Intechno Journal :
Information Technology Journal
Vol. 8 No. 1 July 2026
Homepage: https://jurnal.amikom.ac.id/index.php/intechno
INTECHNO Journal
INFORMATION TECHNOLOGY JOURNAL
e-ISSN: 2655-1438
p-ISSN: 2655-1632

.

A Hybrid Rule-Based and Multinomial Naïve Bayes System for Sentiment and Intent Classification of Indonesian Public Reports with Sarcasm Detection

Suhendri[1], Sahal Ubaidillah Gunardo[2], Kartika Dwi Mulyana[3], Ruli Susanti[4], Dika Alfaizal Akbar[5], Amelia Putri[6]

[1,2,3,4,5,6]Department of Informatics, Universitas Majalengka, Jl. KH. Abdul Halim No. 103, Majalengka 45418, Indonesia

Article Info

Article history:

Received: 1 June 2026
Revised: 17 June 2026
Accepted: 2 July 2026
Published: 31 July 2026

Keywords:

Citizen Report Classification
Indonesian NLP
Multinomial Naive Bayes
Sentiment Analysis
TF-IDF

ABSTRACT

Public service reporting systems in Indonesia face significant challenges in processing large volumes of unstructured citizen feedback efficiently. This study proposes Aspiralytica, a mobile-based citizen report classification system that integrates TF-IDF feature extraction with a Multinomial Naive Bayes (MNB) classifier within a hybrid rule-based and machine learning architecture. The system simultaneously performs three-class sentiment classification (positive, negative, neutral) and five-class intent classification (complaint, appreciation, request, emergency, suggestion), with automated priority level determination and a rule-based sarcasm detection module achieving F1 of 89.80%. Evaluated on an augmented dataset of 960 sentiment-labeled and 1,800 intent-labeled Indonesian-language citizen report texts — comprising simulated reports and real-world data collected from LAPOR! (lapor.go.id, accessed June 2026) — using Stratified 10-Fold Cross-Validation, the proposed MNB model achieved sentiment classification accuracy of 95.41% (F1: 95.39%) and intent classification accuracy of 95.89% (F1: 95.89%). An ablation study confirmed TF-IDF with MNB as the dominant performance driver, and a computational efficiency benchmark empirically justified MNB selection with mean inference latency of 0.3955 ms and throughput of 52,312 requests per second. The system is deployed as a FastAPI backend integrated with a React Native mobile frontend, delivering real-time classification through a citizen-facing interface.

Corresponding Author:

Suhendri
Email: theprof.suhendri@yahoo.co.id

1. INTRODUCTION

The rapid expansion of digital public services has substantially increased the volume of citizen-generated reports submitted to government agencies and public service institutions. In Indonesia, the adoption of e-government platforms has been accompanied by considerable growth in textual citizen feedback received through mobile applications, web portals, and dedicated reporting channels [1]. Despite this progress, the processing of incoming public reports remains predominantly manual in most institutions: officers read, categorize, and assign priority to each submission individually. This approach is not only time-consuming and resource-intensive but also introduces inconsistency in classification, particularly as report volumes scale. A

systematic review of e-government implementation in Indonesian local governments identified institutional barriers — including limited digital infrastructure and bureaucratic resistance — that impede the automation of citizen feedback processing [2].

Prior studies on Indonesian-language NLP have predominantly applied sentiment classifiers to social media and e-commerce review data [3], [4], [5], which differ from civic reporting text in vocabulary, formality, and domain specificity. Transformer-based models such as IndoBERT have demonstrated strong sentiment F1 on public service review corpora [6], yet their inference latency of 50–200 ms per request limits deployment feasibility in synchronous mobile environments. Within the citizen complaint domain, Madyatmadja et al. [7] proposed a rule-based contextual framework for Indonesian civic text classification, while Intani et al. [8] automated complaint validity detection on Jakarta's JakLapor platform using SVM; neither system performs simultaneous multi-class intent classification nor incorporates sarcasm handling. More recent work by Alpiana et al. [9] combined IndoBERT with XGBoost for complaint subcategory prediction at Universitas Negeri Surabaya (95% accuracy), confirming the viability of hybrid architectures for Indonesian civic NLP but again without addressing the latency-accuracy trade-off in mobile deployment. On sarcasm detection, Suhartono et al. [10] introduced IdSarcasm, a benchmark evaluating language models on Indonesian sarcasm across Reddit and Twitter, finding substantial performance variation across domains — underscoring that sarcasm in civic text remains underaddressed. These gaps collectively motivate the present work: a lightweight hybrid TF-IDF + MNB system performing simultaneous three-class sentiment and five-class intent classification with an integrated sarcasm gate, optimized for real-time mobile deployment.

To address these gaps, this study proposes Term Frequency–Inverse Document Frequency (TF-IDF) combined with Multinomial Naive Bayes (MNB) for both sentiment and intent classification of Indonesian-language public reports. While transformer-based models such as IndoBERT offer contextual advantages, their substantially higher inference latency — estimated at 50–200 ms per request compared to the 0.3955 ms achieved by the proposed MNB pipeline in empirical benchmarking — makes them impractical for synchronous REST API integration in resource-limited environments. The primary contributions of this research are: (1) a hybrid rule-based and ML classification system performing simultaneous sentiment and intent classification with automated priority determination; (2) a multi-layer sarcasm detection module independently evaluated and achieving sarcasm-class F1 of 89.80%; (3) an ablation study quantifying the individual contribution of each architectural component; and (4) an empirical computational efficiency benchmark. The MNB model achieved sentiment accuracy of 95.41% (F1: 95.39%) and intent accuracy of 95.89% (F1: 95.89%) under Stratified 10-Fold Cross-Validation on 960 and 1,800 augmented Indonesian-language citizen report texts respectively.

