur kerja dan keterkaitan antarhalaman, sehingga pengguna dapat dengan mudah memahami fungsionalitas sistem. Menu ini berperan sebagai kerangka navigasi yang menghubungkan setiap modul, mulai dari halaman *dashboard*, yang memberikan informasi umum tentang status dan aktivitas sistem, hingga ke bagian yang lebih spesifik. Dalam sistem ini, menu mencakup fitur *import data* yang digunakan untuk melakukan *import data* kedalam sistem, *preprocessing data* yang digunakan untuk membersihkan dan mempersiapkan data mentah sebelum analisis. Halaman *data training*, yang menjadi tempat model dilatih dengan data yang telah diolah. Halaman *data testing*, di mana model diuji untuk mengevaluasi kinerjanya. Halaman *test result*, yang menyajikan hasil analisis dalam bentuk laporan performa. Setiap bagian menu ini dirancang untuk mendukung proses kerja yang sistematis dan memastikan bahwa pengguna dapat mengakses setiap modul dengan mudah dan efisien. Desain struktur menu pada sistem dapat dilihat pada Gambar 1 berikut.



Gambar 1 Rancangan Struktur Menu

2.9 *Flowchart* Alur Penelitian

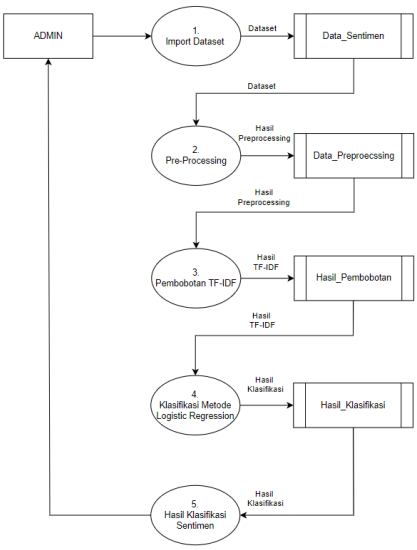
Sistem yang dikembangkan merupakan sebuah aplikasi dengan sebuah konsep sistem khusus yang dirancang untuk menganalisis sentimen terkait pembangunan Ibu Kota Nusantara pada media sosial aplikasi Twitter dengan menggunakan model atau metode *Logistic Regression* untuk mengklasifikasikan *tweet*. Proses dimulai dengan analisis kebutuhan data untuk menentukan jenis dan jumlah data yang diperlukan. Data kemudian dikumpulkan melalui proses *crawling* dari Twitter, diikuti oleh tahap *labeling* oleh anotator untuk memberikan kategori pada setiap data sesuai tujuan penelitian. Data yang telah diberi label diimpor ke dalam *file CSV* untuk mempermudah pengolahan. Selanjutnya, dilakukan *preprocessing data* yang mencakup cek duplikasi, penanganan *missing value*, *case folding*, pembersihan data dari karakter *noise*, tokenisasi, normalisasi, penghapusan *stopwords*, dan *stemming* untuk mendapatkan teks yang bersih dan seragam. Setelah *preprocessing*, dilakukan pembobotan dengan metode TF-IDF untuk memberikan bobot pada kata-kata penting berdasarkan frekuensi dan distribusinya dalam dokumen. Kata-kata dihitung nilai TF, IDF, dan bobot TF-IDF-nya. Data yang telah dibobotkan kemudian diklasifikasikan menggunakan *Logistic Regression* untuk memprediksi kategori atau kelasnya. Hasil klasifikasi diuji akurasinya untuk menilai performa model sebelum penelitian dinyatakan selesai. Proses ini mencerminkan alur terintegrasi dari pengumpulan hingga analisis data untuk mencapai tujuan klasifikasi. Gambaran umum alur penelitian dari sistem yang akan dirancang dan sudah digambarkan dapat dilihat pada Gambar 2.



