SKANIKA: Sistem Komputer dan Teknik Informatika
Volume 8, Nomor 1, Januari 2025, Halaman 1-12 E-ISSN: 2721-4788

Sistem *Monitoring* Keadaan Darurat Berdasarkan *Caption Instagram* Menggunakan *Naïve Bayes* Dengan Jarak *Levenshtein*

Muhammad Fahmi Zulfikar[1], Utomo Budiyanto[2*]
[1,2]Fakultas Teknologi Informasi, Teknik Informatika, Universitas Budi Luhur, Jakarta, Indonesia
Email: [1]2011502065@student.budiluhur.ac.id, [2*]utomo.budiyanto@budiluhur.ac.id
(* : corresponding author)

Abstrak

Di era digital ini, Instagram tidak hanya menjadi platform berbagi foto dan cerita terpopuler, tetapi juga berperan sebagai sumber informasi penting dan kritis seperti informasi mengenai keadaan darurat. Penelitian ini bertujuan untuk memanfaatkan data *caption* dari unggahan media sosial Instagram untuk dihasilkan menjadi peta yang menunjukkan keadaan darurat di wilayah Jakarta Selatan. Dari 1088 unggahan teks yang akan dianalisis, 204 diantaranya berhasil diidentifikasi sebagai teks yang informatif dan relevan untuk penelitian ini. Data informatif tersebut kemudian diproses dan divisualisasikan sebagai sistem informasi geografis menggunakan peta *OpenStreetMap* serta API *Overpass Turbo*, sehingga informasi yang dihasilkan dapat lebih mudah dipahami serta dapat dijadikan bahan analisis lebih lanjut oleh masyarakat atau pihak berwenang. Pendekatan penelitian ini menggabungkan metode TF-IDF dan Multinomial *Naïve Bayes* untuk klasifikasi terhadap teks *caption* pada unggahan instagram yang informatif atau tidak informatif, serta Algoritma Jarak Levenshtein untuk pembersihan dataset dengan cara memperbaiki tulisan salah ketik sehingga dapat mengurangi dimensi dan meningkatkan kualitas data yang akan dianalisis. Tantangan dalam mengatasi ketidakseimbangan data diatasi melalui penerapan tiga teknik augmentasi teks (*text augmentation*), yaitu penggantian sinonim (*synonym replacement*), pertukaran acak (*random swap*), dan penghapusan acak (*random deletion*). Dari proses pelatihan model, diperoleh tingkat akurasi sebesar 92,6%. Hasil ini menunjukkan bahwa metode yang diusulkan tidak hanya efektif dalam meningkatkan akurasi klasifikasi, tetapi juga berhasil memberikan visualisasi yang informatif terkait lokasi dan frekuensi keadaan darurat di wilayah Jakarta Selatan.

Kata Kunci: Klasifikasi, *Instagram*, Keadaan darurat, Sistem Informasi Geografis, Multinomial *Naive Bayes*, Algoritma Jarak Levenshtein

Abstract

In this digital era, Instagram is not only the most popular photo and story sharing platform, but also acts as a source of important and critical information such as information about emergency situation. This research aims to utilize caption data from Instagram posts to produce a map showing the location of emergency situations in South Jakarta. From 1088 captions, 204 of them were successfully identified as informative and relevant texts for this study. The informative data then processed and visualized as a geographic information system using the OpenStreetMap map and the Overpass Turbo API, so that the resulting information can be more easily understood and can be used as material for further analysis by the public or authorities. This research combines the TF-IDF and Multinomial Naïve Bayes methods for classifying informative or uninformative captions on Instagram posts, as well as the Levenshtein Distance Algorithm for cleaning the dataset by correcting typos to reduce dimensions and improve the quality of the data that are going to be analyzed. The challenge in overcoming data imbalance is by implementing three text augmentation techniques, namely synonym replacement, random swap, and random deletion. From the model training process, an accuracy rate of 92,6% was obtained. This result shows that the proposed method is not only effective in improving classification accuracy, but also successfully provides informative visualizations related to the location and frequency of emergencies in South Jakarta.

Keywords*: Classification, Instagram, Emergency, Geographic Information System, Multinomial Naive Bayes, Levenshtein Distance Algorithm*

1. PENDAHULUAN

Dalam era digital yang semakin berkembang, media sosial telah menjadi sumber informasi yang sangat berharga, Media sosial telah menjadi bagian penting dari hidup dan berhasil menjangkau berbagai pengguna mulai dari konsumen produk, pebisnis, pemerintah, organisasi dan komunitas. Media sosial sendiri memiliki makna yang mengacu pada pembuatan,

penyebaran, dan komunikasi suatu konten yang berupa percakapan, penyebaran informasi, atau komunikasi di antara komunitas [1]. Pemanfaatan media sosial saat ini sudah lebih dari untuk berdiskusi, berbagi momen, dan bercerita [2]. Media sosial saat ini telah menjadi wadah untuk menampung dan mencari sumber informasi bagi masyarakat seperti mencari informasi krisis, terutama saat terjadi bencana alam atau bencana karena ulah manusia [3]. Salah satu media sosial yang populer di Indonesia adalah Instagram.

