INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v21i1.8642

VOL. 21. NO. 1 AGUSTUS 2026
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Universitas Nusa Mandiri

SISTEM DETEKSI BANJIR DI KABUPATEN BEKASI BERBASIS DATA TWITTER MENGGUNAKAN SUPPORT VECTOR MACHINE

Reza Okta Pratama[1*]; Nardi[2]; Anton Widodo[3]; Marzuki Sinambela[4]

Instrumentasi MKG[1,2,3,4]
Sekolah Tinggi Meteorologi Klimatologi dan Geofisika, Kota Tangerang, Indonesia [1,2,3,4]
www.stmkg.ac.id[1,2,3,4]
rezaoktapp@gmail.com[1*], nardi@stmkg.ac.id[2], anton.widodo@stmkg.ac.id[3], sinambela.m@gmail.com[4]

(*) Corresponding Author

Abstract— *Floods are the most common hydrometeorological disasters in Indonesia, and Bekasi Regency is among the areas with a high level of vulnerability. This study aims to develop a Twitter-based flood detection system using Support Vector Machines (SVM), with Logistic Regression (LR) and Random Forest (RF) as comparison models. A total of 4,436 tweets from the 2020–2024 period were collected using Tweet Harvest, manually labeled, and then processed through preprocessing, TF-IDF feature extraction, and data splitting using group-based random splitting (80:20). Evaluation was conducted using 10-fold GroupKFold cross-validation with recall as the primary metric, followed by hyperparameter tuning using GridSearchCV. The evaluation results showed that SVM performed best compared to LR and RF. The tuned SVM model achieved a test recall of 0.8673, exceeding the minimum threshold of 0.80. The model was then integrated into a web-based monitoring dashboard that displays interactive maps, statistics, and temporal trends. Black-box testing across nine scenarios achieved a 100% success rate, while a user experience evaluation using the UEQ-S on 18 respondents yielded an overall score of 1.97, categorized as "Very Good." The research results indicate that the developed system is capable of effectively supporting social media-based flood monitoring.*

Keywords*: Flood Detection, Support Vector Machine, Text Mining, TF-IDF, Twitter*

Abstrak— Banjir merupakan bencana hidrometeorologi yang paling sering terjadi di Indonesia, sedangkan Kabupaten Bekasi termasuk wilayah dengan tingkat kerentanan yang tinggi. Penelitian ini bertujuan mengembangkan sistem deteksi banjir berbasis data Twitter menggunakan Support Vector Machine (SVM) dengan Logistic Regression (LR) dan Random Forest (RF) sebagai model pembanding. Sebanyak 4.436 *tweet* periode 2020–2024 dikumpulkan menggunakan *Tweet* Harvest, dilabeli secara manual, kemudian diproses melalui tahapan *preprocessing*, ekstraksi fitur TF-IDF, dan pembagian data menggunakan *group-based random splitting* (80:20). Evaluasi dilakukan menggunakan 10-fold GroupKFold *cross-validation* dengan *recall* sebagai metrik utama, dilanjutkan *hyperparameter tuning* menggunakan GridSearchCV. Hasil evaluasi menunjukkan SVM memberikan performa terbaik dibandingkan LR dan RF. Model SVM hasil *tuning* mencapai *test recall* sebesar 0,8673, melampaui ambang minimum 0,80. Model kemudian diintegrasikan ke dalam *dashboard* pemantauan berbasis web yang menampilkan peta interaktif, statistik, dan tren temporal. Pengujian *black box* pada sembilan skenario memperoleh tingkat keberhasilan 100%, sedangkan evaluasi pengalaman pengguna menggunakan UEQ-S terhadap 18 responden menghasilkan skor keseluruhan 1,97 dengan kategori Sangat Baik. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu mendukung pemantauan banjir berbasis media sosial secara efektif.

Kata Kunci: *Deteksi Banjir, Support Vector Machine, Text Mining, TF-IDF, Twitter*

PENDAHULUAN

Banjir merupakan bencana hidrometeorologi yang paling sering terjadi di Indonesia dan menimbulkan kerugian signifikan pada sektor sosial, ekonomi, serta infrastruktur. Pada tahun 2024, dari 3.472 kejadian bencana di Indonesia, sebanyak 1.420 kejadian atau sekitar 41%

merupakan banjir yang menyebabkan lebih dari 6,3 juta jiwa terdampak maupun mengungsi (BNPB, 2025). Meningkatnya kejadian bencana hidrometeorologi tersebut tidak terlepas dari dampak perubahan iklim yang terjadi di berbagai wilayah dunia, termasuk Indonesia (Azizah dkk., 2022).

Kabupaten Bekasi merupakan salah satu wilayah dengan tingkat kerawanan banjir tinggi akibat kombinasi faktor alam dan antropogenik, seperti urbanisasi, berkurangnya kapasitas serapan lahan, serta pertumbuhan kawasan industri (Agustina, 2021). Berdasarkan data BMKG (2021) periode 1991–2020, Kabupaten Bekasi memiliki rata-rata curah hujan tahunan sebesar 1.767 mm dengan 84 hari hujan per tahun. Selain itu, sekitar 72% wilayahnya berada pada ketinggian 0–25 meter di atas permukaan laut sehingga rentan terhadap limpasan permukaan dan banjir (Agustina, 2021).

Pemantauan banjir secara konvensional yang mengandalkan sensor hidrologi maupun laporan manual masih menghadapi keterbatasan dalam cakupan spasial, biaya operasional, dan akurasi data (Segovia-Cardozo dkk., 2023). Perkembangan teknologi digital memungkinkan pemanfaatan media sosial sebagai sumber data *crowdsourcing* yang melengkapi sensor fisik, terutama pada wilayah dengan keterbatasan cakupan sensor (Yang dkk., 2022; Esparza dkk., 2024). Selain itu, jumlah unggahan media sosial mengenai banjir memiliki korelasi temporal yang signifikan dengan data curah hujan aktual sehingga mendukung penerapan pendekatan *social sensing* untuk pemantauan banjir (Yu & Wang, 2024).

Di antara berbagai media sosial, Twitter (kini X) memiliki karakteristik yang mendukung pemantauan bencana secara mendekati *real-time*. Moghadas dkk. (2023) menunjukkan bahwa Twitter dapat dimanfaatkan untuk memantau kejadian banjir melalui analisis teks, sedangkan Veigel dkk. (2025) menunjukkan bahwa lonjakan aktivitas Twitter berkorelasi dengan tahap banjir aktual sehingga berpotensi dimanfaatkan sebagai sistem *crowdsourcing* untuk memantau kondisi lapangan.