Gambar 2 *Flowchart* Alur Penelitian

2.10 DFD Level 0

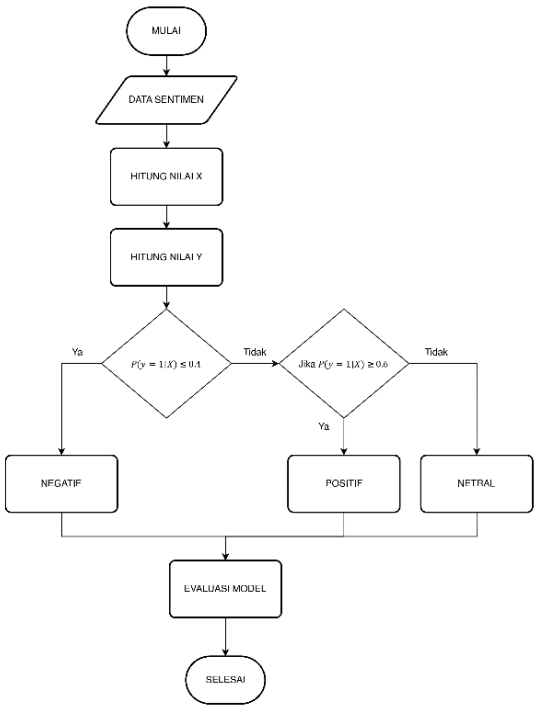
Sistem yang akan dikembangkan adalah aplikasi analisis sentimen berbasis *website* yang dirancang untuk menganalisis sentimen terkait pembangunan Ibu Kota Nusantara dengan menggunakan metode *Logistic Regression* untuk mengklasifikasikan *tweet* dari media sosial Twitter. DFD level 0 menggambarkan alur proses secara umum dalam sistem analisis sentimen, dimulai dari entitas eksternal, yaitu admin, yang berperan sebagai pengguna untuk menginisiasi sistem. Proses pertama adalah mengimpor *dataset*, yang berisi data mentah seperti *tweet* atau komentar yang akan dianalisis sentimennya. Setelah itu, *dataset* melalui tahap *preprocessing* untuk membersihkan dan mengolah data agar siap dianalisis. Proses ini mencakup tokenisasi untuk membagi teks menjadi kata-kata individual, penghapusan *stopwords* untuk menghilangkan kata-kata yang tidak bermakna, *stemming* untuk mengubah kata menjadi bentuk dasarnya, dan normalisasi untuk menyamakan bentuk kata. Setelah *preprocessing*, dilakukan pembobotan menggunakan metode TF-IDF untuk menentukan bobot kata-kata penting dalam dokumen. Data yang telah dibobotkan kemudian diklasifikasikan menggunakan algoritma *Logistic Regression*, yang memprediksi sentimen dari data sebagai positif, negatif, atau netral. Hasil akhirnya adalah klasifikasi sentimen dari data yang dianalisis, memberikan gambaran mengenai sentimen yang terkandung dalam *dataset* tersebut. Hasil dapat dilihat pada keterangan Gambar 3.



Gambar 3 DFD Level 0

2.11 *Flowchart* Metode *Logistic Regression*

Metode *Logistic Regression* digunakan dalam sistem ini untuk mengklasifikasikan sentimen *tweet* terkait pembangunan Ibu Kota Nusantara. Metode *Logistic Regression* adalah teknik statistik untuk memodelkan hubungan variabel independen (fitur teks *tweet*) dan variabel dependen biner atau multinomial (kategori sentimen seperti positif, negatif, atau netral). Dalam alurnya, teks *tweet* diolah menjadi representasi numerik menggunakan pembobotan TF-IDF, yang membantu model memahami distribusi kata dalam *tweet*. Setelah itu, algoritma menghitung nilai Y, yaitu probabilitas suatu data termasuk dalam kelas tertentu, seperti positif. Proses klasifikasi dilakukan berdasarkan nilai probabilitas. Jika probabilitas sentimen positif lebih besar atau sama dengan 0.6, maka data diklasifikasikan sebagai positif. Jika probabilitasnya kurang atau sama dengan 0.4, maka diklasifikasikan sebagai negatif. Sedangkan nilai yang berada di antara kedua batas tersebut dikategorikan sebagai netral. Setelah klasifikasi selesai, model dievaluasi untuk mengukur performanya menggunakan metrik seperti akurasi, *precision*, *recall*, dan *F1-score*. Proses diakhiri dengan menampilkan hasil evaluasi yang menunjukkan kinerja model. Gambaran umum *flowchart* metode *logistic regression* dari sistem yang telah dibuat dapat dilihat pada Gambar 4.

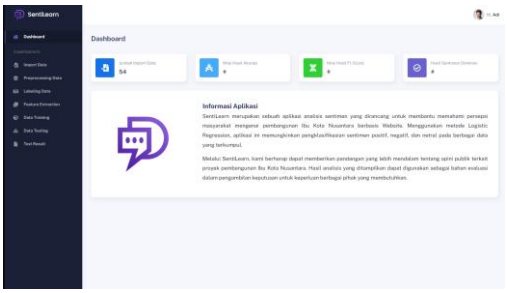


Gambar 4 *Flowchart* Metode *Logistic Regression*

3. HASIL DAN PEMBAHASAN

3.1 Halaman *Dashboard*

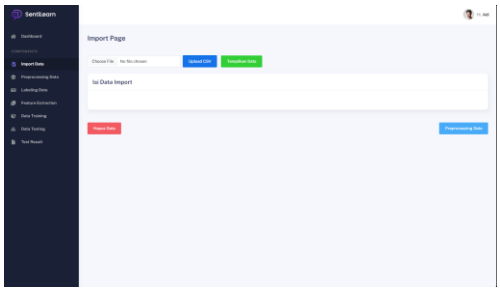
Halaman *Dashboard* yang dirancang untuk memberikan gambaran umum dan visualisasi data secara menyeluruh. Halaman ini memungkinkan pengguna untuk memantau berbagai metrik dan statistik secara langsung, memberikan pemahaman cepat tentang data yang dianalisis serta mempermudah pengambilan keputusan berbasis data. Hasil halaman *dashboard* dapat dilihat pada keterangan Gambar 5.