Menurut laporan "Digital 2024" yang dirilis oleh *We Are Social*, Instagram menduduki peringkat pertama sebagai aplikasi media sosial favorit secara global dan peringkat kedua di Indonesia. Pada kuartal pertama tahun 2024, *Instagram* memiliki 2 miliar pengguna aktif di seluruh dunia, dengan 85,3% dari 139 juta pengguna media sosial di Indonesia adalah pengguna Instagram [4]. Instagram juga telah dimanfaatkan sebagai media penyebaran informasi situasi darurat. Di Indonesia, situasi darurat sering terjadi di seluruh wilayah seperti kebakaran, banjir, angin puting beliung, gempa bumi, dan lain-lain [3]. Menurut data BPBD sepanjang tahun 2023 di daerah Jakarta telah terjadi 65 kasus banjir, 864 kasus kebakaran, 234 kasus pohon tumbang, dan 4 kasus angin kencang [5].

Penelitian ini mengusulkan pemanfaatan Instagram sebagai sumber data untuk klasifikasi teks keadaan darurat di Jakarta Selatan. Pembuatan model prediktif dilakukan dengan melakukan klasifikasi teks menggunakan metode naïve bayes dengan algoritma jarak *Levenshtein* dilanjut visualisasi hasil prediksi melalui *Geographic Information System* (GIS) dengan peta *OpenStreetMap*. Visualisasi akan membantu masyarakat dan pihak berwajib dalam memahami pola keadaan darurat dengan memperkirakan wilayah mana saja yang rentan mengalami keadaan darurat untuk pengambilan keputusan yang lebih tepat. Algoritma jarak *levenshtein* digunakan untuk memperbaiki tulisan salah ketik yang mana membantu model dalam menggeneralisir data.

Pengolahan data media sosial dan penggunaan GIS pernah dilakukan di beberapa penelitian sebelumnya. Penelitian tentang klasifikasi *tweet* media sosial *Twitter* terkait bencana di Indonesia menggunakan *Support Vector Machine* dan *Naïve Bayes* dengan akurasi masing-masing 81,03% dan 80,03% [3]. Penelitian ini terbatas pada *Twitter* saja dan tidak menggunakan *platform* media sosial lain yang juga berpotensi memberikan informasi darurat seperti Instagram. Krishnan dkk., (2023) membahas klasifikasi menggunakan model bahasa pra-latih BERT dan CNN untuk mendeteksi titik bencana dari teks dan gambar di *Twitter* yang masing-masing mendapatkan 92% dan 74% akurasi serta menggunakan GIS untuk visualisasi lokasi bencana dari hasil klasifikasi [6]. Penelitian ini juga menggunakan *Twitter* dan belum mengeksplorasi penggunaan metode lain yang lebih sederhana. Sistem *monitoring* bencana menggunakan GIS dengan mengklasifikasi teks dari *Twitter* menggunakan metode K-*Nearest Neighbor* lalu membandingkan berbagai metrik jarak dan menghasilkan nilai *confusion matrix* terbaik sebesar 86% [7]. Penelitian ini juga menggunakan Twitter dan kombinasi metrik jarak tetapi menggunakan metode yang berbeda.

Pemanfaatan Instagram sebagai sumber data untuk prediksi dan integrasi visualisasi GIS untuk pemetaan keadaan darurat di Jakarta Selatan diharapkan dapat meningkatkan efisiensi masyarakat dan pihak berwajib dalam mengidentifikasi dan merespons kejadian di sekitar secara lebih cepat, sehingga dapat memfasilitasi manajemen keadaan darurat yang lebih efektif.

2. METODE PENELITIAN

2.1 Penambangan Teks (*Text Mining*)

Penambangan teks mengacu pada ekstraksi informasi berharga dan pola pada teks yang sebelumnya tidak diketahui. Proses ini dapat dilakukan dengan cara otomatis (*unsupervised*) atau semi-otomatis (*supervised*) dari data tekstual yang sangat besar dan tidak terstruktur, seperti teks bahasa alami [8]. Penambangan teks saat ini penting khususnya di era media sosial.

Sumber data tersebut biasanya diperoleh dari berbagai jenis dokumen teks. Tujuannya untuk menemukan kata-kata yang dapat mewakili isi dari dokumen tersebut sehingga dapat dilakukan analisis [9]. Analisis ini dapat mencakup klasifikasi teks, *clustering*, ekstraksi entitas, atau pencarian hubungan antar kata dalam teks.

Secara khusus, penambangan teks adalah proses ekstraksi informasi dan pengetahuan yang berguna dari kumpulan data besar tidak terstruktur berupa teks yang berasal dari berbagai sumber. Proses ini dapat dilakukan secara *supervised* atau *unsupervised*. Metode ini memudahkan peneliti dan praktisi dalam mendapatkan atau membuat informasi lebih detail dari data tekstual dengan jumlah yang banyak.

2.2 Pengumpulan Data Teks

Pengumpulan data menggunakan *scraper Instaloader*. Instaloader adalah alat yang digunakan untuk mengunduh konten dan informasi dari Instagram. Contohnya seperti foto, *video*, *instastory*, profil pengguna, *tag*, serta lokasi. Beberapa jenis informasi hanya dapat diakses secara terbatas dengan informasi *login* Instagram seperti lokasi dan *instastory*.

Pada penelitian ini hanya akan mengunduh *caption* dari unggahan. Rentang waktu diambil dari bulan Maret tahun 2021 sampai bulan April tahun 2024. Sementara sumber informasi berasal dari akun @jakarta.terkini.