Berbagai penelitian telah memanfaatkan Twitter untuk klasifikasi informasi kebencanaan dan cuaca. Caniago & Habibi (2023) menggunakan SVM dengan pembobotan TF-IDF untuk mengklasifikasikan tingkat keparahan banjir di Indonesia dan memperoleh akurasi sebesar 90,61%. Purwandari dkk. (2023) mengembangkan sistem *crawling* informasi cuaca berbasis Twitter menggunakan SVM dan TF-IDF dengan akurasi sebesar 85%. Pada penelitian lain, Random Forest sedikit lebih unggul dibandingkan SVM untuk analisis sentimen metaverse (Sari & Suryono, 2024), sedangkan Logistic Regression menunjukkan performa lebih baik dibandingkan Linear SVM dan Random Forest pada klasifikasi persepsi publik mengenai banjir di Sumatera (Ristiana & Mutmainah, 2026). Perbedaan performa tersebut menunjukkan bahwa keunggulan suatu algoritma bersifat *dataset-dependent*, sehingga evaluasi komparatif tetap diperlukan pada konteks data dan wilayah yang berbeda. Dalam konteks deteksi banjir, kemampuan model untuk mendeteksi sebanyak mungkin kejadian banjir (*recall*) menjadi aspek yang diprioritaskan karena kegagalan mendeteksi kejadian (*false negative*) dapat menyebabkan masyarakat terdampak tidak memperoleh informasi secara tepat waktu (Thieken dkk., 2023). Oleh karena itu, penelitian ini menggunakan *recall* sebagai metrik evaluasi utama dalam membandingkan performa algoritma.

Meskipun demikian, data Twitter berupa teks yang singkat, tidak terstruktur, dan banyak menggunakan bahasa informal, emotikon, serta emoji sehingga memerlukan tahapan *preprocessing* yang tepat agar proses klasifikasi lebih akurat (Palomino & Aider, 2022; Aryanti & Suria, 2025). Selain itu, integrasi model *machine learning* dengan sistem berbasis web telah terbukti mendukung proses pemantauan melalui *dashboard* dan visualisasi data (Marzouk dkk., 2022).

Berdasarkan penelitian terdahulu, masih terdapat beberapa kesenjangan penelitian. Sebagian besar penelitian berfokus pada pengembangan model klasifikasi secara *offline* tanpa integrasi dengan sistem pemantauan berbasis web yang mendukung pemantauan *real-time*. Selain itu, belum ditemukan sistem pemantauan banjir berbasis Twitter yang secara khusus dikembangkan untuk Kabupaten Bekasi. Evaluasi komparatif antara Support Vector Machine (SVM), Logistic Regression (LR), dan Random Forest (RF) dengan penekanan pada metrik *recall* untuk klasifikasi *tweet* banjir *real-time* juga masih belum dilakukan secara spesifik dalam konteks Indonesia.

Oleh karena itu, penelitian ini bertujuan mengembangkan sistem pemantauan banjir berbasis Twitter untuk Kabupaten Bekasi yang mengintegrasikan proses *crawling*, klasifikasi *tweet*, dan visualisasi hasil dalam *dashboard* berbasis web. Penelitian mengevaluasi tiga algoritma klasifikasi, yaitu SVM sebagai kandidat utama serta LR dan RF sebagai model pembanding. SVM dipilih karena kemampuannya menangani data teks berdimensi tinggi yang direpresentasikan menggunakan TF-IDF (Aryanti & Suria, 2025), sedangkan LR dipilih karena efisiensi komputasi dan kemampuannya bekerja pada pola linear (Hassan dkk., 2022). RF

INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v21i1.8642

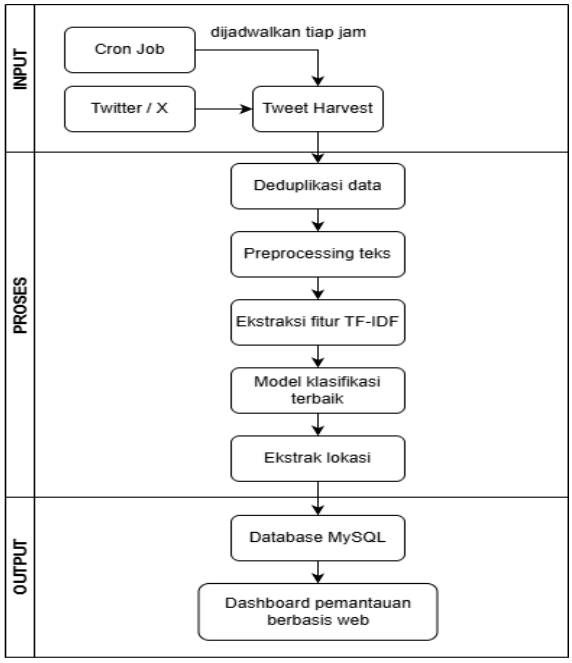
VOL. 21. NO. 1 AGUSTUS 2026
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Universitas Nusa Mandiri

digunakan sebagai pembanding karena memiliki ketahanan yang baik terhadap *class noise* (Alharbi, 2024). Kontribusi penelitian ini terletak pada pengembangan sistem pemantauan banjir yang mengintegrasikan pengumpulan data Twitter, klasifikasi menggunakan tiga algoritma, serta penyajian informasi spasial dan temporal melalui *dashboard* berbasis web untuk mendukung pemantauan banjir di Kabupaten Bekasi.

BAHAN DAN METODE

Penelitian ini menggunakan pendekatan kuantitatif dengan desain eksperimental komparatif untuk mengembangkan sistem pemantauan banjir berbasis Twitter di Kabupaten Bekasi. Penelitian terdiri atas tiga tahapan utama, yaitu pengumpulan data, pembangunan model klasifikasi melalui evaluasi komparatif tiga algoritma (SVM, LR, dan RF), serta implementasi model terbaik pada *dashboard* pemantauan berbasis web.

Arsitektur sistem terdiri atas modul input, pemrosesan, dan output. Modul input melakukan pengumpulan data dari Twitter/X menggunakan *Tweet* Harvest, yang dipicu secara berkala melalui mekanisme *cron job*. Selanjutnya, data diproses melalui tahapan deduplikasi, *preprocessing*, ekstraksi fitur TF-IDF, klasifikasi, dan ekstraksi lokasi. Hasil klasifikasi disimpan ke dalam basis data dan ditampilkan melalui *dashboard* pemantauan berbasis web. Arsitektur sistem secara keseluruhan ditunjukkan pada Gambar 1.



Sumber: (Hasil Penelitian, 2026)

Gambar 1. Arsitektur Sistem

Sumber Data dan Pengumpulan Data

Data penelitian bersumber dari platform media sosial Twitter (X) yang dikumpulkan melalui *web scraping* menggunakan *Tweet* Harvest dengan *auth token* API X. Pengumpulan data mencakup periode 2020–2024. Proses *crawling* dilakukan secara bertahap menggunakan fitur *until* untuk membatasi rentang waktu pada setiap sesi dan fitur *max_id* untuk melanjutkan pengumpulan dari ID *tweet* terakhir yang diambil sehingga mencegah duplikasi maupun kesenjangan data, dengan strategi pencarian berupa kombinasi kata kunci "banjir" dan nama 23 kecamatan di Kabupaten Bekasi. Strategi tersebut dipilih untuk memfokuskan proses *crawling* pada *tweet* yang secara eksplisit membahas kejadian banjir di wilayah penelitian sekaligus meminimalkan data yang tidak relevan (*noise*).