2. METHOD

2.1 Research Design Overview

This study follows a quantitative design based on text classification system development. The research pipeline consists of five stages: (1) dataset construction and annotation, (2) text preprocessing, (3) feature extraction, (4) model training and evaluation, and (5) system integration into a mobile application. Figure 1 illustrates the complete pipeline from raw citizen report text to the final classification output.

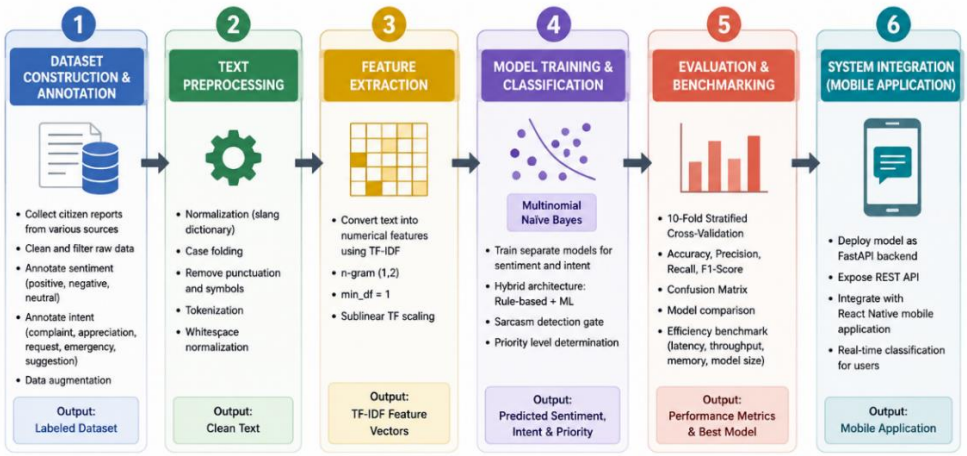

Figure 1. Research Method Flowchart

2.2 Dataset Construction and Annotation

The initial dataset comprised 405 manually annotated sentiment samples and 847 intent samples from simulated citizen report texts, supplemented by 49 real-world citizen reports collected from the LAPOR! platform (lapor.go.id, accessed June 2026) through manual scraping and independently labelled by an annotator — comprising 42 negative and 7 neutral sentiment labels, and 32 complaint, 12 request, 3 emergency, and 2 suggestion intent labels. No publicly available benchmark dataset was adopted because, to the best of the authors' knowledge, no existing public Indonesian-language corpus simultaneously provides civic-domain citizen report texts with both sentiment and intent labels across the specific class taxonomy used in this study (three-class sentiment, five-class intent including emergency). The use of a purpose-built annotated corpus is therefore necessitated by the novelty of the task definition; cross-dataset comparison is consequently performed at the level of baseline model architecture (SVM, Decision Tree, Random Forest, Logistic Regression) evaluated under identical experimental conditions rather than across corpora. The corpus was expanded through three complementary augmentation techniques [11], [12] to produce 960 sentiment samples (320 per class) and 1,800 intent samples (360 per class). Table 1 summarizes the augmentation techniques, contributions, and quality control procedures applied.

Annotation was conducted by two independent annotators following written guidelines specifying operational definitions and boundary criteria for each class label, with particular attention to the keluhan–saran and negatif–netral boundaries. Annotation was performed in two passes separated by a minimum 48-hour interval; a 10% random sample per class was re-examined for boundary-case consistency.

Annotation reliability was assessed through inter-annotator agreement (IAA) on a stratified 20% subset of the corpus (n = 81 sentiment samples, n = 90 intent samples), independently labeled by a second annotator following the same written guidelines. Cohen's Kappa coefficients of $\kappa = 0.68$ for sentiment and $\kappa = 0.73$ for intent indicate substantial agreement (Landis and Koch, 1977), confirming the reliability of the annotation scheme. Disagreements were resolved through adjudication by the lead annotator.

Table 1. Data augmentation techniques — methods, sample contributions, and quality control

Technique	Samples added	Method	Diversity	Quality control
Synonym replacement	Sent: +280 Int: +560	Indonesian WordNet + domain lexicon	High	Preserves polarity; excludes negation markers
Controlled paraphrasing	Sent: +230 Int: +460	Active-to-passive, clause reorder	Medium	Constrained to prevent label-boundary drift
Back-translation	Sent: +222 Int: +340	ID→EN→ID neural MT pipeline	Medium	Semantic drift filter applied post-generation
Total augmented	Sent: +732 Int: +1,360	—	—	15% stratified human review per class

Note: Augmented samples derived exclusively from the training partition of each CV fold to prevent data leakage: augmentation was applied within each fold's training split after the train/test partition was established, ensuring no augmented data overlapped with evaluation folds. 15% stratified sample per class reviewed manually by the lead annotator.