Dengan demikian, *caption* yang dikumpulkan akan dianalisis lebih lanjut untuk dipahami pola teks terkait keadaan darurat di wilayah Jakarta Selatan. Rentang waktu yang panjang memungkinkan analisis yang lebih mendalam terhadap perubahan atau pergeseran dalam pola teks. Penggunaan akun yang tepat juga berpengaruh terhadap jumlah teks yang berkaitan dengan keadaan darurat.

2.3 Pra-pemrosesan Data Teks (*Text Preprocessing)*

Pra-pemrosesan teks adalah tahap untuk mendapatkan fitur utama dari dokumen dan meningkatkan relevansi antara kata dengan dokumen serta relevansi antara kata dengan kelas atau label [10]. Dalam penelitian ini, pra-pemrosesan yang dilakukan meliputi beberapa langkah. Pra-pemrosesan yang digunakan adalah *casefolding*, penghapusan tautan dan karakter non alfabet, pengubahan kata singkat seperti kata “gk” atau “nggak” menjadi kata asli yaitu “tidak”.

Setelah kata menjadi satu ragam, maka dilakukan tokenisasi, penghapusan *stopword*, *stemming*, normalisasi kata yang mengubah kata modifikasi seperti “iyaaa” menjadi kata asli yaitu “iya”, dan perbaikan kata salah ketik menggunakan algoritma jarak Levenshtein.

Algoritma Jarak Levenshtein bekerja dengan melakukan komparasi dan menghitung jumlah pengubahan antar kata tidak dikenal dengan kata asli yang ada dalam kamus. Dalam penelitian ini digunakan Kamus Besar Bahasa Indonesia. Kata dalam kamus dengan jumlah pengubahan paling sedikit akan dimasukkan ke dalam daftar kandidat kata perbaikan yang benar. Selanjutnya pemilihan kata asli dilakukan menggunakan n-gram agar sesuai dengan konteks.

2.4 Pelabelan Data Teks

Karena metode yang digunakan adalah *supervised*, dibutuhkan label untuk membedakan teks yang cocok dan tidak cocok untuk visualisasi. Oleh sebab itu, diberikan dua label, yaitu label “*informative*” dan “*not informative*”. Pelabelan data dilakukan dengan dua cara. Pertama menggunakan *regular expression* untuk mengidentifikasi pola tertentu dalam teks.

Kedua secara manual menggunakan bantuan pakar kebencanaan. Konsultasi dilakukan untuk memberikan penilaian berdasarkan konteks dan relevansi informasi dalam teks. Dengan kombinasi kedua metode ini pelabelan dapat menghasilkan data yang akurat dan sesuai dengan tujuan penelitian.

Dari pelabelan didapat bahwa hanya 204 teks berlabel “*informative*” dan menyisakan 884 teks berlabel “*not informative*”. Dari total 1088 data terdapat ketidakseimbangan data terhadap label “*not informative*”. Oleh sebab itu dilakukan *oversampling* dengan teknik *text augmentation* dengan tiga proses, yaitu *Synonym Replacement* (SR), *Random Swap* (RS), dan *Random Deletion* (RD) dijalankan secara berurutan. Penggunaan proses ini berdasarkan penelitian yang telah dilakukan sebelumnya oleh Azizah dkk., [11] tentang *Easy Data Augmentation* yang menghasilkan bahwa *random insertion* kurang berpengaruh terhadap model *Naive Bayes*.

2.5 Klasifikasi Data Teks

Klasifikasi teks pada penelitian ini menggunakan TF-IDF dan Multinomial *Naive Bayes*. TF-IDF adalah metode untuk mengetahui nilai kata dalam suatu dokumen sehingga dapat menyimpulkan kata tersebut dianggap penting atau tidak [7]. Hasil TF-IDF berupa kumpulan nilai kepentingan tiap kata dalam bentuk vektor.

Term Frequency (TF) adalah bobot sebuah kata dalam sebuah data yang dihitung berdasarkan frekuensi kemunculannya dalam kumpulan data tersebut. Semakin tinggi nilai TF akan semakin tinggi bobot kata dalam dokumen tersebut. TF akan mengidentifikasi seberapa sering suatu kata muncul dalam dokumen tertentu.

Inverse Document Frequency (IDF) adalah faktor yang memperkirakan kelangkaan suatu *term* dalam seluruh dokumen. IDF akan rendah jika sebuah kata muncul di banyak dokumen dan akan tinggi jika hanya muncul di beberapa dokumen. Perhitungan TF dan IDF dapat dikalkulasikan dengan rumus 1 dan 2. Nilai kepentingan kata dirumuskan dengan mengalikan TF dan IDF pada rumus 3.

$$\mathrm{TF} = \frac{F}{N} \tag{1}$$

Dimana:
TF = *Term Frequency*
F = Frekuensi atau banyak kemunculan kata dalam kalimat
N = Panjang kalimat

$$\mathrm{IDF} = ln\left(\frac{N}{DF}\right) \tag{2}$$

Dimana:
IDF = *Inverse Document Frequency*
N = Jumlah total seluruh dokumen
DF = Frekuensi kemunculan kata dalam semua dokumen

$$\mathrm{TFIDF} = \mathrm{TF} \times \mathrm{IDF} \tag{3}$$

Dimana:
TF = *Term Frequency*
IDF = *Inverse Document Frequency*

Multinomial *Naïve Bayes* merupakan metode pembelajaran probabilistik *supervised*, sehingga setiap data perlu diberikan label sebelum dilakukan *training* [12]. Pada metode ini, kelas atau label dokumen ditentukan oleh kata-kata yang muncul dengan jumlah kemunculannya [13]. Metode ini akan mencari nilai probabilitas posterior, yaitu probabilitas suatu kejadian yang diperbarui setelah mempertimbangkan data probabilitas prior dan data kemungkinan (*likelihood*).