Dari proses tersebut diperoleh 4.436 *tweet* yang kemudian diberi label secara manual menjadi dua kategori, yaitu *tweet* banjir *real-time* (label 1) dan *tweet* non-*real-time*/non-banjir (label 0). Proses pelabelan dilakukan secara manual oleh satu anotator. Setiap *tweet* dievaluasi secara individual dengan mengacu pada isi *tweet* untuk menilai konteks informasi, waktu kejadian, serta penyebutan lokasi yang relevan dengan Kabupaten Bekasi sebelum ditetapkan sebagai *tweet real-time* atau *non-real-time*. *Tweet* dikategorikan sebagai *real-time* apabila melaporkan kejadian banjir saat unggahan dipublikasikan, secara eksplisit menggunakan kata banjir dalam konteks bencana, bukan berasal dari akun spam, serta memuat informasi lokasi yang relevan dengan Kabupaten Bekasi. Sebaliknya, *tweet* yang tidak memenuhi kriteria tersebut, termasuk konten promosi, penggunaan metaforis kata banjir, berita historis, peringatan umum, maupun laporan dari wilayah administratif lain dengan nama lokasi yang serupa, dikategorikan sebagai non-*real-time*. Distribusi dataset hasil pelabelan disajikan pada Tabel 1.

Tabel 1. Distribusi Label dalam Dataset

Label	Kategori	Jumlah	Persentase
0	Non-banjir / Non-*real-time*	3412	76,92%
1	Banjir *real-time*	1.024	23,08%
Total		4.436	100%

Sumber: (Hasil Penelitian, 2026)

Distribusi tersebut menunjukkan ketidakseimbangan kelas dengan rasio sekitar 3,3:1 antara *tweet* non-banjir dan *tweet* banjir *real-time*. Dataset penelitian ini tidak dipublikasikan secara terbuka karena diperoleh dari platform Twitter (X), sehingga pendistribusiannya perlu memperhatikan ketentuan layanan (*Terms of Service*) dan kebijakan

penggunaan data dari platform tersebut. Meskipun demikian, penelitian ini menjelaskan secara rinci sumber data, strategi *crawling*, proses pelabelan, tahapan *preprocessing*, serta metode pengembangan model sehingga prosedur penelitian dapat diterapkan kembali pada data yang dikumpulkan sesuai dengan kriteria yang sama.

Analisis Data

Analisis data diawali dengan *preprocessing* terhadap seluruh data berlabel yang meliputi text cleaning, *case folding*, *tokenization*, slang normalization, *stopword removal*, dan *stemming* menggunakan algoritma Sastrawi. Setelah *preprocessing*, satu data dihapus karena menghasilkan teks kosong sehingga dataset akhir berjumlah 4.435 *tweet*. Selanjutnya, data dibagi menggunakan metode group-based random splitting dengan rasio 80:20 untuk memastikan teks yang identik hanya muncul pada satu subset dan mencegah data leakage (Kapoor & Narayanan, 2023). Ekstraksi fitur dilakukan menggunakan TF-IDF Vectorizer yang di-fit secara eksklusif pada data pelatihan dengan konfigurasi max_*feature*s = 1.000, n-gram range = (1,2), min_df = 2, max_df = 0,8, sublinear_tf = True, dan normalisasi L2.

Tiga algoritma klasifikasi, yaitu Support Vector Machine (SVM), Logistic Regression (LR), dan Random Forest (RF), digunakan untuk membangun model klasifikasi. Seluruh algoritma menerapkan mekanisme *class_weight = balanced* untuk mengurangi pengaruh ketidakseimbangan kelas melalui pemberian bobot yang berbanding terbalik terhadap frekuensi masing-masing kelas. Pendekatan ini merupakan implementasi *cost-sensitive learning* yang memberikan penalti lebih besar terhadap kesalahan klasifikasi pada kelas minoritas dibandingkan kelas mayoritas (Thölke dkk., 2023).

Pelatihan dan Evaluasi Model

Sebelum proses pelatihan, ditemukan sebanyak 600 data (17,0%) pada data pelatihan memiliki representasi teks yang identik setelah *preprocessing*. Duplikat tersebut tidak dihapus, tetapi ditangani menggunakan strategi GroupKFold selama *cross-validation* untuk mencegah kebocoran informasi antar-fold. Evaluasi awal terhadap ketiga algoritma dilakukan menggunakan 10-fold GroupKFold *cross-validation* dengan *recall* sebagai metrik evaluasi utama, mengingat konteks deteksi banjir menempatkan kegagalan deteksi sebagai risiko yang lebih kritis dibandingkan alarm palsu. Ambang batas minimum *recall* sebesar 0,80 ditetapkan sebagai target kinerja minimum selama proses seleksi model.

Berdasarkan hasil evaluasi awal, algoritma dengan nilai *test recall* tertinggi dipilih untuk menjalani proses *hyperparameter tuning*. Pada tahap ini, GridSearchCV menggunakan nilai *recall* sebagai fungsi objektif (*scoring*), sedangkan kombinasi *hyperparameter* terbaik ditentukan berdasarkan rata-rata (*mean recall*) yang diperoleh dari seluruh *fold* selama proses 10-fold *GroupKFold cross-validation*. Setelah kombinasi *hyperparameter* terbaik diperoleh, model dilatih ulang menggunakan seluruh *training set* dan dievaluasi pada *test set* sebagai data evaluasi independen.

Implementasi *Pipeline* Klasifikasi

Implementasi *pipeline* klasifikasi dilakukan menggunakan Python dan dijalankan secara otomatis melalui *cron job*. *Pipeline* mencakup proses *crawling* data Twitter, deduplikasi, *preprocessing*, klasifikasi menggunakan model yang telah dipilih, ekstraksi lokasi, serta penyimpanan hasil klasifikasi ke dalam basis data.

Ekstraksi lokasi dilakukan menggunakan *regular expression* (*regex*) pada teks *tweet* sebelum *preprocessing* untuk mengenali nama kecamatan di Kabupaten Bekasi. Proses ini menggunakan *dictionary* yang berisi 23 nama kecamatan beserta koordinat geografisnya. Nama kecamatan yang berhasil dikenali kemudian dipetakan ke koordinat geografis sehingga *tweet* yang terdeteksi sebagai banjir *real-time* dapat direpresentasikan secara spasial. Selanjutnya, hasil klasifikasi yang diprediksi sebagai banjir *real-time* beserta informasi lokasi disimpan ke dalam basis data MySQL yang terdiri atas tabel data historis dan tabel monitoring *real-time*.

Pengembangan sistem dilakukan menggunakan Python 3.12.3 dengan pustaka utama meliputi *pandas* dan *NumPy* untuk pengolahan data, *NLTK* untuk *tokenization* dan *stopword removal*, *Sastrawi* untuk *stemming* bahasa Indonesia, *scikit-learn* untuk ekstraksi fitur TF-IDF, pelatihan, dan evaluasi model *machine learning*, serta *mysql-connector-python* untuk integrasi dengan basis data MySQL.

Untuk memastikan sistem yang diimplementasikan berfungsi sesuai tujuan, dilakukan evaluasi melalui dua pendekatan. Pertama, pengujian *black box* terhadap seluruh fitur utama *dashboard* untuk memverifikasi fungsionalitas sistem dari perspektif pengguna dengan ambang batas keberhasilan minimum sebesar 95%. Kedua, evaluasi pengalaman pengguna menggunakan instrumen User Experience Questionnaire Short (UEQ-S) terhadap 18 responden yang berdomisili di Kabupaten

INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v21i1.8642

VOL. 21. NO. 1 AGUSTUS 2026
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Universitas Nusa Mandiri

Bekasi, mencakup dua dimensi, yaitu Pragmatic Quality dan Hedonic Quality.