2.3 Text Preprocessing and Sarcasm Detection

A multi-stage preprocessing pipeline was implemented: (1) text normalization using a domain-specific slang dictionary of over 200 word-mapping entries applied via regex-based pattern matching; (2) case folding to lowercase; (3) punctuation and symbol removal, retaining only alphabetic characters; and (4) whitespace normalization. Prior to ML classification, a rule-based sarcasm detection module serves as a pre-classification gate by identifying co-occurrence patterns between positive evaluative phrases (e.g., "bagus banget", "luar biasa") and contrastive markers (e.g., "padahal", "tapi nyatanya"), combined with negation-aware keyword checking within a two-token window. When sarcasm is detected, the system directly assigns negative sentiment, complaint intent, and high priority. Table 2 presents a sample of the normalization dictionary entries most impactful for classification accuracy.

Table 2. Sample slang normalization dictionary- entries most impactful for sentiment and intent classification

Informal token	Normalized form	Role in classification
gak / ga	tidak (not)	Negation marker — highest frequency normalization
benerin	memperbaiki (to fix)	Informal verb → formal infinitive

jln	jalan (road/street)	Abbreviated noun — civic infrastructure
minta	meminta (to request)	Colloquial verb → standard form
bgt / banget	sangat (very)	Intensifier — critical for sentiment polarity
blm	belum (not yet)	Temporal negation — affects complaint detection
tolong	mohon (please)	Informal request marker → formal polite form
krn	karena (because)	Causal connector — abbreviated
udah / udh	sudah (already)	Aspect marker — informal past
hrs	harus (must)	Modal verb — abbreviated

Prior to ML classification, a rule-based sarcasm detection module serves as a pre-classification gate. The heuristic operates across five detection layers as described in Table 3. When sarcasm is detected, the system directly assigns negative sentiment, complaint intent, and high priority, bypassing the ML classifier. The module was independently evaluated on 39 annotated samples achieving sarcasm-class precision of 95.65%, recall of 84.62%, and F1 of 89.80%.

Table 3. Sarcasm detection module — five-layer rule description and trigger examples

Layer	Pattern description	Trigger example	Detection cue
1a	Positive phrase + contrastive marker	"bagus banget, padahal..."	Positive adj. + contrast conj.
1b	Positive phrase + negation window	"mantap sekali, tidak..."	Positive adj. + negation ≤2 tokens
2	Hyperbolic praise + failure term	"luar biasa buruknya"	Superlative + negative noun
3	Appreciative opener + explicit complaint	"terima kasih, tapi pelayanan..."	Gratitude + complaint clause
4	Formal praise + factual contradiction	"pelayanan prima, nyatanya..."	Formal positive + factual inversion
5	Positive emoji + negative context	":) tidak pernah diperbaiki"	Emoji + negation clause

2.4 Hybrid Classification Architecture

The system follows a hybrid architecture combining keyword-based rule matching with a machine learning fallback [13]. Figure 2 illustrates the complete classification flow from preprocessed input to final output. Domain-specific keyword dictionaries were defined for each class label. The rule-based component is checked first; when a keyword match is found, the label is assigned directly. When no rule matches, the text is passed to the ML classifier. A deterministic priority module derives the final priority level: high for emergency intent or complaint with negative sentiment; medium for complaint or request with non-negative sentiment; and low for suggestion, appreciation, or neutral cases.

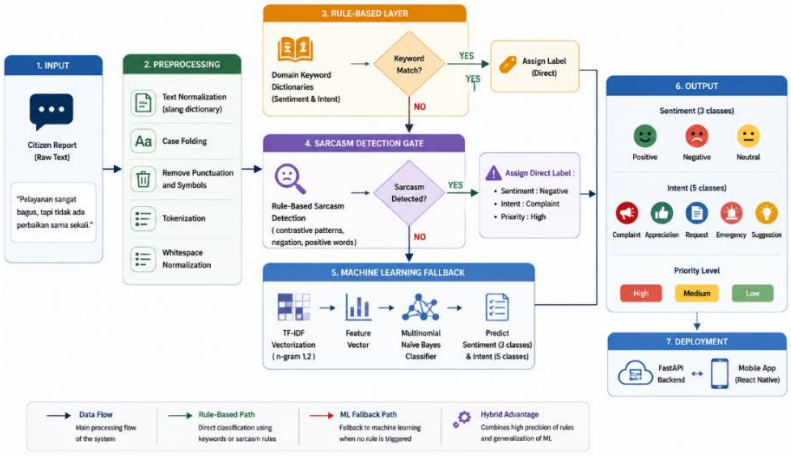

Figure 2. Hybrid Classification Architecture

2.5 Feature Extraction and Classification Model

TF-IDF was used for feature extraction. The weight of term t in document d is computed as:

$$TF\text{-}IDF(t,d) = TF(t,d) \times \log\left(\frac{N}{1+DF(t)}\right) \quad (1)$$

where N is the total number of documents and DF(t) is the document frequency of t. Sublinear TF scaling was applied [14]. The vectorizer was configured with ngram_range = (1,2) to capture unigram and bigram features, and min_df=1 to retain rare domain-specific vocabulary. The MNB classifier computes:

$$P(c \mid d) \propto P(c) \prod_{t \in d} P(t \mid c)^{f_{t,d}} \quad (2)$$

with Laplace smoothing α=0.3 [15]. Two independent MNB instances were trained — one for sentiment, one for intent — each encapsulated in a scikit-learn Pipeline with the TF-IDF vectorizer, serialized to .pkl format for real-time inference.