Dalam metode ini rentan terjadi *underflow* atau hasil perkalian sangat kecil yang dapat mengakibatkan suatu kata menjadi tidak dapat direpresentasikan dengan akurat. Oleh karena itu dilakukan penambahan logaritma terhadap rumus. Penambahan langkah ini akan memberikan hasil yang lebih stabil [14].

Probabilitas *prior* disebut sebagai tingkat kepercayaan awal terhadap hipotesis atau kelas. Dengan kata lain, prior mengukur kemungkinan keyakinan awal dari suatu kelas [15]. Probabilitas *prior* dikalkulasikan dengan rumus 4.

$$\log P(L_x) = \ln(\frac{N_x}{N}) \tag{4}$$

Dimana:
P = Probabilitas
L_x = Label atau kelas x
N_x = Jumlah dokumen dalam label atau kelas x
N = Jumlah total seluruh dokumen

$P(L_x)$ = *Prior Probability*, probabilitas awal label x berdasarkan distribusi kelas dalam dataset

Dalam klasifikasi teks, *likelihood* digunakan untuk mencari seberapa besar kemungkinan suatu teks termasuk dalam suatu kelas berdasarkan kemunculan setiap kata dalam teks. *Likelihood* dikalkulasikan dengan rumus 5.

$$\log P(T_i|L_x) = \ln(\frac{N_{ix} + \alpha}{N_x + \alpha V}) \quad (5)$$

Dimana:

T_i = Teks atau dokumen i
N_{ix} = Jumlah kemunculan kata i dalam dokumen kelas x
α = Parameter smoothing
V = Jumlah total kata unik di semua dokumen
$P(T_i|L_x)$ = *Likelihood*, probabilitas bahwa fitur-fitur pada teks i termasuk dalam kelas atau label x

Kedua probabilitas tersebut akan digunakan dalam mencari probabilitas posterior yang dikalkulasikan dengan rumus 6.

$$P(L_x|d) \propto P(L_x) + \sum_{i=1}^{n} ln\ P(T_i|L_x) \quad (6)$$

Dimana:

$P(L_x|d)$ = *Posterior probability*, probabilitas bahwa teks i termasuk dalam kelas atau label x diberikan dokumen d.
d = Dokumen yang akan diprediksi kelasnya.

2.6 Pengujian Model

Pengujian menggunakan dua metode evaluasi, yaitu *Confusion Matrix* dan *Evaluation Metrics*. *Confusion Matrix* adalah sebuah tabel yang menggambarkan jumlah data uji yang diklasifikasikan dengan benar dan jumlah data uji yang salah klasifikasi [16]. Evaluasi ini melakukan perhitungan terhadap jumlah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *dan False Negative* (FN). Tabel 1 menjabarkan *Confusion Matrix* untuk klasifikasi biner.

Tabel 1. *Confusion Matrix* untuk klasifikasi biner

Kelas Aktual	Kelas prediksi	
	1	0
1	TP	FN
0	FP	TN

Dimana:

TP (*True Positive*) = Jumlah kelas 1 yang diklasifikasikan benar sebagai kelas 1
TN (*True Negative*) = Jumlah kelas 0 yang diklasifikasikan benar sebagai kelas 0
FP (*False Positive*) = Jumlah kelas 0 yang diklasifikasikan salah sebagai kelas 1
FN (*False Negative*) = Jumlah kelas 1 yang diklasifikasikan salah sebagai kelas 0

Evaluation Metrics adalah evaluasi kuantitatif yang digunakan untuk pelatihan dan evaluasi. Metrik ini memberikan informasi dari performa serta akurasi model dalam mengklasifikasikan data. Metrik yang digunakan adalah *accuracy, recall, precision,* dan *f1-score* [17]. Metrik tersebut masing-masing dikalkulasikan pada formula 7, 8, 9, dan 10.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$precision = \frac{TP}{TP + FP} \quad (8)$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

$$f1 - score = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{10}$$

Dengan kedua metode evaluasi ini, dapat dilakukan penilaian menyeluruh terhadap efektivitas model dalam klasifikasi teks. *Confusion Matrix* memberikan detail spesifik mengenai kesalahan klasifikasi, sedangkan *Evaluation Metrics* memberikan persentase dari performa model. Kombinasi keduanya akan memberikan evaluasi model berdasarkan analisis kesalahan dan performa keseluruhan model.

2.7 Visualisasi

Sistem *monitoring* akan menggunakan visualisasi peta. Visualisasi dilakukan setelah lokasi berhasil diekstraksi dari teks. Ekstraksi ini penting untuk pencarian koordinat dan peletakan marka.