HASIL DAN PEMBAHASAN

Hasil *Preprocessing*

Pipeline preprocessing yang diterapkan pada 4.436 *tweet* berhasil menghasilkan representasi teks yang lebih bersih dan konsisten. Dari keseluruhan data awal, satu *tweet* dihapus karena menghasilkan teks kosong setelah seluruh tahapan *preprocessing*, sehingga dataset akhir yang digunakan pada tahap klasifikasi berjumlah 4.435 *tweet*. Contoh transformasi teks pada setiap tahapan *preprocessing* disajikan pada Tabel 2.

Berdasarkan Tabel 2, tahapan *case folding* dan cleaning bertujuan menyeragamkan seluruh teks menjadi huruf kecil serta menghilangkan elemen yang tidak relevan (noise), seperti mention, URL, dan karakter selain huruf. Selanjutnya, *tokenization* memecah teks menjadi unit kata agar setiap token dapat diproses secara individual (Caniago & Habibi, 2023; Sari & Suryono, 2024). Tahap slang normalization mengubah kata tidak baku menjadi bentuk standar, sedangkan *stopword removal* menghilangkan kata-kata yang memiliki kontribusi rendah terhadap proses klasifikasi. Tahap *stemming* kemudian mereduksi kata berimbuhan menjadi bentuk dasar sehingga menghasilkan representasi teks yang lebih seragam dan relevan untuk proses klasifikasi.

Tabel 2. Hasil *Preprocessing* Teks

Tahap	Hasil
Teks Asli	"Semoga cepet surut banjirnya 😭 #Banjir Cibitung-Bekasi https://..."
Case folding	semoga cepet surut banjirnya #banjir cibitung-bekasi
Cleaning	semoga cepet surut banjirnya banjir cibitung bekasi
Tokenization	['semoga','cepet','surut','banjirnya','banjir','cibitung','bekasi']
Normalization	['semoga','cepet','surut','banjirnya','banjir','cibitung','bekasi']
Stopword removal	['semoga','cepet','surut','banjirnya','banjir','cibitung','bekasi']
Stemming	moga cepet surut banjir banjir cibitung bekas

Sumber: (Hasil Penelitian, 2026)

Sebagaimana ditunjukkan pada Tabel 2, proses *preprocessing* menghasilkan perubahan bentuk kata sesuai dengan fungsi setiap tahapan. Pada tahap *stemming*, kata *banjirnya* direduksi menjadi *banjir*, sedangkan nama lokasi seperti Bekasi berubah menjadi *bekas*. Perubahan tersebut merupakan karakteristik algoritma Nazief–Adriani yang tidak membedakan kata umum dan entitas bernama. Temuan ini sejalan dengan penelitian Sinaga & Nainggolan (2023), yang melaporkan perubahan serupa pada nama tempat, seperti "Maluku" menjadi "malu" dan "Kuba" menjadi "ba". Meskipun demikian, penelitian tersebut menunjukkan bahwa algoritma Nazief–Adriani tetap memiliki akurasi rata-rata yang tinggi, yaitu 97,73% berdasarkan pengujian terhadap 30 dokumen teks berbahasa Indonesia. Distorsi pada nama lokasi tersebut tidak memengaruhi proses ekstraksi lokasi pada penelitian ini karena ekstraksi lokasi dilakukan menggunakan teks tweet asli sebelum tahap *preprocessing*. Perubahan tersebut juga tidak berdampak signifikan terhadap proses klasifikasi karena model memanfaatkan representasi keseluruhan konteks kejadian banjir, bukan nama lokasi sebagai fitur utama.

Evaluasi Model *Baseline*

Recall dipilih sebagai metrik evaluasi utama karena dalam konteks deteksi banjir, kegagalan mendeteksi kejadian banjir (*false negative*) merupakan risiko yang jauh lebih kritis dibandingkan alarm palsu. Thieken dkk. (2023) menemukan bahwa warga terdampak yang tidak menerima peringatan menghadapi keterlambatan evakuasi, sementara Reinert dkk. (2025) menunjukkan bahwa kegagalan menerjemahkan informasi prakiraan banjir menjadi tindakan tepat waktu terbukti menghambat respons bencana secara signifikan. Ambang batas minimum *recall* sebesar 0,80 ditetapkan sebagai target kinerja minimum agar model tetap mampu mendeteksi sebagian besar kejadian banjir (*real-time*) pada data uji. Nilai tersebut dipilih dengan mempertimbangkan dua aspek, yaitu konsekuensi operasional dari *false negative* pada sistem deteksi banjir serta capaian *recall* penelitian terdahulu yang berada pada kisaran 0,8295 hingga 0,88 (Adili & Chen, 2024; Mansoor dkk., 2025; Sufi & Khalil, 2022). Dengan mempertimbangkan karakteristik dataset penelitian yang memiliki ketidakseimbangan kelas lebih tinggi, nilai 0,80 ditetapkan sebagai target kinerja minimum yang tetap realistis untuk dicapai sekaligus mampu mempertahankan kemampuan deteksi kejadian banjir.

Evaluasi *baseline* ketiga algoritma (SVM, LR, dan RF) dilakukan menggunakan 10-Fold GroupKFold *cross-validation* pada *training set* dan evaluasi pada *test set* independen. Hasil evaluasi menunjukkan bahwa SVM menghasilkan performa terbaik dengan CV *recall* rata-rata sebesar 0,8583 (±0,0256) dan *test recall* 0,8520, diikuti oleh LR dengan CV *recall* 0,8438 (±0,0368) dan *test recall* 0,8265. Selain memiliki *recall* yang lebih tinggi, SVM juga menunjukkan standar deviasi yang lebih rendah dibandingkan LR, sehingga mengindikasikan performa yang lebih stabil antar-

fold. Meskipun pemilihan model didasarkan pada nilai *recall* sesuai tujuan penelitian, metrik *precision*, *F1-score*, *ROC-AUC*, dan *PR-AUC* juga dianalisis untuk memberikan evaluasi performa yang lebih komprehensif.

Sebaliknya, RF memperoleh performa yang jauh lebih rendah dengan CV *recall* sebesar 0,6125 (±0,0599) dan *test recall* 0,5918. Kesenjangan ini dapat dikaitkan dengan karakteristik representasi TF-IDF yang menghasilkan matriks *sparse* berdimensi tinggi. Ghosh & Cabrera (2021) menjelaskan bahwa ketika hanya sebagian kecil fitur yang benar-benar informatif, pemilihan *feature subspace* secara acak pada setiap node menyebabkan rata-rata fitur informatif yang terpilih menjadi sangat sedikit sehingga akurasi setiap *decision tree* dan performa *ensemble* menurun. Kondisi tersebut relevan dengan penelitian ini karena matriks TF-IDF memiliki sparsity sebesar 98,48%, sehingga peluang random sampling memilih fitur informatif menjadi kecil dan sensitivitas model terhadap kelas minoritas berkurang. Sebaliknya, SVM dioptimalkan untuk menemukan hyperplane *maximum-margin* (Prosise, 2022) dan mampu menangani pola nonlinier melalui fungsi *kernel* (Valkenborg dkk., 2023), sehingga lebih sesuai untuk ruang fitur TF-IDF yang *sparse* dan berdimensi tinggi. Berdasarkan hasil tersebut, SVM dipilih sebagai algoritma yang dilanjutkan ke tahap *hyperparameter tuning*.