Gambar 5 Halaman *Dashboard*

3.2 Halaman *Import* File

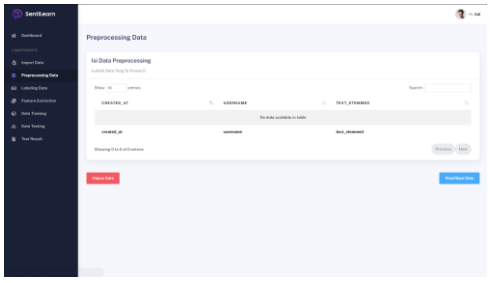
Halaman *Import Data* halaman yang ditujukan untuk melakukan pengunggahan *file* CSV. Fitur ini memungkinkan pengguna untuk menambahkan data baru secara cepat dan efisien, memperkaya sumber informasi yang akan dianalisis tanpa perlu melalui proses manual yang rumit. Dengan adanya fitur ini, data dapat langsung tersimpan dan siap untuk diproses lebih lanjut dalam aplikasi. Hasil halaman *import file* bisa dilihat pada Gambar 6.



Gambar 6 Halaman *Import* File

3.3 Halaman *Preprocessing Data*

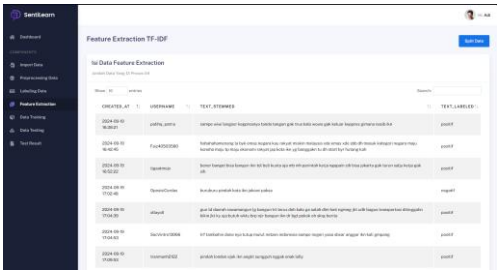
Halaman *preprocessing data* ditujukan untuk menampilkan proses pengolahan awal data untuk keperluan analisis sentiment. Pada tahap ini, data melewati proses pembersihan teks, normalisasi, tokenisasi dan beberapa tahap lainnya sehingga kualitasnya lebih optimal untuk dianalisis. Halaman ini mempermudah pengguna dalam menyiapkan data secara otomatis dan memastikan bahwa data siap digunakan oleh model analisis sentimen. Hasil halaman *Preprocessing Data* bisa dilihat pada Gambar 7.



Gambar 7 Halaman *Preprocessing Data*

3.4 Halaman *Feature Extraction*

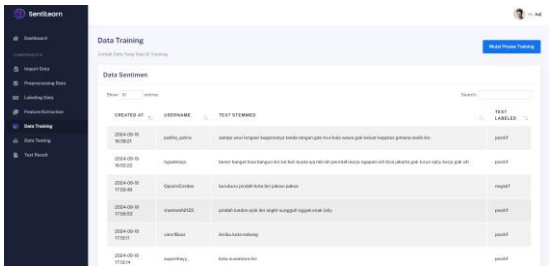
Halaman *Feature Extraction* (TF-IDF) adalah halaman yang menyediakan proses ekstraksi fitur dari data teks yang menggunakan teknik ini TF-IDF. Proses ini membantu mengidentifikasi kata kunci di dalam *dataset* yang relevan dengan sentimen yang dianalisis. Dengan menggunakan teknik proses pengolahan data TF-IDF, halaman ini dapat mengubah teks menjadi representasi numerik atau vector yang lebih mudah diproses oleh model analisis sentimen. Lalu dari halaman ini juga dilakukan *split data* atau pembagian data sentimen sebesar 80 persen data untuk dilakukan pelatihan *data training* dan 20 persen dilakukan untuk pengujian *data testing*. Hasil halaman *Feature Extraction* dapat dilihat pada Gambar 8.



Gambar 8 Halaman *Feature Extraction*

3.5 Halaman *Data Training*

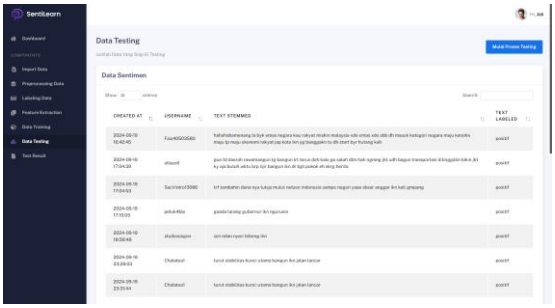
Halaman *Data Training* merupakan sebuah halaman yang diperuntukkan untuk mempersiapkan dan melaksanakan proses pelatihan model analisis sentimen. Pada halaman ini, *dataset* yang telah dilabeli dan diproses sebelumnya dibagi menjadi data pelatihan yang akan dipergunakan untuk melakukan pelatihan model dan pengujian data untuk evaluasi nantinya. Proses pelatihan model dilakukan dengan menggunakan algoritma klasifikasi seperti *Logistic Regression* atau metode lainnya. Hasil halaman *Data Training* dapat dilihat pada Gambar 9.