Ekstraksi lokasi dan jenis keadaan darurat dilakukan dengan *regular expression*. *Regular Expression* akan mencari kata yang sama dengan nama lokasi yang terdapat dalam *database OpenStreetMaps* untuk dicari koordinatnya menggunakan *Overpass Turbo*. Ekstraksi jenis dilakukan untuk memberikan informasi mengenai keadaan darurat apa yang terjadi pada suatu lokasi.

Visualisasi untuk sistem monitoring berupa peta wilayah Jakarta Selatan dan marka untuk menunjukkan lokasi keadaan darurat. *OpenStreetMap* akan menampilkan peta dan marka lokasi sementara koordinat untuk letak marka didapat menggunakan API *Overpass Turbo*. Visualisasi ini dibuat dengan sederhana agar sistem *monitoring* ini dapat dimanfaatkan dengan mudah oleh pengguna.

3. HASIL DAN PEMBAHASAN

Dari hasil pengumpulan data didapatkan 1088 data lalu dilakukan pelabelan yang menghasilkan 204 teks informatif dan 884 teks tidak informatif untuk divisualisasikan. Data kemudian dibagi menjadi 2 dengan perbandingan 30% data uji dan 70% data latih.

3.1 Hasil Pra-Pemrosesan Data

Data yang masih berantakan dibersihkan sesuai dengan langkah pra-pemrosesan pada sub-bab 2.3. Dari langkah tersebut didapat hasil daftar kata dasar yang berasal dari teks asli seperti pada tabel 2. Dalam daftar kata terdapat beberapa perbaikan kata seperti “lalulibtas” menjadi “lalu lintas” dan “terpatau” menjadi “pantau”. Karena ada perbaikan yang memiliki 2 kata, maka dilakukan tokenisasi lagi untuk memisah kedua kata tersebut.

Tabel 2. Tabel Sampel Teks Asli dengan Hasil Pra-pemrosesan

Teks Asli	Teks Hasil
Truk mengalami patah as roda di Ruas Jalan Gatot Subroto dari arah Kuningan menuju Pancoran siang ini, Sabtu, 22/7/2023. Sementara lalulintas terpantau macet. . Untuk share info di @jakarta.terkini kirim melalui DM / mention story /atau WA cek di Bio . Jakarta terkini, dekat dengan Jakarta! Credit by: @tmcpoldametro	['truk', 'alami', 'patah', 'as', 'roda', 'ruas', 'jalan', 'gatot', 'subroto', 'arah', 'kuningan', 'pancoran', 'siang', 'sabtu', 'lalu', 'lintas', 'pantau', 'macet']
16:10 Wib Pantauan lalulibtas kawasan Mampang prapatan sore ini terpatau macet, Kamis 14/4/22 #infolalin #mampang Kiriman dari:@wayan_eko.putra #jakartaterkini dekat dengan #jakarta	['wib', 'pantau', 'lalu', 'lintas', 'kawasan', 'mampang', 'prapatan', 'sore', 'pantau', 'macet', 'kamis', 'kirim']

3.2 Penerapan TFIDF dan Multinomial Naive Bayes

Hasil perhitungan TFIDF berupa vektor numerik yang merepresentasikan probabilitas kemunculan kata-kata pada dokumen. Sampel data pada tabel 3 dikalkulasikan menggunakan rumus TFIDF. Perhitungan dimulai dari tiap kata atau *term* pada teks pertama.

Tabel 3. Tabel Data Sampel

No.	Teks	Label
1	truk alami patah as roda ruas jalan gatot subroto arah kuningan pancoran siang sabtu lalu lintas pantau macet	*Informative*
2	sarap murah Jakarta selatan hotel bintang reservasi promo batas *guys*	*Not informative*
3	*hidden store sneakers* jaksel *check* banget *sneakers* langka alias rare promo jalan radio jaksel *vans find us at google maps check* aja *guys*	*Not informative*
4	ranjau jari jari payung hasil sapu ruas jalan gatot subroto bpk ri patra jasa arah pancoran rabu juni imbau kendara lintas hati hati	*Informative*

$$\text{TFtruk} = \frac{1}{18}$$
$$\text{TFtruk} = 0{,}056$$
$$\dots$$
$$\text{TFmacet} = \frac{1}{18}$$
$$\text{TFmacet} = 0{,}056$$

Pada teks 1 kemunculan kata dibagi dengan total banyak kata pada teks tersebut. Perhitungan dilakukan dari kata pertama hingga terakhir. Setelah dihitung *term frequency* maka dapat dihitung *inverse document frequency*.

$$\text{IDFtruk} = \ln\left(\frac{4}{1}\right)$$
$$\text{IDFtruk} = 1{,}386$$
$$\dots$$
$$\text{IDFmacet} = \ln\left(\frac{4}{1}\right)$$
$$\text{IDFmacet} = 1{,}386$$

Pada perhitungan IDF menggunakan logaritma natural untuk mencegah perbedaan yang ekstrem antara kata-kata yang sangat umum dengan yang sangat jarang. Terakhir dilakukan perkalian antara TF dengan IDF masing-masing *term*.

$$\text{TFIDFtruk} = 0{,}056 \times 1{,}386 = 0{,}077$$
$$\text{TFIDFruas} = 0 \times 1{,}386 = 0$$
$$\text{T}F\text{IDFmacet} = 0{,}056 \times 1{,}386 = 0{,}077$$

Perhitungan TFIDF ini dilakukan hingga teks terakhir dalam data. Hasil perhitungan disusun menjadi vektor dengan banyak anggota nilai adalah jumlah seluruh *term* pada dokumen digabung dalam satu vektor tetapi hanya muncul satu kali. Letak nilai sesuai letak *term* tersebut. Tabel 4 menampilkan hasil untuk dua dari empat sampel.