Hasil Tuning *Hyperparameter*

Tuning *hyperparameter* dilakukan pada model SVM yang dipilih berdasarkan nilai *test recall* tertinggi pada tahap evaluasi *baseline*. Proses GridSearchCV menghasilkan kombinasi *hyperparameter* dengan nilai *mean recall* tertinggi selama 10-fold *GroupKFold cross-validation*, yang selanjutnya dipilih sebagai model terbaik untuk dievaluasi pada *test set* independen. Karena fungsi *scoring* yang digunakan pada GridSearchCV adalah *recall*, proses optimasi diarahkan untuk memaksimalkan kemampuan model dalam mendeteksi kelas banjir (*real-time*) tanpa mempertimbangkan metrik evaluasi lain sebagai kriteria pemilihan model. Ruang pencarian *hyperparameter* yang digunakan disajikan pada Tabel 3.

Tabel 3. Ruang Pencarian *Hyperparameter* GridSearchCV

Hyperparameter	Nilai yang Diuji
C (regularisasi)	0,1; 1; 10
kernel	rbf, linear
gamma	scale, auto
class_weight	None, balanced

Sumber: (Hasil Penelitian, 2026)

GridSearchCV menghasilkan kombinasi *hyperparameter* terbaik dengan CV *recall* sebesar 0,8907, meningkat dari 0,8583 pada model *baseline*. Model terbaik selanjutnya dilatih ulang menggunakan seluruh *training set* dan dievaluasi pada *test set* independen. Konfigurasi *hyperparameter* terbaik disajikan pada Tabel 4.

Tabel 4. Konfigurasi *Hyperparameter* Terbaik dari GridSearchCV

Hyperparameter	Nilai Terpilih
C	0,1
kernel	rbf
gamma	scale
class_weight	balanced

Sumber: (Hasil Penelitian, 2026)

Konfigurasi terbaik menggunakan *class_weight = balanced*, yang meningkatkan sensitivitas model terhadap kelas minoritas (*tweet* banjir *real-time*). Hal ini sejalan dengan karakteristik dataset penelitian yang memiliki rasio ketidakseimbangan kelas sekitar 3,3:1. Nilai C sebesar 0,1 menunjukkan regularisasi yang lebih kuat sehingga menghasilkan margin yang lebih lebar dengan toleransi yang lebih tinggi terhadap kesalahan klasifikasi. Selain itu, penggunaan *kernel* RBF memungkinkan SVM memodelkan batas keputusan nonlinier, sehingga konfigurasi ini lebih sesuai untuk representasi fitur TF-IDF yang *sparse* dan berdimensi tinggi pada penelitian ini.

Perbandingan Model Keseluruhan

Perbandingan performa seluruh model, mulai dari *baseline* hingga SVM hasil tuning *hyperparameter*, disajikan pada Tabel 5.

Tabel 5. Perbandingan Performa Model Keseluruhan

Metrik	SVM	*LR*	*RF*	*SVM Tuned*
CV *Recall* (mean ± SD)	0,8583 ± 0,0256	0,8438 ± 0,0368	0,6125 ± 0,0599	0,8907
Test *Recall*	0,8520	0,8265	0,5918	0,8673
Precision	0,5219	0,5243	0,6237	0,4533
F1-score	0,6473	0,6416	0,6073	0,5954
ROC-AUC	0,8816	0,8741	0,8668	0,8634
PR-AUC	0,6019	0,6261	0,6041	0,5900

Sumber: (Hasil Penelitian, 2026)

Dibandingkan SVM *baseline*, proses *hyperparameter tuning* meningkatkan nilai *CV recall* dari 0,8583 menjadi 0,8907 (peningkatan sebesar 3,24 poin persentase) dan *test recall* dari 0,8520 menjadi 0,8673 (peningkatan sebesar 1,53 poin persentase). Peningkatan tersebut diperoleh karena GridSearchCV mengoptimalkan kombinasi

VOL. 21. NO. 1 AGUSTUS 2026
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Universitas Nusa Mandiri

hyperparameter berdasarkan nilai *mean recall* selama *10-fold GroupKFold cross-validation*. Sebagai konsekuensinya, nilai *precision*, *F1-score*, *ROC-AUC*, dan *PR-AUC* sedikit menurun, yang menunjukkan adanya *trade-off* antara peningkatan sensitivitas model dan ketepatan prediksi positif. Namun demikian, karena tujuan utama penelitian ini adalah meminimalkan *false negative* pada sistem deteksi banjir, *recall* tetap digunakan sebagai dasar utama dalam pemilihan model terbaik. Nilai *test recall* sebesar 0,8673 menunjukkan bahwa model mampu mendeteksi 86,73% kejadian banjir *real-time* pada data pengujian dan melampaui ambang batas minimum *recall* sebesar 0,80 yang telah ditetapkan.

Hasil penelitian ini menunjukkan bahwa SVM hasil *hyperparameter tuning* memberikan performa terbaik berdasarkan metrik *recall* yang menjadi fokus utama evaluasi, sedangkan pada kelompok model *baseline*, Random Forest memperoleh nilai *recall* terendah. Temuan tersebut sejalan dengan penelitian Ristiana & Mutmainah (2026), yang juga melaporkan bahwa Random Forest menghasilkan performa lebih rendah dibandingkan Logistic Regression dan Linear SVM pada klasifikasi *tweet* berbasis TF-IDF. Hasil ini menunjukkan bahwa, pada dataset dan skenario klasifikasi dalam penelitian ini, pendekatan berbasis SVM dan Logistic Regression memberikan performa yang lebih baik dibandingkan Random Forest.

Implementasi Sistem dan Performa *Pipeline*

Pipeline klasifikasi diimplementasikan menggunakan bahasa pemrograman Python dan dijalankan secara otomatis melalui mekanisme *cron job* yang dijadwalkan setiap jam. Pada setiap siklus eksekusi, sistem melakukan pengumpulan *tweet* terbaru, deduplikasi, *preprocessing*, klasifikasi menggunakan model SVM terbaik, ekstraksi lokasi, dan penyimpanan hasil klasifikasi ke dalam basis data MySQL. Basis data terdiri atas dua tabel, yaitu tabel data historis dan tabel monitoring *real-time*, yang keduanya menggunakan skema kolom identik sebagaimana ditunjukkan pada Tabel 6.

Tabel 6. Skema Kolom Tabel Database

Kolom	Tipe Data	Keterangan
id_laporan	INT	Primary Key, Auto Increment. ID unik untuk setiap record.
full_text	TEXT	Teks asli *tweet*.
created_at	DATETIME	Timestamp waktu *tweet* dibuat di platform Twitter/X.
location	VARCHAR(150)	Informasi lokasi yang berhasil diekstraksi dari teks *tweet*.
latitude	DECIMAL(9,6)	Koordinat lintang lokasi kejadian dalam format desimal.
longitude	DECIMAL(9,6)	Koordinat bujur lokasi kejadian dalam format desimal.

Sumber: (Hasil Penelitian, 2026)

Tabel historis digunakan sebagai dasar visualisasi tren temporal pada *dashboard*, sedangkan tabel monitoring menyimpan data hasil *crawling* periodik untuk pemantauan kondisi terkini. Pemisahan kedua tabel ini memungkinkan sistem membedakan antara data akumulatif jangka panjang dan data operasional *real-time* tanpa mengorbankan konsistensi struktur penyimpanan. Ringkasan waktu eksekusi setiap tahap pada proses pengembangan model disajikan pada Tabel 7.