Gambar 9 Halaman *Data Training*

3.6 Halaman *Data Testing*

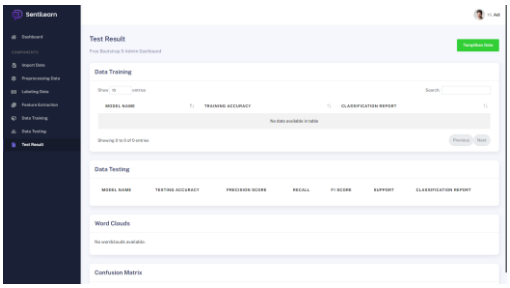
Halaman ini adalah halaman yang diperuntukkan menguji model analisis sentimen yang telah dilakukan uji latih menggunakan hasil *data training* sebelumnya. Pada halaman ini, beberapa data dilakukan proses uji untuk mengevaluasi seberapa baik model dalam mengklasifikasikan sentimen. Hasil evaluasi ini memberikan gambaran akurasi model dalam mengenali sentimen pada data yang belum pernah dilihat sebelumnya oleh model. Hasil halaman *Data Testing* dapat dilihat pada Gambar 10.



Gambar 10 Halaman *Data Testing*

3.7 Halaman *Test Result*

Halaman *Test Result* adalah halaman yang menampilkan hasil dari pengujian model analisis sentimen, termasuk *metric* evaluasi seperti nilai akurasi, hasil pada nilai presisi, lalu hasil pada nilai *recall*, dan hasil nilai *F1-score*. Halaman ini juga menampilkan *classification report*, *confusion matrix* untuk memberikan kesimpulan hasil dari model. Halaman ini memudahkan pengguna untuk mengevaluasi keakuratan model dan mengidentifikasi area perbaikan. Hasil halaman *Test Result* dapat dilihat pada hasil Gambar 11.



Gambar 11 Halaman *Test Result*

3.8 Hasil Klasifikasi Label

Tabel 1 menunjukkan hasil klasifikasi label sentimen positif pada *input* teks, di mana kolom *input text* berisi *tweet* asli yang dianalisis, sementara kolom *output text* menunjukkan hasil setelah *preprocessing data*. Misalnya, dalam contoh yang diberikan, teks asli yang berisi opini tentang Istana Negara dan Ibu Kota Nusantara (IKN) disederhanakan untuk menampilkan kata kunci yang relevan, tanpa mengubah makna kalimat, guna memudahkan analisis sentimen.

Tabel 1. Hasil Klasifikasi Label Positif

Input Text	***Output Text***
Semua Istana Negara di Indonesia punya nilai2 sejarah perjuangan Bangsa yg patut dibangakan Rakyat indonesia. Sedang istana IKN hanya nilai2 ambisi Dinasti Jokowi yg memaksakan hutang/mengeruk uang APBN rakyat disaat negara sdg skarat hutang a	istana negara indonesia nilai sejarah juang bangsa patut dibangakan rakyat indonesia istana ikn nilai ambisi dinasti jokowi paksa hutangmengeruk uang apbn rakyat saat negara skarat hutang a

Tabel 2 menunjukkan hasil klasifikasi label sentimen negatif pada *input text*, di mana kolom *input text* berisi *tweet* asli yang dianalisis, dan kolom *output text* menunjukkan hasil setelah *preprocessing data*. Contohnya, *tweet* yang mempertanyakan kualitas air PDAM di IKN disederhanakan untuk menampilkan kata-kata kunci seperti "pdam," "air," dan "ikn," yang relevan untuk analisis sentimen negatif.

Tabel 2. Hasil Klasifikasi Label Negatif

Input Text	***Output Text***
kalian sadar ga PDAM itu singkatan Perusahaan Daerah Air Minum. suka penasaran di mana daerah yg air PDAM-nya beneran bisa diminum. ternyata di IKN	sadar pdam singkat usaha daerah air minum suka penasaran daerah air pdamnya benar minum ikn

Tabel 3 menunjukkan hasil klasifikasi label sentimen netral pada *input text*, dengan *Input Text* berisi *tweet* asli yang tidak memihak atau mengungkapkan opini kuat, sementara *Output Text* adalah hasil pemrosesan teks setelah *preprocessing data*. Dalam contoh, *tweet* yang memberikan dukungan untuk seseorang yang pindah ke IKN disederhanakan dengan kata-kata yang lebih ringkas dan fokus pada elemen-elemen penting seperti "pindah," "keluarga," dan "IKN," yang mencerminkan sentimen netral.