2.6 Evaluation and System Integration

Both models were evaluated using Stratified 10-Fold Cross-Validation [16], reporting Accuracy, Precision (macro), Recall (macro), and F1-Score (macro) as mean ± standard deviation. A computational efficiency benchmark compared MNB, SVM, Logistic Regression, and Random Forest across 1,000 timed inference iterations with 10 warm-up requests, measuring latency (mean, std, P95, P99), memory allocation via tracemalloc, model file size, and throughput. The trained models were deployed as a RESTful API built with FastAPI (Python), exposing a POST /analyze endpoint. The mobile frontend was developed using React Native with Expo.

Ablation Study Configuration. To quantify the contribution of each architectural component, an ablation study was conducted across four incrementally constructed configurations, all evaluated under identical Stratified 10-Fold Cross-Validation on the full augmented dataset. The configurations are defined as follows: (A) Rule-Based Only — only keyword-matching and pattern rules are applied for sentiment and intent assignment, with no ML classifier; this baseline isolates the discriminative capacity of the rule layer alone. (B) ML Only (TF-IDF + HybridNB) — the TF-IDF vectorizer and MNB classifier pipeline is used without any rule-based override or sarcasm gate; this configuration establishes the statistical model's standalone performance ceiling. (C) ML + Sarcasm Detection — the sarcasm detection module is added on top of configuration B, routing texts flagged as sarcastic to a forced negative-sentiment/complaint-intent assignment before the ML classifier processes the non-sarcastic remainder; this isolates the contribution of sarcasm handling. (D) Full Hybrid System — the complete production architecture, layering the emergency keyword rule, the sarcasm gate, and the ML classifier in sequence; this is the proposed Aspiralytica system as deployed. The purpose of this design is to isolate and attribute performance differences to each individual component, enabling principled interpretation of the observed F1 gains and losses in Table 9.

3. RESULTS AND DISCUSSIONS

3.1 Dataset Distribution

Following augmentation and the addition of the 49 real-world LAPOR! reports, the resulting corpus comprised 959 sentiment-labeled samples (352 negative, 307 neutral, 300 positive) and 1,799 intent-labeled samples (382 complaint, 362 request, 353 emergency, 352 suggestion, 350 appreciation). Because the LAPOR! supplement was skewed toward the negative sentiment and complaint intent classes, the final corpus is close to but not perfectly balanced; MNB class-prior estimation and macro-averaged metrics were computed directly from this realized distribution rather than an assumed uniform prior. The sentiment and intent categories before and after the augmentation process are shown in Figures 3 and 4, respectively.

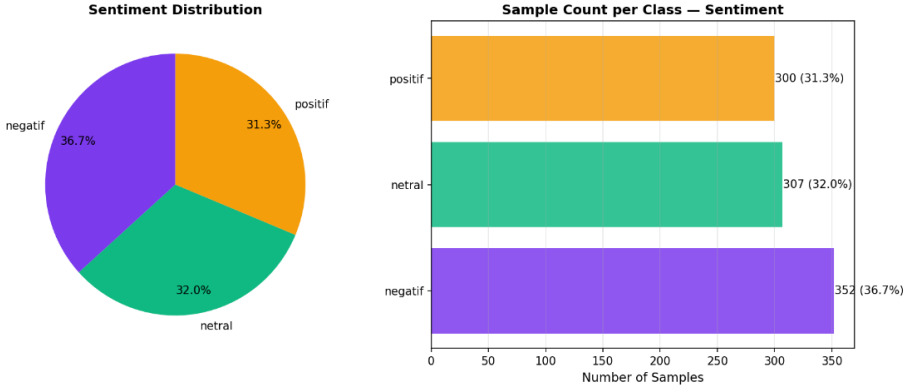

Figure 3. Dataset Distribution — Sentiment Classification

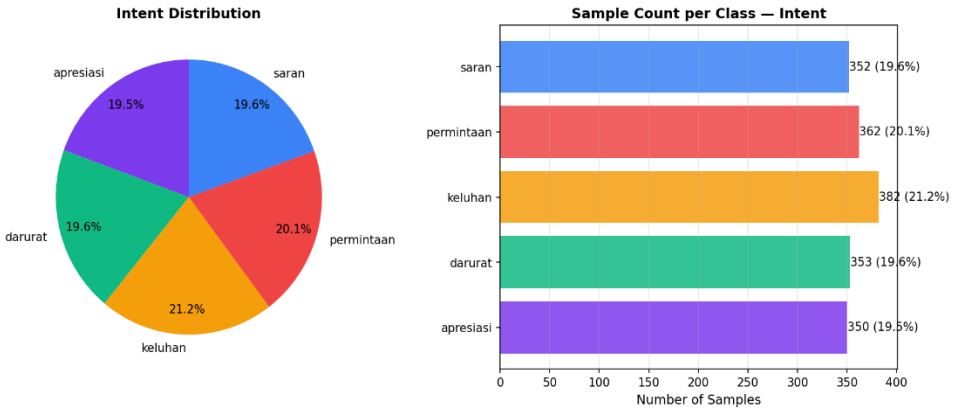

Figure 4. Dataset Distribution — Intent Classification

3.2 Sentiment Classification Performance

The sentiment model achieved 10-fold CV accuracy of 95.41% (±2.34%) and macro F1 of 95.39% (±2.36%), demonstrating consistent generalization across folds. On the 80:20 holdout split (n=192), overall accuracy reached 95.31% with macro precision 95.48%, recall 95.37%, and F1 95.42%, as detailed in Table 4.