Tabel 7. Ringkasan Waktu Eksekusi Proses Pengembangan Model

No.	Deskripsi	Waktu Eksekusi
1	*Preprocessing* Teks	5 mnt 53,34 dtk
2	Pembagian Data	0,03 dtk
3	Ekstraksi Fitur TF-IDF	0,17 dtk
4	Pelatihan & Evaluasi SVM *Baseline*	4,21 dtk
5	Tuning *Hyperparameter* (GridSearchCV)	19,58 dtk
Total		6 mnt 17,33 dtk

Sumber: (Hasil Penelitian, 2026)

Berdasarkan Tabel 7, tahap *preprocessing* mendominasi waktu eksekusi karena mencakup seluruh proses pembersihan dan normalisasi teks, sedangkan tahap berikutnya bekerja pada representasi numerik sehingga membutuhkan waktu yang jauh lebih singkat. Waktu eksekusi pada Tabel 7 merupakan hasil pengukuran selama proses pengembangan model menggunakan keseluruhan dataset penelitian sebanyak 4.436 *tweet*, sehingga tidak merepresentasikan waktu eksekusi pipeline operasional pada setiap siklus *cron job*. Setelah model diperoleh, sistem operasional hanya menjalankan proses *crawling*, *preprocessing*, transformasi TF-IDF menggunakan *vectorizer* yang telah dilatih, klasifikasi menggunakan model SVM terbaik, ekstraksi lokasi, dan penyimpanan hasil ke basis data.

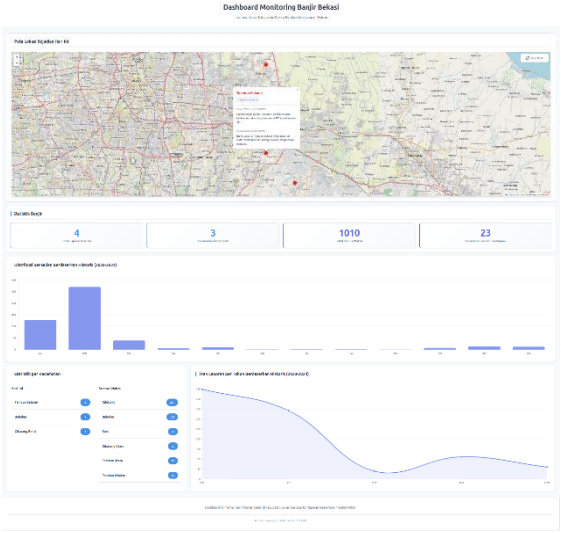
Total waktu pengembangan model sebesar 6 menit 17,33 detik menunjukkan bahwa proses pembangunan model dapat diselesaikan secara efisien. Temuan ini sejalan dengan (Purwandari dkk., 2023), yang menerapkan mekanisme penjadwalan serupa untuk proses *crawling* data cuaca secara otomatis setiap jam. Dengan demikian, meskipun waktu pada Tabel 7 merupakan waktu pengembangan model, sistem operasional yang dijalankan melalui *cron job* setiap jam tetap memiliki margin waktu yang memadai untuk

VOL. 21. NO. 1 AGUSTUS 2026
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Universitas Nusa Mandiri
INTI NUSA MANDIRI
DOI: https://doi.org/10.33480/inti.v21i1.8642

mendukung pemantauan kondisi banjir secara mendekati *real-time*.

Implementasi *Dashboard* Pemantauan Berbasis Web

Dashboard pemantauan banjir berhasil diimplementasikan sebagai aplikasi berbasis web menggunakan PHP sebagai *backend*, HTML5, CSS3, dan JavaScript sebagai *frontend* dengan MySQL sebagai sistem manajemen basis data. *Dashboard* memanfaatkan hasil klasifikasi yang dihasilkan oleh *pipeline* untuk menyajikan informasi deteksi banjir secara spasial dan temporal kepada pengguna secara *real-time*. Gambar 2 menunjukkan tampilan *dashboard* yang dikembangkan.



Sumber: (Hasil Penelitian, 2026)

Gambar 2. Tampilan *Dashboard* Website Pemantauan Banjir

Berdasarkan implementasi tersebut, *dashboard* berhasil menyediakan sejumlah fitur yang mendukung pemantauan banjir secara *real-time*. Lokasi laporan diperoleh melalui ekstraksi nama kecamatan dari teks *tweet* yang kemudian dipetakan ke koordinat geografis sehingga dapat divisualisasikan sebagai penanda pada peta interaktif. *Dashboard* juga menampilkan empat kartu ringkasan statistik (Total Laporan Hari Ini, Kecamatan Terdampak Hari Ini, Total Keseluruhan, dan Kecamatan Pernah Terdampak), grafik batang tren bulanan berdasarkan data historis periode 2020–2024, grafik garis tren tahunan, serta statistik per kecamatan yang dapat digunakan untuk menavigasi lokasi pada peta. Untuk laporan yang berada pada koordinat yang sama, sistem mengelompokkannya dalam satu penanda dengan *popup* yang menampilkan maksimal lima laporan terbaru beserta nama kecamatan, timestamp, dan isi *tweet*.

Integrasi antara model klasifikasi SVM dengan *dashboard* pemantauan berbasis web memungkinkan hasil deteksi banjir dari data Twitter dapat diakses dan diinterpretasikan secara langsung oleh pengguna tanpa memerlukan pemahaman teknis terhadap proses *machine learning*. Pendekatan ini sejalan dengan penelitian Marzouk dkk. (2022) yang mengintegrasikan model klasifikasi *machine learning* dengan sistem berbasis web untuk mendukung pengambilan keputusan melalui visualisasi data secara interaktif. Penyajian informasi dalam bentuk peta interaktif, statistik ringkasan, dan visualisasi tren pada penelitian ini relevan untuk konteks pemantauan bencana karena membantu pengguna memperoleh gambaran kondisi banjir secara cepat sehingga mendukung respons terhadap kejadian banjir yang terdeteksi.

Evaluasi Pengujian *Black box*

Pengujian *black box* dilakukan untuk memverifikasi performa fungsional *dashboard* dari perspektif pengguna. Pengujian mencakup seluruh fitur utama sistem, sedangkan hasil pengujian disajikan pada Tabel 8.

Tabel 8. Hasil Pengujian *Black box*

No.	Skenario Uji	Hasil
1	Akses halaman *dashboard* (pemuatan komponen)	Berhasil
2	Peta interaktif dengan penanda lokasi laporan banjir	Berhasil
3	Interaksi *popup* pada penanda (detail lokasi, waktu, dan teks *tweet*)	Berhasil
4	Pengelompokan beberapa laporan pada penanda lokasi yang sama	Berhasil
5	Tombol reset posisi peta ke tampilan awal	Berhasil
6	Kartu ringkasan statistik (Total Laporan Hari Ini, Kecamatan Terdampak Hari Ini, Total Keseluruhan, dan Kecamatan Pernah Terdampak)	Berhasil
7	Visualisasi tren historis (grafik batang bulanan dan grafik garis tahunan)	Berhasil
8	Statistik per kecamatan (hari ini dan kumulatif semua waktu)	Berhasil
9	Navigasi peta otomatis melalui klik nama kecamatan pada tabel statistik harian	Berhasil

Sumber: (Hasil Penelitian, 2026)

Berdasarkan Tabel 8, seluruh skenario uji berhasil dijalankan sehingga tingkat keberhasilan sistem mencapai 100% (9 dari 9 skenario uji), melampaui ambang batas minimum sebesar 95% yang telah ditetapkan. Hasil ini menunjukkan bahwa seluruh fungsi utama *dashboard* telah

VOL. 21. NO. 1 AGUSTUS 2026
P-ISSN: 0216-6933 | E-ISSN: 2685-807X
Diterbitkan Oleh: LPPM Universitas Nusa Mandiri

beroperasi sesuai spesifikasi tanpa ditemukan kegagalan fungsional.