Tabel 3. Hasil Klasifikasi Label Netral

Input Text	*Output Text*
Sok mas silahkan pindah kesana segera bawa keluarga ya mas semangat semoga betah ya jd penduduk IKN cayoo	sok mas sila pindah kesana bawa keluarga ya mas semangat moga betah ya duduk ikn cayoo

3.9 Hasil Pengujian Klasifikasi

Hasil performa dari pengujian model klasifikasi, *dataset* terbagi dengan skema 80% dari 1472 *dataset* digunakan untuk pelatihan *training data* dan nilai sebesar 20% sisa *dataset* digunakan untuk melakukan *testing data*. Pengevelauasian akan dilakukan melakukan perhitungan terhadap hasil klasifikasi dari *data training* berdasarkan tabel *classification report* yang digunakan dalam pemodelan untuk menilai kinerja model. Hasil yang diproses oleh sistem yaitu klasifikasi *data training* telah disajikan pada Tabel 4.

Tabel 4. *Classification Report Data Training*

Classification Report				
Index	***Precision (%)***	***Recall (%)***	***F1-Score (%)***	***Support***
Negative	94%	93%	94%	227.0
Neutral	95%	92%	93%	405.0
Positive	94%	96%	95%	545.0
Weighted Avg	94%	94%	94%	1177.0

Selanjutnya hasil performa pengujian pada model klasifikasi *data testing* menggunakan metode *Logistic Regression* telah disajikan pada Tabel 5.

Tabel 5. *Classification Report Data Testing*

Classification Report				
Index	***Precision (%)***	***Recall (%)***	***F1-Score (%)***	***Support***
Negative	41%	22%	29%	49.0
Neutral	60%	49%	54%	106.0
Positive	58%	75%	65%	140.0
Weighted Avg	57%	57%	55%	295.0

Dari hasil perhitungan evaluasi pada Tabel 4 dengan banyak data 1177 pada data training, menunjukkan hasil yang diperoleh *precision score* sebesar 94%, selanjutnya untuk hasil *recall* nilainya yang didapatkan sebesar 94%, dan yang terakhir hasil *f1-score* nilai yang didapatkan sebesar 94%. Selain itu data yang dihasilkan dari perhitungan *data testing* pada Tabel 5 yang menggunakan 295 data. Dalam pengujian menggunakan model *Logistic Regression* menunjukan bahwa hasil *precision* nilainya sebesar 56%, perhitungan *recall* nilainya sebesar 57%, dan *F1-Score* nilainya yang didapatkan sebesar 55%.
Dari pengujian juga menunjukkan hasil *confussion matrix data testing* telah disajikan pada Tabel 6.

Tabel 6. Confusion Matrix

Predicted Class	**True Class**			**Total**
	Negatif	**Netral**	**Positif**	
Negatif	11	10	28	49
Netral	6	52	48	106
Positif	10	25	105	140
Total	27	87	181	295

Setelah memperoleh hasilnya pada tabel *confussion matrix*, maka nilai tersebut akan dipergunakan untuk menghitung nilai *recall* menggunakan rumus persamaan (6), nilai *precision* melalui rumus persamaan (7), serta nilai *accuracy* dengan menggunakan

rumus persamaan (8). Hasil tersebut merupakan hasil yang dihitung melalui klasifikasi yang telah dimasukkan kedalam *confussion matrix*. Berikut adalah hasil evaluasi manual yang telah dihitung.

a. Recall

Recall digunakan untuk mengukur seberapa banyak kasus yang benar-benar ada dalam kelas tertentu dapat diprediksi dengan benar oleh model [25]. *Recall* untuk setiap kelas dihitung dengan rumus:

$$Recall_{(class)} = \frac{True\ Positive}{True\ Positive + False\ Neative} \quad (6)$$

Perhitungan *recall* untuk setiap kelas :

$$Recall_{(positif)} = \frac{TP}{TP+FN} = \frac{105}{10+25+105} = \frac{105}{140} = 0.750$$

$$Recall_{(negatif)} = \frac{TP}{TP+FN} = \frac{11}{11+10+28} = \frac{11}{49} = 0.224$$

$$Recall_{(netral)} = \frac{TP}{TP+FN} = \frac{52}{6+52+48} = \frac{52}{106} = 0.491$$

Perhitungan rata-rata dari *recall* untuk keseluruhan kelas adalah sebagai berikut:

$$Recal = \frac{(49\times0.224)+(106\times0.491)+(140\times0.750)}{49+106+140} = 0.569 \text{ x } 100\% = 56.9\%$$

b. Precision

Precision digunakan untuk mengukur seberapa banyak prediksi yang benar untuk setiap kelas dibandingkan dengan total prediksi untuk kelas tersebut [25]. *Precision* untuk setiap kelas dihitung dengan rumus:

$$Precision_{(class)} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (7)$$

Perhitungan Precision untuk setiap kelas :

$$Precision_{(positif)} = \frac{TP}{TP+FP} = \frac{105}{28+48+105} = \frac{105}{181} = 0.581$$

$$Precision_{(negatif)} = \frac{TP}{TP+FP} = \frac{11}{11+6+10} = \frac{11}{27} = 0.407$$

$$Precision_{(netral)} = \frac{TP}{TP+FP}x = \frac{52}{10+52+25} = \frac{52}{87} = 0.598$$