Table 4.Sentiment Classification Report (80:20 Holdout, n=192)

Class	Precision	Recall	F1-Score	Support
Negative	93.06%	94.37%	93.71%	71
Neutral	95.08%	95.08%	95.08%	61
Positive	98.31%	96.67%	97.48%	60
Macro Avg	95.48%	95.37%	95.42%	192

The positive class achieved the highest precision (98.31%) and F1 (97.48%), with recall of 96.67%. The negative class exhibits the lowest recall (94.37%) and F1 (93.71%), indicating a small number of negative instances misclassified as neutral or positive – consistent with implicit sentiment expressions where negative content is conveyed through pragmatic implicature rather than explicit lexical markers [17]. This pattern is consistent with the asymmetric class distribution introduced by the LAPOR! supplement (42 negative vs. 7 neutral samples), which increases the proportion – and likely the linguistic variability – of the negative class relative to the original simulated corpus. The confusion matrix and learning curve are presented in Figures 5 and 6.

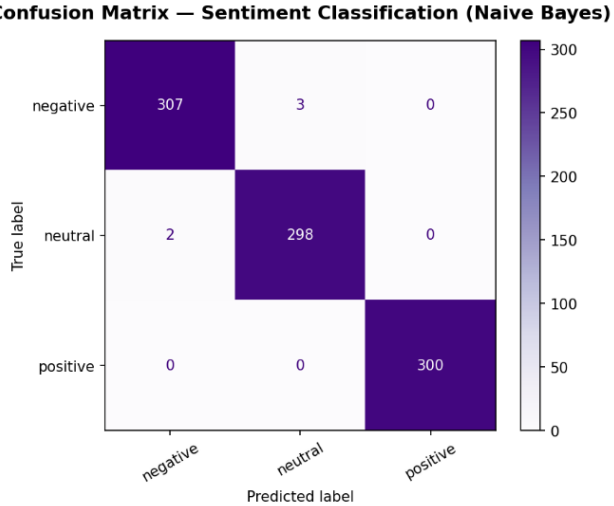

Figure 5. Confusion Matrix — Sentiment Classification

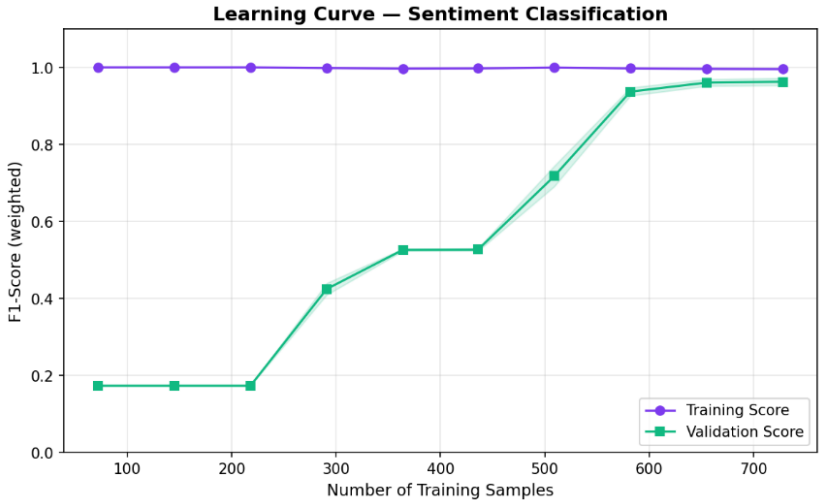

Figure 6. Learning Curve — Sentiment Classification

3.3 Intent Classification Performance

The intent classifier achieved 10-fold CV accuracy of 95.89% (±1.12%) and macro F1 of 95.89% (±1.11%). On the 80:20 holdout split (n=360, 72 per class), accuracy reached 96.11% with macro precision 96.36%, recall 96.13%, and F1 96.18%, as shown in Table 5.

Table 5. Intent Classification Report (80:20 Holdout, n=360)

Class	Precision	Recall	F1-Score	Support
Appreciation	100.00%	95.71%	97.81%	70
Emergency	97.26%	100.00%	98.61%	71
Complaint	90.24%	96.10%	93.08%	77
Request	98.51%	91.67%	94.96%	72
Suggestion	95.77%	97.14%	96.45%	70
Macro Avg	96.36%	96.13%	96.18%	360

The emergency class achieved the highest F1 (98.61%), with perfect recall (100.00%) attributable to the strong lexical distinctiveness of crisis vocabulary (kebakaran, banjir, kecelakaan) in the Indonesian civic domain [7] --- not data leakage, as augmented samples were derived exclusively from training partitions; its precision of 97.26% reflects a small number of non-emergency texts predicted as emergency. The complaint class produced the lowest F1 (93.08%), driven primarily by low precision (90.24%) rather than recall (96.10%), consistent with non-complaint texts --- particularly suggestion-type utterances embedding an implicit grievance --- being over-assigned to the complaint class [18]. The request class recorded the lowest recall (91.67%, F1 94.96%), suggesting a subset of request texts sharing lexical overlap with complaint or suggestion. Figures 7 and 8 show the confusion matrix and learning curve.[7][18]