Pendekatan pengujian fungsional ini sejalan dengan penelitian Mahadika dkk. (2023), yang juga menerapkan *black box testing* terhadap setiap fungsi pada sistem informasi berbasis web sebagai tahap validasi awal sebelum pengujian oleh pengguna dan ahli sistem informasi. Dalam sistem pemantauan banjir, keberhasilan fungsi *dashboard* penting untuk menjamin keandalan penyajian informasi, seperti akurasi penanda lokasi, pembaruan statistik, dan navigasi peta. Dengan tingkat keberhasilan 100%, *dashboard* dinilai memenuhi kebutuhan fungsional sesuai rancangan sistem sehingga layak dilanjutkan ke tahap evaluasi pengalaman pengguna.

Evaluasi Pengalaman Pengguna (UEQ-S)

Evaluasi pengalaman pengguna dilakukan menggunakan User Experience Questionnaire Short (UEQ-S) terhadap 18 responden yang berdomisili di Kabupaten Bekasi setelah mencoba *dashboard* secara langsung. Responden dipilih karena merupakan target pengguna utama sistem sehingga penilaiannya merepresentasikan konteks penggunaan yang sebenarnya. Instrumen UEQ-S mengevaluasi pengalaman pengguna melalui delapan pasangan kata sifat yang dikelompokkan ke dalam dua dimensi, yaitu Pragmatic Quality dan Hedonic Quality. Seluruh data dinyatakan valid dan tidak ditemukan respons yang mencurigakan sehingga seluruhnya digunakan dalam analisis.

Tabel 9. Hasil Evaluasi UEQ-S per Item

Dimensi	Item	Mean	SD
Pragmatic Quality	Menghambat–Mendukung	2,28	0,89
	Rumit–Mudah	2,11	1,13
	Tidak Efisien–Efisien	1,94	1,11
	Membingungkan–Jelas	1,72	1,32
Hedonic Quality	Membosankan–Menarik	1,78	1,26
	Tidak Menarik–Menarik	2,11	1,08
	Konvensional–Inovatif	1,94	1,21
	Biasa–Terdepan	1,89	1,08

Sumber: (Hasil Penelitian, 2026)

Tabel 10. Ringkasan Evaluasi UEQ-S per Skala

Skala	Mean	SD	Cronbach's Alpha	*Benchmark*
Pragmatic Quality	2,01	1,02	0,935	Sangat Baik
Hedonic Quality	1,93	1,08	0,951	Sangat Baik
Keseluruhan	1,97	1,02	-	Sangat Baik

Sumber: (Hasil Penelitian, 2026)

Nilai Cronbach's Alpha sebesar 0,935 pada Pragmatic Quality dan 0,951 pada Hedonic Quality menunjukkan bahwa kedua skala memiliki reliabilitas yang sangat tinggi ($\alpha > 0,7$). Item Menghambat–Mendukung memperoleh skor tertinggi (2,28), yang mengindikasikan bahwa sistem dinilai sangat mendukung kegiatan pemantauan, sedangkan item Membingungkan–Jelas memperoleh skor terendah (1,72), meskipun masih berada pada kategori positif, sehingga kejelasan antarmuka masih memiliki ruang untuk ditingkatkan.

Secara keseluruhan, hasil evaluasi menunjukkan bahwa *dashboard* memperoleh skor rata-rata UEQ-S sebesar 1,97, dengan skor Pragmatic Quality sebesar 2,01 dan Hedonic Quality sebesar 1,93. Ketiga skala berada pada kategori Sangat Baik, yang menunjukkan bahwa *dashboard* mampu memberikan pengalaman pengguna yang positif, baik dari aspek fungsional maupun aspek nonfungsional. Skor Pragmatic Quality mengindikasikan bahwa sistem dipersepsikan mudah digunakan dan mendukung aktivitas pemantauan banjir secara efektif, sedangkan skor Hedonic Quality menunjukkan bahwa *dashboard* dinilai menarik dan memberikan pengalaman penggunaan yang positif. Temuan tersebut didukung oleh hasil evaluasi per item, di mana aspek Menghambat–Mendukung memperoleh skor tertinggi, sementara aspek Membingungkan–Jelas memperoleh skor terendah, sehingga kejelasan antarmuka masih memiliki ruang untuk ditingkatkan pada pengembangan sistem selanjutnya.

KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem deteksi banjir berbasis data Twitter untuk Kabupaten Bekasi menggunakan Support Vector Machine (SVM). Proses pengumpulan dan *preprocessing* menghasilkan dataset yang layak untuk membangun model klasifikasi. Hasil evaluasi komparatif menunjukkan bahwa SVM memberikan performa terbaik dibandingkan Logistic Regression (LR) dan Random Forest (RF). Setelah dilakukan *hyperparameter tuning* menggunakan GridSearchCV, model terbaik mencapai *test recall* sebesar 0,8673 sehingga melampaui target *recall* minimum sebesar 0,80. Model tersebut selanjutnya berhasil diintegrasikan ke dalam *dashboard* pemantauan banjir berbasis web melalui mekanisme *crawling* otomatis. *Dashboard* memenuhi seluruh skenario pengujian *black box* dengan tingkat keberhasilan 100% dan

memperoleh skor UEQ-S sebesar 1,97 (Sangat Baik), yang menunjukkan bahwa sistem memiliki performa fungsional dan pengalaman pengguna yang baik.

Penelitian ini masih terbatas pada perbandingan algoritma *machine learning* klasik, yaitu Support Vector Machine (SVM), Logistic Regression (LR), dan Random Forest (RF). Oleh karena itu, penelitian selanjutnya dapat memperluas kajian dengan membandingkan algoritma *machine learning* klasik tersebut dengan pendekatan berbasis *deep learning* maupun *transformer*, seperti LSTM, BERT, IndoBERT, atau model berbasis *embedding*, untuk memperoleh evaluasi yang lebih komprehensif. Selain itu, penelitian ini masih terbatas pada satu wilayah studi, yaitu Kabupaten Bekasi. Oleh karena itu, penelitian selanjutnya diharapkan dapat melakukan penelitian serupa pada wilayah lain untuk mengevaluasi kemampuan generalisasi pendekatan yang digunakan terhadap karakteristik data yang berbeda. Penelitian ini juga belum mengevaluasi signifikansi statistik perbedaan performa antaralgoritma, sehingga analisis menggunakan uji statistik inferensial dapat dipertimbangkan pada penelitian selanjutnya untuk melengkapi evaluasi performa model. Di samping itu, disarankan mengembangkan mekanisme ekstraksi lokasi yang lebih fleksibel, misalnya melalui pendekatan *Named Entity Recognition* (NER), agar mampu mengenali beberapa lokasi dalam satu *tweet* maupun lokasi yang hanya disebutkan pada tingkat kabupaten. Kejelasan antarmuka *dashboard* juga dapat ditingkatkan melalui penambahan panduan penggunaan atau *tooltip* informatif, serta mempertimbangkan integrasi sumber data alternatif, seperti platform media sosial lain atau sumber berita daring, guna mengurangi ketergantungan pada satu platform yang kebijakan akses datanya dapat berubah.