Perhitungan rata-rata dari *precision* untuk keseluruhan kelas adalah sebagai berikut:

$$Precision = \frac{(27\times0.407)+(87\times0.598)+(181\times0.581)}{27+87+181} = 0.557 \text{ x } 100\% = 55.7\%$$

c. Accuracy

Accuracy digunakan untuk mengukur seberapa banyak prediksi yang benar dibandingkan dengan total prediksi yang dilakukan oleh model [25].

$$Accuracy = \frac{True\ Positif + True\ Negative}{Total\ Data}\ x\ 100\% = \frac{168}{295}\ x\ 100\% = 56.9\% \quad (8)$$

Berdasarkan informasi yang ditunjukkan *confusion matrix*, terdapat 295 kalimat yang diprediksi sebagai positif, negatif, dan netral. Hasil prediksi menunjukkan bahwa terdapat 105 kalimat yang diprediksi sebagai kalimat positif. Lalu terdapat 11 kalimat yang diprediksi sebagai negatif, terdapat 10 kalimat diprediksi negatif tetapi faktanya merupakan kalimat netral. Selain itu juga terdapat 48 kalimat yang hasilnya diprediksi sebagai netral tetapi faktanya merupakan positif. Selain itu terdapat pula 56 kalimat yang diprediksi sebagai kalimat netral.

Untuk memberikan gambaran yang lebih jelas tentang kinerja metode klasifikasi teks lainnya, berikut disajikan tabel 7 perbandingan performa beberapa algoritma yang umum digunakan berdasarkan literatur terkait.

Tabel 7. Perbandingan Metode Dengan Berbagai Literatur

Penelitian	Metode Digunakan	Akurasi (%)	Keterangan
Studi Komparasi *Naïve Bayes*, SVM, dan *Logistic Regression* [26]	*Naïve Bayes*	76%	Mengasumsikan fitur independen, lebih baik dalam kategori tertentu dengan data biner.
	Support Vector Machine (SVM)	92%	Tahan terhadap dataset yang tidak seimbang, menghasilkan performa stabil.
	Logistic Regression	92%	Stabil pada data biner namun cenderung lemah pada dataset multi-kelas.
Analisis Sentimen Ulasan Aplikasi Google Meet [27]	*Support Vector Machine* (SVM)	87%	Kernel linear menghasilkan akurasi tertinggi dibanding kernel lainnya (RBF, sigmoid, dll).
	Logistic Regression	85%	Performa sedikit lebih rendah dibanding SVM linear kernel.
Sentimen Metaverse [28]	*Naïve Bayes*	90%	Mengalami peningkatan performa setelah optimasi SMOTE.
	Logistic Regression	95%	Setelah optimasi SMOTE, performa meningkat signifikan pada kategori minoritas.

Tabel di atas menggambarkan kinerja beberapa metode klasifikasi teks berdasarkan berbagai literatur. *Naïve Bayes* menunjukkan performa yang baik pada data dengan distribusi kategori tertentu, seperti pada literatur pertama dan ketiga, terutama setelah dilakukan optimasi SMOTE. *Logistic Regression*, meskipun sering digunakan, memiliki kekurangan pada *dataset* multi-kelas, namun tetap kompetitif setelah optimasi seperti pada literatur ketiga. Sementara itu, SVM, khususnya dengan kernel linear, cenderung memberikan akurasi tinggi dan konsistensi dalam *dataset* biner, seperti yang terlihat pada literatur ketiga. Hasil ini menunjukkan bahwa pemilihan metode dan teknik optimasi sangat bergantung pada karakteristik *dataset* serta tujuan klasifikasi yang ingin dicapai.

Penyebab rendahnya performa pada *data testing* dapat dijelaskan oleh beberapa faktor, salah satunya adalah ketidakseimbangan *dataset* yang terdiri dari tiga kategori yaitu positif, negatif, dan netral. Ketidakseimbangan ini mengakibatkan model kesulitan dalam memprediksi kategori yang kurang dominan, yang tercermin pada hasil yang kurang optimal pada *data testing*. Selain itu, metode *Logistic Regression* yang digunakan dalam penelitian ini lebih cocok untuk klasifikasi biner, meskipun telah disesuaikan dengan parameter seperti *solver liblinear* dan *class weight balanced* untuk menangani ketidakseimbangan kelas. Namun, metode ini masih memiliki keterbatasan dalam menangani masalah klasifikasi *multi-class* seperti yang ada pada *dataset* ini, yang turut berkontribusi pada rendahnya performa model pada *data testing*.

4. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan sebuah *website* untuk analisis sentimen dan menguji kinerja sistem menggunakan metode *Logistic Regression* pada studi kasus analisis *tweet* terkait pembangunan Ibu Kota Nusantara (IKN). Hasil penelitian menunjukkan bahwa *tweet* dengan sentimen positif mendominasi, diikuti oleh sentimen netral, dan terakhir sentimen negatif. Sistem ini berfungsi sesuai harapan dan dapat memberikan wawasan yang mendalam tentang persepsi publik terhadap pembangunan IKN. Dampak praktisnya adalah kemampuannya mendukung pembuat kebijakan dalam memahami opini publik secara *real-time*, sehingga dapat memperkuat perencanaan dan komunikasi kebijakan yang lebih efektif.

Namun, performa model pada *data testing* yang kurang optimal disebabkan oleh ketidakseimbangan *dataset* dengan kategori positif, negatif, dan netral. Ketidakseimbangan ini menyulitkan model memprediksi kategori yang kurang dominan. Selain itu, metode *Logistic Regression* yang lebih cocok untuk klasifikasi biner memiliki keterbatasan dalam menangani klasifikasi *multi-class* meskipun telah disesuaikan dengan parameter seperti *solver liblinear* dan *class weight balanced*. Untuk penelitian selanjutnya, disarankan mengeksplorasi metode lain seperti *Support Vector Machine* (SVM), *Random Forest*, atau pendekatan berbasis *Deep Learning* (misalnya *Recurrent Neural Network* atau *Transformer-based models*). Selain itu, peningkatan *preprocessing data* melalui *balancing dataset* dengan teknik *oversampling* atau *undersampling*, serta data *augmentation*, dapat membantu meningkatkan akurasi model dan mengoptimalkan sistem dalam mendukung analisis kebijakan publik.

DAFTAR PUSTAKA

[1] G. Aji, Z. Arfani, A. M. Sari, R. Seprtiani, and U. K. H. Abdurrahman Wahid, "Dampak Pemindahan Ibukota Negara Baru terhadap Ekonomi dan Sosial di Provinsi Kalimantan Timur," *J. Ilmu Huk.*, vol. 1, no. 5, pp. 2985–5624, 2023, [Online]. Available: http://jurnal.kolibi.org/index.php/kultura

[2] I. A. Ricky, I. F. Hanif, F. N. Hasan, E. S. Sinduningrum, Z. Halim, and N. Nunik, "Analisis Sentimen Opini Masyarakat Terkait Penyelenggaraan Sistem Elektronik Menggunakan Metode Logistic Regression," *J. Linguist. Komputasional*, vol. 5, no. 2, p. 77, 2022, [Online]. Available: https://t.co/23c4krbjp

[3] I. R. Ainunnisa and S. Sulastri, "Analisis Sentimen Aplikasi Tiktok dengan Metode Support Vector Machine (SVM), Logistic Regression dan NaÃ¯ve Bayes," *J. Teknol. Sist. Inf. dan Apl.*, vol. 6, no. 3, pp. 423–430, 2023, doi: 10.32493/jtsi.v6i3.31076.

[4] I. Rahmawati, T. Rika Fitriani, A. No'eman, and A. Y. P. Yusuf, "Analisis Sentimen Menggunakan Algoritma Logistic Regression Pada Penerbangan Lion Air berdasarkan Ulasan Platform Online," *J. Ris. Inform. dan Teknol. Inf.*, vol. 1, no. 1, pp. 11–16, 2023, doi: 10.58776/jriti.v1i1.60.

[5] geeksforgeeks.org, "Logistic Regression in Machine Learning," geeksforgeeks.org. [Online]. Available: https://www.geeksforgeeks.org/understanding-logistic-regression/

[6] Ash Shiddicky and Surya Agustian, "Analisis Sentimen Masyarakat Terhadap Kebijakan Vaksinasi Covid-19 pada Media Sosial Twitter menggunakan Metode Logistic Regression," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 2, pp. 99–106, 2022, doi: 10.37859/coscitech.v3i2.3836.

[7] T. Ridwansyah, "Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 2, no. 5, pp. 178–185, 2022, doi: 10.30865/klik.v2i5.362.

[8] N. Hendrastuty, A. Rahman Isnain, and A. Yanti Rahmadhani, "Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine," *J. Inform. J. Pengemb. IT*, vol. 6, no. 3, pp. 150–155, 2021.

[9] I. Verawati and B. S. Audit, "Algoritma Naïve Bayes Classifier Untuk Analisis Sentiment Pengguna Twitter Terhadap Provider By.u," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1411, 2022, doi: 10.30865/mib.v6i3.4132.