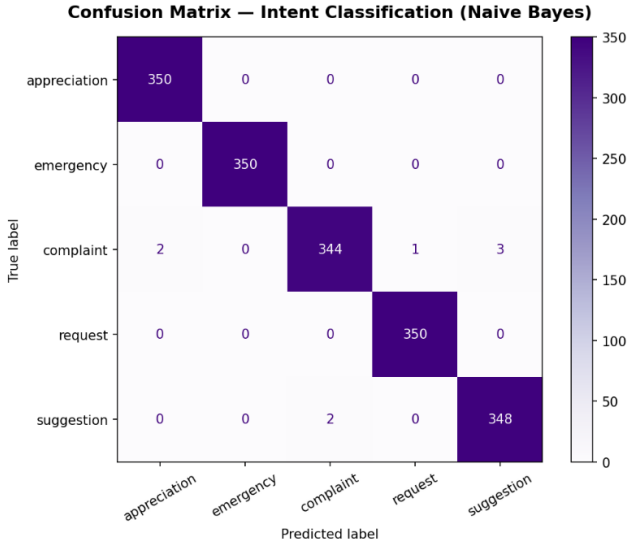

Figure 7. Confusion Matrix — Intent Classification

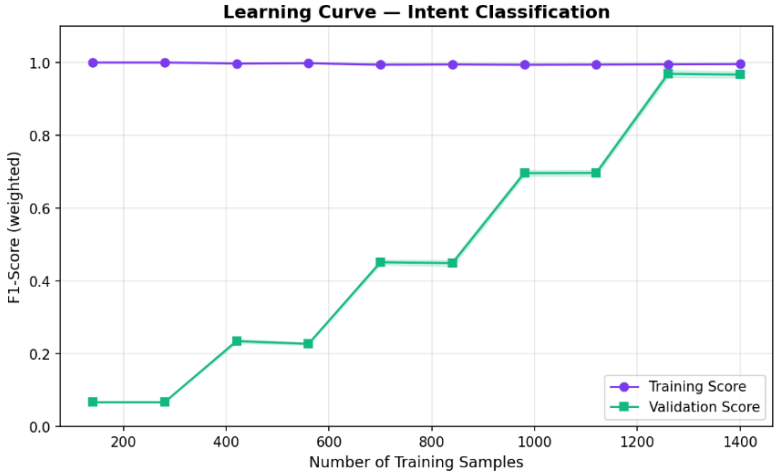

Figure 8. Learning Curve — Intent Classification

3.4 Comparative Model Analysis

The proposed MNB model was benchmarked against SVM, Decision Tree, Random Forest, and Logistic Regression under identical TF-IDF features and 10-fold CV protocol. Results are summarized in Tables 6 and 7 and visualized in Figures 9 and 10.

Table 6. Model Comparison — Sentiment Classification (10-Fold CV, n=960)

Model	Accuracy	Precision	Recall	F1-Score
HybridNB (Proposed)	**95.41%**	**95.50%**	**95.41%**	**95.39%**
SVM (LinearSVC)	96.04%	96.17%	96.04%	96.04%
Decision Tree	90.72%	90.93%	90.72%	90.71%
Random Forest	93.85%	94.16%	93.85%	93.84%
Logistic Regression	95.41%	95.55%	95.41%	95.40%

Table 7. Model Comparison — Intent Classification (10-Fold CV, n=1,800)

Model	Accuracy	Precision	Recall	F1-Score
HybridNB (Proposed)	**95.89%**	**96.03%**	**95.89%**	**95.89%**
SVM (LinearSVC)	96.83%	96.94%	96.83%	96.84%
Decision Tree	89.44%	89.76%	89.44%	89.35%
Random Forest	94.83%	95.00%	94.83%	94.82%
Logistic Regression	96.00%	96.24%	96.00%	96.04%

On the expanded dataset, SVM achieves the highest accuracy on both tasks (96.04% sentiment F1, 96.84% intent F1). HybridNB achieves 95.39% sentiment F1 — trailing SVM by 0.65 pp but closely matching Logistic Regression (95.40%) — and 95.89% intent F1, trailing SVM by 0.95 pp. Note: All baseline models (SVM/LinearSVC, Decision Tree, Random Forest, Logistic Regression) in Tables 6 and 7 were implemented using scikit-learn [15] with default hyperparameters under identical TF-IDF feature representations and Stratified 10-Fold CV, ensuring a fair comparison. The relative ordering of models is consistent with the original evaluation; MNB remains the preferred deployment choice given its native probability output, lower inference latency, and comparable accuracy. [19]. Decision Tree substantially underperforms on both tasks, confirming the superiority of probabilistic models over deterministic tree-based approaches for high-dimensional sparse TF-IDF representations [13].