Tabel 4. Contoh Hasil Perhitungan TFIDF dari Data Sampel

No	Teks	Vektor
1	Teks 1	['0,077', '0,039', '0,077', '0,039', '0,077', '0,077', '0,077', '0,039', '0,077', '0,077', '0,077', '0,016', '0,039', '0,039', '0,077', '0,077', '0,039', '0,077', '0,000']
2	Teks 2	['0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,000', '0,139', '0,139', '0,139', '0,139', '0,139', '0,139', '0,069', '0,139',

'0,069', '0,139', '0,000']

Setelah vektor TFIDF dikalkulasi, maka dapat dilanjut dengan pengklasifikasian teks menggunakan Multinomial *Naïve Bayes*. Pada perhitungan ini akan menggunakan data sampel sebelumnya. Tahap pertama dari klasifikasi Multinomial *Naïve Bayes* adalah perhitungan probabilitas prior.

$$Prior \text{ untuk label } informative$$
$$\ln\left(\frac{2}{4}\right) = -0{,}693$$
$$Prior \text{ untuk label } not\ informative$$
$$\ln\left(\frac{2}{4}\right) = -0{,}693$$

Prior dihitung dengan membagi banyaknya teks dengan suatu label dibagi dengan banyaknya teks lalu dilakukan logaritma natural. Banyak teks untuk setiap label pada sampel adalah 2 dengan total sampel adalah 4. Selanjutnya menjumlahkan setiap *term* dari suatu kelas dan menjumlahkannya lagi dengan 1 untuk *smoothing* agar tidak terjadi hasil probabilitas 0 yang dapat menyebabkan kesalahan klasifikasi.

$$term\ 1 \text{ untuk } label\ Informative = teks\ 1 + teks\ 4$$
$$= 0{,}077 + 0{,}000 = 0{,}077$$
$$= 0{,}077 + 1 = 1{,}077$$
...
$$term\ 60 \text{ untuk } label\ Informative$$
$$= 0{,}000 + 0{,}060 = 0{,}060$$
$$= 0{,}060 + 1 = 1{,}060$$

Teks yang memiliki label informative adalah teks 1 dan 4, maka keduanya dijumlahkan dari term awal hingga akhir sesuai dengan urutan pada vektor. Selanjutnya perhitungan *term* pada label kedua.

$$term\ 1 \text{ untuk } label\ not\ Informative = teks\ 2 + teks\ 3$$
$$= 0{,}000 + 0{,}000 = 0{,}000$$
$$= 0{,}000 + 1 = 1{,}000$$
...
$$term\ 60 \text{ untuk } label\ not\ Informative$$
$$= 0{,}000 + 0{,}000 = 0{,}000$$
$$= 0{,}000 + 1 = 1{,}000$$

Semua *term* yang telah di *smoothing* selanjutnya dijumlahkan sesuai label.

$$smoothed\ informative = 1{,}077 + 1{,}069 + \cdots + 1{,}060 = 61{,}252$$
$$smoothed\ not\ informative = 1{,}000 + 1{,}000 + \cdots + 1{,}000 = 61{,}526$$

Setelah semua *smoothed* sudah dijumlahkan, selanjutnya dilakukan perhitungan *likelihood* dari setiap label.

$$likelihood\ informative = \ln\left(\frac{1{,}077}{61{,}252}\right) = -4{,}041$$
...
$$\ln\left(\frac{1{,}060}{61{,}252}\right) = -4{,}057$$

$$likelihood\ not\ informative = \ln\left(\frac{1{,}000}{61{,}526}\right) = -4{,}119$$

...

$$\ln\left(\frac{1{,}000}{61{,}526}\right) = -4{,}119$$

Perhitungan *likelihood* dilakukan dengan membagi setiap *term* dibagi dengan hasil penjumlahan *smoothed* dan menggunakan logaritma natural. Perhitungan ini dilakukan pada tiap label. Terakhir adalah menghitung probabilitas posterior dari teks 1.

$$posterior\ informative =$$
$$-0{,}693 + (0{,}077 \times -4{,}041) + (0{,}039 \times -4{,}048) + \cdots + (0{,}000 \times -4{,}057) = -5{,}128$$
$$posterior\ not\ informative =$$
$$-0{,}693 + (0{,}000 \times -4119) + (0{,}000 \times -4{,}119) + \cdots + (0{,}000 \times -4{,}119) = -5{,}212$$

Perhitungan posterior dilakukan dengan mengalikan semua *term* pada teks dengan *likelihood* dan menambahkannya dengan *prior*. Nilai *likelihood* dan *prior* menggunakan hasil perhitungan sebelumnya, sementara nilai *term* menyesuaikan dengan teks yang akan diklasifikasi. Pada teks 1 didapat posterior label "*informative*" adalah -5,128 dan posterior "*not informative*" adalah -5,212. Karena posterior *informative* adalah yang terbesar, maka teks 1 diklasifikasikan sebagai *informative*.