[10] I. Budianto, S. N. Anwar, J. T. Lomba, J. Nomor, and K. Semarang, "Analisis Sentiment Pengguna Twitter Mengenai Program Vaksinasi Covid-19 Menggunakan Algoritma Naïve Bayes," *J. Teknol. Inf.*, vol. 6, no. 1, pp. 37–43, 2022.

[11] Permana A, Taufiq R, and Wijaya M, "Implementasi Algoritma Naïve Bayes Terhadap Review Aplikasi KFCKU," *J. Tek.*, vol. Vol. 12 N, no. 02, pp. 128–137, 2023.

[12] A. D. Adhi Putra, "Sentiment Analysis on User Reviews of the Bibit and Bareksa Application with the KNN Algorithm," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 2, pp. 636–646, 2021.

[13] T. M. Permata Aulia, N. Arifin, and R. Mayasari, "Perbandingan Kernel Support Vector Machine (Svm) Dalam Penerapan Analisis Sentimen Vaksinisasi Covid-19," *SINTECH (Science Inf. Technol. J.*, vol. 4, no. 2, pp. 139–145, 2021, doi: 10.31598/sintechjournal.v4i2.762.

[14] J. Supriyanto, D. Alita, and A. R. Isnain, "Penerapan Algoritma K-Nearest Neighbor (K-NN) Untuk Analisis Sentimen Publik Terhadap Pembelajaran Daring," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 4, no. 1, pp. 74–80, 2023, doi: 10.33365/jatika.v4i1.2468.

[15] H. Tuhuteru, "Analisis Sentimen Masyarakat Terhadap Pembatasan Sosial Berksala Besar Menggunakan Algoritma Support Vector Machine," *Inf. Syst. Dev.*, vol. 5, no. 2, pp. 7–13, 2020.

[16] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 9, pp. 4305–4313, 2022.

[17] D. Alita and A. R. Isnain, "Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier," *J. Komputasi*, vol. 8, no. 2, pp. 50–58, 2020, doi: 10.23960/komputasi.v8i2.2615.

[18] E. H. Muktafin, K. Kusrini, and E. T. Luthfi, "Analisis Sentimen pada Ulasan Pembelian Produk di Marketplace Shopee Menggunakan Pendekatan Natural Language Processing," *J. Eksplora Inform.*, vol. 10, no. 1, pp. 32–42, 2020, doi: 10.30864/eksplora.v10i1.390.

[19] Scikit-learn.org, "No Title," scikit-learn.org. [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html

[20] Www.geeksforgeeks.org, "No Title," www.geeksforgeeks.org. [Online]. Available: https://www.geeksforgeeks.org/how-does-the-classweight-parameter-in-scikit-learn-work/

[21] geeksforgeeks.org, "How To Do Train Test Split Using Sklearn In Python," geeksforgeeks.org.
[22] K. Kelvin, J. Banjarnahor, E. I. -, and M. NK Nababan, "Analisis perbandingan sentimen Corona Virus Disease-2019 (Covid19) pada Twitter Menggunakan Metode Logistic Regression Dan Support Vector Machine (SVM)," *J. Sist. Inf. dan Ilmu Komput. Prima(JUSIKOM PRIMA)*, vol. 5, no. 2, pp. 47–52, 2022, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v5i2.2365.
[23] R. Deswandi Yahya, S. Adi Wibowo, and N. Vendyansyah, "Analisis Sentimen Untuk Deteksi Ujaran Kebencian Pada Media Sosial Terkait Pemilu 2024 Menggunakan Metode Support Vector Machine," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 2, pp. 1182–1189, 2024, doi: 10.36040/jati.v8i2.9076.
[24] S. Andayani and A. Ryansyah, "Implementasi Algoritma TF-IDF Pada Pengukuran Kesamaan Dokumen," *JuSiTik J. Sist. dan Teknol. Inf. Komun.*, vol. 1, no. 1, p. 53, 2017, doi: 10.32524/jusitik.v1i1.218.
[25] 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.
[26] M. Z. Anbari and B. Sugiantoro, "Studi Komparasi Metode Analisis Sentimen Naïve Bayes, SVM, dan Logistic Regression Pada Piala Dunia 2022," *J. Media Inform. Budidarma*, vol. 7, no. 2, p. 688, 2023, doi: 10.30865/mib.v7i2.5383.
[27] A. Novantika, "Analisis sentimen ulasan pengguna aplikasi video conference google meet menggunakan metode svm dan logistic regression," *Prism. Pros. Semin. Nas. Mat.*, vol. 5, pp. 808–813, 2022, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/prisma/
[28] B. Ramadhani and R. R. Suryono, "Komparasi Algoritma Naïve Bayes dan Logistic Regression Untuk Analisis Sentimen Metaverse," *J. Media Inform. Budidarma*, vol. 8, no. 2, p. 714, 2024, doi: 10.30865/mib.v8i2.7458.