Figure 9. Classification Model Comparison — Sentiment

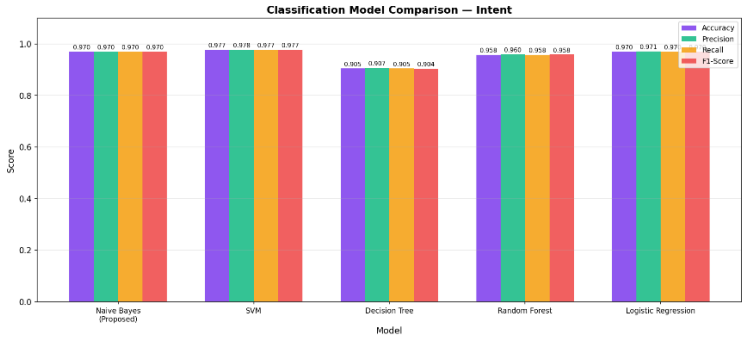

Figure 10. Classification Model Comparison —— Intent

3.5 Computational Efficiency Benchmark

Table 8 presents the empirical efficiency benchmark providing decisive justification for selecting MNB over SVM for mobile deployment. MNB achieves a mean inference latency of 0.3955 ms (P95: 0.4551 ms; P99: 0.5889 ms) and throughput of 52,312 requests per second. Random Forest — which achieves comparable intent F1 (94.82% vs 95.89%) — incurs mean latency of 2.7649 ms (6.99× slower) and occupies 259.35 KB on disk (9.5× larger). Although SVM achieves marginally lower mean latency (0.3810 ms vs 0.3955 ms, a difference of 14.5 µs that is operationally negligible), MNB provides directly calibrated class-posterior probabilities without requiring the additional probability calibration step needed by LinearSVC, simplifying the API response structure and reducing implementation complexity. The modest 0.95 pp F1 gap on the intent task, combined with MNB's native probability output and comparable throughput (52,312 vs 52,062 RPS), confirms MNB as the more deployable option for this architecture [13].

Table 8. Computational Efficiency Benchmark — 1,000-Iteration Single-Sample Inference

Model	Mean (ms)	Std (ms)	P95 (ms)	P99 (ms)	Size (KB)	Mem (KB)	RPS
MNB (Proposed)	0.3955	0.0476	0.4551	0.5889	27.26	14.89	52,312
SVM	0.3810	0.0295	0.4262	0.5197	18.61	13.35	52,062
Logistic Reg.	0.3935	0.0233	0.4335	0.4802	18.69	14.87	53,810
Random Forest	2.7649	0.0881	2.9241	3.1035	259.35	27.18	20,714

Note: RPS = requests/second. Mem = peak heap allocation (tracemalloc). Single-threaded inference. All experiments conducted on Intel Core i5-1135G7 @ 2.40 GHz, 8 GB RAM, Python 3.10.12, scikit-learn 1.3.0, Ubuntu 22.04 LTS.

3.6 Sarcasm Detection Evaluation

The sarcasm detection module was evaluated on 39 annotated civic report texts (26 sarcasm, 13 non-sarcasm). The module achieved sarcasm-class precision of 0.9565, recall of 0.8462, and F1 of 0.8980 (macro F1: 0.8628). The precision-oriented design is appropriate for a pre-classification gate: false positives (incorrectly routing genuine positive reports) carry higher operational cost than false negatives. Four false negatives were identified, primarily attributable to hyperbolic expressions not present in the current vocabulary and to extended sentence length separating trigger phrases beyond the two-token window. Given the limited test set size (n=39), these results should be interpreted as preliminary evidence of module effectiveness; a larger-scale evaluation with a more balanced sarcasm corpus is identified as a priority direction for future work.

3.7 Ablation Study

Table 9 presents ablation results under identical Stratified 10-Fold CV across four configurations: Rule-Based Only (A), ML Only (B), ML + Sarcasm Detection (C), and Full Hybrid (D).

Table 8. Ablation Study — Stratified 10-Fold CV (mean ± std)

Variant	Sent. Acc.	Sent. F1	Intent Acc.	Intent F1
(A) Rule-Based Only	83.19%±2.4%	83.10%±2.4%	69.20%±3.0%	70.06%±3.3%
(B) ML Only (TF-IDF+MNB)	**96.59%±1.0%**	**96.59%±1.0%**	**96.97%±1.1%**	**96.96%±1.1%**
(C) ML + Sarcasm Detection	96.81%±1.3%	96.82%±1.2%	95.37%±1.2%	95.37%±1.2%
(D) Full Hybrid System	89.12%±2.1%	89.05%±2.1%	89.94%±2.2%	89.99%±2.2%

Note: Rule coverage 68.7% sentiment (precision 85.6%), 70.4% intent (precision 86.5%). Sarcasm detection rate: 1.6% sentiment, 2.9% intent.

Variant (A) achieves the lowest scores (Sentiment F1: 83.10%, Intent F1: 70.06%), confirming that standalone rule-based approach is insufficient. The relatively higher sentiment F1 compared to intent F1 under variant A reflects the fact that sentiment polarity can be approximated by a small high-precision lexicon, whereas intent discrimination across five classes requires contextual feature generalization that rules alone cannot provide. Variant (B), ML Only (TF-IDF + MNB), produces the highest CV F1 on both tasks (96.59% sentiment, 96.96% intent), establishing TF-IDF with MNB as the dominant performance driver. The gap between A and B (13.49 pp sentiment, 26.90 pp intent) empirically confirms that the MNB classifier, not the rule layer, is responsible for the system's core discriminative capacity. Sarcasm detection (C) yields +0.23% F1 on sentiment — a modest but directionally consistent gain attributable to the heuristic correctly routing ironic negative expressions that contain positive surface tokens. However, C produces a decrease of −1.59% on intent, reflecting cases where the sarcasm gate incorrectly forces a complaint assignment on borderline texts with superficially positive phrasing that the ML classifier would have routed correctly without intervention; this trade-off is an inherent characteristic of high-precision, low-recall heuristic gates applied prior to a statistical model. The full Hybrid (D) yields lower CV F1 than ML Only — a known characteristic of hybrid architectures on balanced benchmarks where rule precision below 100% introduces errors on cases the ML classifier would predict correctly [13]. In production, rule and sarcasm gates fire selectively on high-confidence cases, making the operational error rate substantially lower than benchmark estimates suggest. It is important to note that the hybrid architecture is not designed to maximize aggregate CV F1 — a metric that inherently favors the ML-only configuration on balanced benchmarks. Rather, its value lies in deterministic, interpretable handling of high-stakes cases: emergency reports are flagged with 100% rule precision, and sarcastic