3.3 Pengujian Klasifikasi

Hasil dari kedua evaluasi menunjukkan performa yang cukup baik. Pada hasil C*onfusion Matrix* menunjukkan sebanyak 489 klasifikasi adalah benar, sedangkan 39 klasifikasi lainnya salah. Dari hasil dapat disimpulkan bahwa model memiliki performa yang cukup baik dalam mengklasifikasi teks media sosial. Rincian hasil ini dapat dilihat pada tabel 5.

Tabel 5. Tabel *Confusion Matrix*

	Not informative	*Informative*	Total Klasifikasi benar	Total Klasifikasi salah
Not informative	239	23	489	39
Informative	16	250		

Hasil *Evaluation Matrix* juga menunjukkan performa yang cukup baik. Dari hasil metriks ditunjukkan bahwa model ini memiliki persentase akurasi yang cukup baik dalam mengklasifikasi teks. Rincian hasil metriks ditampilkan pada tabel 6.

Tabel 6. Tabel *Evaluation Metrics*

Naive Bayes				
Label	*Precision*	*Recall*	*F1-Score*	*Support*
0	0,94	0,91	0,92	262
1	0,92	0,94	0,93	266
Macro Avg	0,93	0,93	0,93	528
Akurasi: 0,926				

Secara keseluruhan, dari hasil *Confusion Matrix* dan *Evaluation Metrics* menunjukkan bahwa model ini memiliki performa yang cukup baik dalam mengklasifikasi teks media sosial. Hal ini disebabkan oleh teks pada data telah digeneralisir sehingga memudahkan proses pelatihan model. Dengan demikian, model ini cukup baik untuk mendukung keakuratan informasi untuk sistem monitoring ini.

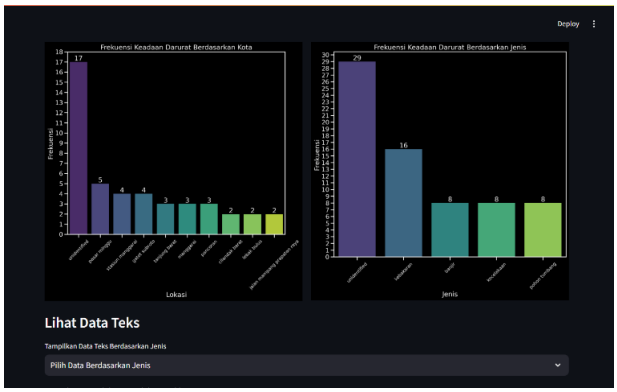
3.4 Fitur dan Visualisasi

Hasil fitur dan visualisasi dari sistem monitoring ini memungkinkan pengguna untuk memantau lokasi keadaan darurat di peta Jakarta Selatan melalui peta *OpenStreetMap*. Peta ini menampilkan berbagai elemen visual seperti marka titik, garis, dan lingkaran dengan radius 500 meter, yang membantu dalam mengidentifikasi kemungkinan area secara lebih jelas. Untuk penempatan marka, digunakan API *Overpass Turbo* yang menyediakan koordinat berdasarkan

data lokasi yang telah diekstraksi sebelumnya. Tampilan hasil visualisasi disajikan pada gambar 1.

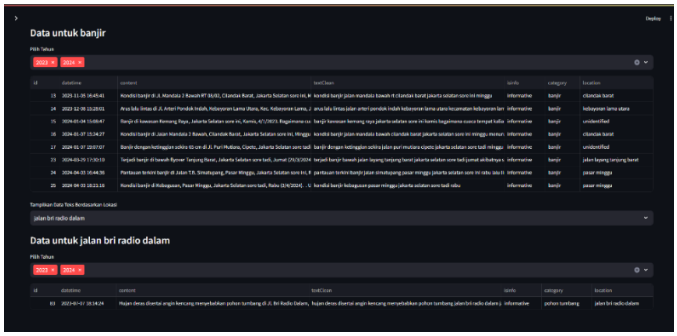
Gambar 1. Tampilan Visualisasi Peta

Marka titik pada gambar 1 menunjukkan informasi dari keadaan darurat yang terjadi. Sementara marka garis dan lingkaran menunjukkan perkiraan lokasi lebih detail berdasarkan informasi pada teks. Jika pada teks disebutkan titik spesifik seperti sekitar gedung atau objek lokasi maka akan ditampilkan lingkaran. Jika tidak maka akan ditampilkan garis sepanjang jalan sesuai nama dan koordinat. Selain peta, ditampilkan juga grafik frekuensi keadaan darurat. Tampilan dari fitur tersebut disajikan pada gambar 2.



Gambar 2. Tampilan Grafik Sebagai Fitur Pendukung Visualisasi

Fungsi grafik pada gambar 2 adalah sebagai fitur pendukung dari visualisasi. Tujuannya sebagai analisis untuk pengguna dalam melihat lokasi mana yang sering terjadi. Grafik jenis bertujuan untuk melihat kejadian apa yang sering terjadi di Jakarta Selatan. Disediakan juga fitur untuk melihat data teks seperti yang ditampilkan pada gambar 3.