REFERENSI

Adili, P., & Chen, Y. (2024). Fast disaster event detection from social media: an active learning method. *International Journal of Computers Communications & Control*, *19*(2).

Agustina, I. H. (2021). Study of flood-prone areas in Bekasi Regency. *Jurnal Geografi Gea*, *21*(2), 180–187.

Alharbi, A. A. (2024). Classification Performance Analysis of Decision Tree-Based Algorithms with Noisy Class Variable. *Discrete Dynamics in Nature and Society*, *2024*(1), 6671395.

Aryanti, N. N. A., & Suria, O. (2025). Analisis Sentimen Terhadap Pemutusan Hubungan Kerja di Indonesia: Komparasi IndoBERT dengan SVM, Random Forest, dan Decision Tree dengan Optimasi TF-IDF. *Rabit: Jurnal Teknologi dan Sistem Informasi Univrab*, *10*(2), 1158–1176.

Azizah, M., Subiyanto, A., Triutomo, S., & Wahyuni, D. (2022). Pengaruh perubahan iklim terhadap bencana hidrometeorologi di kecamatan cisarua-kabupaten bogor. *PENDIPA Journal of Science Education*, *6*(2), 541–546.

BMKG. (2021). *Peta Rata-Rata Curah Hujan dan Hari Hujan Periode 1991–2020 Indonesia*. Pusat Informasi Perubahan Iklim BMKG.

BNPB. (2025). *Data Bencana Indonesia 2024* (Vol. 3). Pusat Data Informasi dan Komunikasi Kebencanaan, BNPB.

Caniago, R., & Habibi, M. (2023). Klasifikasi Situasi Bencana Alam Banjir Menggunakan Support Vector Machine Berdasarkan Data Twitter. *INFORMATIKA*, *15*(1), 16–22.

Esparza, M., Farahmand, H., Liu, X., & Mostafavi, A. (2024). Enhancing inundation monitoring of road networks using crowdsourced flood reports. *Urban Informatics*, *3*(1), 25.

Ghosh, D., & Cabrera, J. (2021). Enriched random forest for high dimensional genomic data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *19*(5), 2817–2828.

Hassan, S. U., Ahamed, J., & Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification. *Sustainable Operations and Computers*, *3*, 238–248.

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, *4*(9).

Mahadika, D. A., Aristyagama, Y. H., & Budiyanto, C. W. (2023). Evaluation of Website Based Information System To Monitor Student Learning Progress In Schools Using ISO/IEC 9126 Standards And GTMetrix. *IJIE (Indonesian Journal of Informatics Education)*, *7*(1), 42–51.

Mansoor, M., Rehman, Z. U., Afzal, S., Gil, J. M., Jo, J., & Yoon, Y. C. (2025). NLP Framework for Disaster Prediction Using Social Media Analytics and Large Language Models. *International Journal on Advanced Science, Engineering & Information Technology*, *15*(4).

Marzouk, R., Alluhaidan, A. S., & El_Rahman, S. A. (2022). An analytical predictive models and secure web-based personalized diabetes monitoring system. *IEEE Access*, *10*, 105657–105673.

Moghadas, M., Fekete, A., Rajabifard, A., & Kötter, T. (2023). The wisdom of crowds for improved

disaster resilience: a near-real-time analysis of crowdsourced social media data on the 2021 flood in Germany. *GeoJournal*, *88*(4), 4215–4241.

Palomino, M. A., & Aider, F. (2022). Evaluating the effectiveness of text pre-processing in sentiment analysis. *Applied Sciences*, *12*(17), 8765.

Prosise, J. (2022). *Applied Machine Learning and AI for Engineers: Solve Business Problems that Can't be Solved Algorithmically*. O'Reilly Media, Inc.

Purwandari, K., Perdana, R. B., Sigalingging, J. W., Rahutomo, R., & Pardamean, B. (2023). Automatic smart crawling on twitter for weather information in indonesia. *Procedia Computer Science*, *227*, 795–804.

Reinert, J., Dittmer, C., Lorenz, D. F., & Klopries, E. M. (2025). Design Flaws at the Interface of Flood Forecasting, Early Warning and Disaster Response in the Disaster in Western Germany in July 2021—An Interdisciplinary Analysis. *Journal of Flood Risk Management*, *18*(3), e70099.

Ristiana, I., & Mutmainah. (2026). Analisis Sentimen Bencana Banjir Sumatera Menggunakan TF-IDF Dan Logistic Regression. *Jurnal Sistem Informasi dan Teknologi (SINTEK)*, *6*(1), 57–65.

Sari, P. K., & Suryono, R. R. (2024). Komparasi algoritma Support Vector Machine dan Random Forest untuk analisis sentimen metaverse. *Jurnal Mnemonic*, *7*(1), 31–39.

Segovia-Cardozo, D. A., Bernal-Basurco, C., & Rodríguez-Sinobas, L. (2023). Tipping bucket rain gauges in hydrological research: Summary on measurement uncertainties, calibration, and error reduction strategies. *Sensors*, *23*(12), 5385.

Sinaga, A., & Nainggolan, S. P. (2023). Analisis perbandingan akurasi dan waktu proses algoritma stemming Arifin-Setiono dan Nazief-Adriani pada dokumen teks Bahasa Indonesia. *Sebatik*, *27*(1), 63–69.

Sufi, F. K., & Khalil, I. (2022). Automated disaster monitoring from social media posts using AI-based location intelligence and sentiment analysis. *IEEE Transactions on Computational Social Systems*, *11*(4), 4614–4624.

Thieken, A. H., Bubeck, P., Heidenreich, A., Von Keyserlingk, J., Dillenardt, L., & Otto, A. (2023). Performance of the flood warning system in Germany in July 2021–insights from affected residents. *Natural Hazards and Earth System Sciences*, *23*(2), 973–990.

Thölke, P., Mantilla-Ramos, Y. J., Abdelhedi, H., Maschke, C., Dehgan, A., Harel, Y., & Jerbi, K. (2023). Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data. *NeuroImage*, *277*, 120253.

Valkenborg, D., Rousseau, A. J., Geubbelmans, M., & Burzykowski, T. (2023). Support vector machines. *American Journal of Orthodontics and Dentofacial Orthopedics*, *164*(5), 754–757.

Veigel, N., Kreibich, H., De Bruijn, J. A., Aerts, J. C., & Cominola, A. (2025). Content analysis of multi-annual time series of flood-related Twitter (X) data. *Natural Hazards and Earth System Sciences*, *25*(2), 879–891.

Yang, T., Xie, J., Li, G., Zhang, L., Mou, N., Wang, H., & Wang, X. (2022). Extracting disaster-related location information through social media to assist remote sensing for disaster analysis: The case of the flood disaster in the Yangtze River Basin in China in 2020. *Remote Sensing*, *14*(5), 1199.

Yu, C., & Wang, Z. (2024). Multimodal social sensing for the spatio-temporal evolution and assessment of nature disasters. *Sensors*, *24*(18), 5889.