Gambar 3. Tampilan Data Teks Sebagai Fitur Pendukung Visualisasi

Data teks pada gambar 3 juga berfungsi sebagai fitur pendukung dari visualisasi. Tujuannya agar pengguna mengetahui teks yang dijadikan sebagai acuan sistem *monitoring* ini. Data ini

menunjukkan teks asli dan hasil pembersihan tanpa tokenisasi, hasil *labeling*, serta hasil ekstraksi lokasi dan jenis.

4. KESIMPULAN

Sistem monitoring ini dapat memetakan perkiraan lokasi yang disebutkan pada teks. Sistem ini juga dapat memberikan grafik jumlah jenis kejadian yang sering terjadi dan jumlah lokasi yang sering mengalami kejadian. Pemetaan dilakukan menggunakan *OpenStreetMap* dengan API *Overpass Turbo* untuk visualisasi titik lokasi kejadian pada peta.

Klasifikasi teks *caption* unggahan Instagram mengenai keadaan darurat mendapatkan hasil akurasi sebesar 92,6%. Akurasi tersebut adalah hasil dari pembersihan lebih lanjut terhadap kata-kata tidak teratur menjadi kata dasar. Dari pembersihan ini data teks menjadi tidak terlalu beragam dan memudahkan proses pembelajaran model.

Penggunaan *Large Language Model* (LLM) dapat membantu dalam menyederhanakan model. Dengan LLM model dapat memperoleh pemahaman yang lebih baik terhadap konteks suatu kata. Kombinasi dari metode pembersihan dan penggunaan LLM dapat meningkatkan hasil klasifikasi untuk sistem ini.

DAFTAR PUSTAKA

[1] D. Zeng, H. Chen, R. Lusch, and S.-H. Li, "Social Media Analytics and Intelligence," *IEEE Intell. Syst.*, vol. 25, no. 6, pp. 13–16, 2010.

[2] W. Gao, L. Li, X. Zhu, and Y. Wang, "Detecting Disaster-Related Tweets Via Multimodal Adversarial Neural Network," *IEEE Multimed.*, vol. 27, no. 4, pp. 28–37, 2020.

[3] W. Gata, F. Amsury, N. K. Wardhani, I. Sugiyarto, D. N. Sulistyowati, and I. Saputra, "Informative Tweet Classification of the Earthquake Disaster Situation In Indonesia," in *2019 5th International Conference on Computing Engineering and Design (ICCED)*, Singapore, Singapore, Apr. 2019, pp. 1–6.

[4] "Digital 2024," *Digital 2024-We Are Social Indonesia*, Jan. 31, 2024. https://wearesocial.com/id/blog/2024/01/digital-2024/ (accessed Apr. 26, 2024).

[5] "Infografis Kejadian Bencana 2023," *Infografis Kejadian Bencana 2023*, 2024. https://bpbd.jakarta.go.id/perpustakaan/220/infografis-kejadian-bencana-2023 (accessed Apr. 26, 2024).

[6] H. Krishnan, A. Roy, A. K. Menon, D. S, and H. M. Babu, "Natural Disaster Detection Using Social Media," in *2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, Ernakulam, India, 2023, pp. 1–6.

[7] K. M. A. Pasaribu, R. E. Saputra, and C. Setianingsih, "Sistem Informasi Monitoring Bencana Alam Dari Data Media Sosial Menggunakan Metode K-Nearest Neighbor," *E-Proceeding Eng.*, vol. 8, Aug. 2021.

[8] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi, "Text Mining in Big Data Analytics," *Big Data Cogn. Comput.*, vol. 4, no. 1, pp. 1-34, Jan. 2020.

[9] I. Afdhal, R. Kurniawan, I. Iskandar, R. Salambue, E. Budianita, and F. Syafria, "Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia," vol. 5, no. 1, pp. 122-130, 2022.

[10] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations," *Organ. Res. Methods*, vol. 25, no. 1, pp. 114–146, 2022.

[11] A. N. Azizah, M. Falach Asy'ari, I. Wisma Dwi Prastya, and D. Purwitasari, "Easy Data Augmentation untuk Data yang Imbalance pada Konsultasi Kesehatan Daring," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 10, no. 5, pp. 1095–1104, 2023.

[12] A. Sabrani and J. Majapahit, "Metode Multinomial Naïve Bayes Untuk Klasifikasi Artikel Online Tentang Gempa Di Indonesia," vol. 2, no. 1, pp. 89-100, 2020.

[13] N. L. Octaviani, E. Hari Rachmawanto, C. A. Sari, and I. M. S. De Rosal, "Comparison of Multinomial Naïve Bayes Classifier, Support Vector Machine, and Recurrent Neural

Network to Classify Email Spams," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, Indonesia, Sep. 2020, pp. 17–21.

[14] S. Kadam, A. Gala, P. Gehlot, A. Kurup, and K. Ghag, "Word Embedding Based Multinomial Naive Bayes Algorithm for Spam Filtering," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, Aug. 2018, pp. 1–5.

[15] D. Berrar, "Bayes' Theorem and Naive Bayes Classifier," in *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, 2018, pp. 403–412.

[16] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," vol. 5, no. 2, pp. 697-711, 2021.

[17] H. Dalianis, "Evaluation Metrics and Evaluation," in *Clinical Text Mining*, Cham: Springer International Publishing, 2018, pp. 45–53.