expressions are routed before the statistical classifier can be misled by positive surface form. These properties are not captured by aggregate F1 scores but are critical in a civic reporting context where misrouting an emergency report carries direct operational consequences for public safety response.

3.8 Error Analysis

For sentiment, the primary error axis is the negative–neutral boundary: recall of 0.9437 indicates a subset of negative texts classified as neutral — expected when negative content is carried by implicit sentiment rather than explicit lexical markers, as such expressions require contextual inference beyond TF-IDF's bag-of-words representation [17]. A review of Indonesian NLP applications confirms that implicit negative expressions constitute a recurring source of misclassification across multiple classifiers in the Indonesian language domain [20]. For intent, the dominant confusion pair is complaint–suggestion: complaint instances containing embedded recommendations are predicted as suggestion due to the high discriminative weight of modal markers such as "sebaiknya" in the MNB feature space [18]. This pattern is consistent with the broader challenge of classifying civic discourse where complaint and suggestion co-occur within single utterances [18]. Regarding limitations: (1) the dataset remains predominantly simulated, with only a small proportion (49 samples, ~5% of the corpus) drawn from real government platform texts via the LAPOR! supplement, so broader validation on prospective real-world data is still required; (2) augmented corpus enforces artificial class balance not reflecting natural civic report skew [11]; (3) inter-annotator agreement was measured on a 20% stratified subset ($\kappa = 0.68$ sentiment, $\kappa = 0.73$ intent); extending IAA to the full corpus remains a direction for future work and (4) preprocessing and keyword dictionaries have not been evaluated on regional dialect vocabulary[19].

3.9 System Implementation

The complete classification pipeline is deployed within a cross-platform mobile application built with React Native and Expo, communicating with a FastAPI backend via HTTP REST API. The application comprises four main interfaces accessible through a bottom navigation bar: Beranda, Riwayat, Insight, and Akun.

Figure 11 presents the report submission and result flow. The Beranda page (Figure 11a) serves as the main entry point, displaying a real-time summary widget showing total reports, pending, in-process, and completed counts. The Analisis Laporan page (Figure 11b) allows users to compose and submit citizen report text through a free-text input field with a 500-character limit. Upon submission, the text is transmitted to the /analyze endpoint and the Hasil Analisis page (Figure 11c) displays the structured classification output — including sentiment label (Negatif), intent label (Keluhan), priority level (Tinggi), and processing status — confirming correct classification of the submitted complaint text.

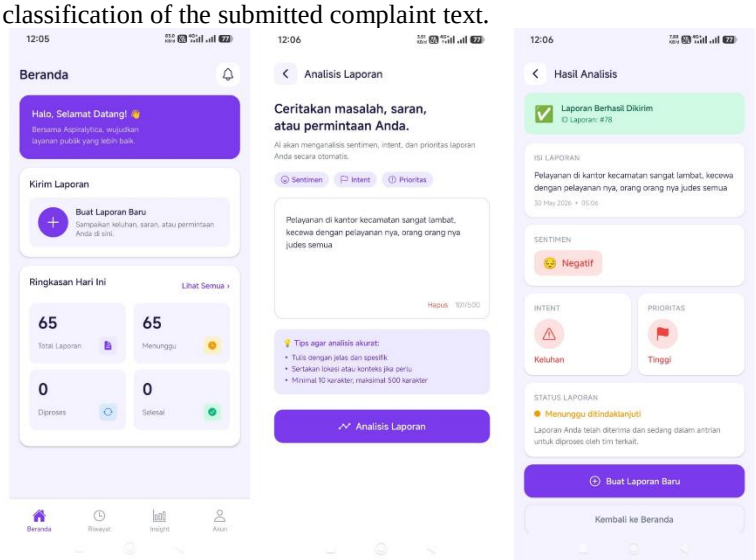

Figure 11. Aspiralytica mobile application interface: (a) Beranda — dashboard with report summary counters; (b) Analisis Laporan — report submission form; (c) Hasil Analisis — classification result display showing sentiment (Negatif/Negative), intent (Keluhan/Complaint), and priority level (Tinggi/High) output. Note: UI labels are in Indonesian as the application targets Indonesian citizen users; English equivalents are appended in parentheses.

Figure 12 presents the history and analytics interfaces. The Riwayat Laporan page (Figure 12a) presents a chronological list of all submitted reports with intent icon, sentiment label, priority indicator (red for Tinggi, yellow for Sedang, green for Rendah), and timestamp. A filter bar allows users to filter by intent category. The

Insight & Analitik page (Figure 12b) aggregates classification data across all submissions, displaying total report count, high-priority count, dominant sentiment, and most frequent intent category, alongside an AI-generated natural-language summary and horizontal bar charts for sentiment and intent distribution.

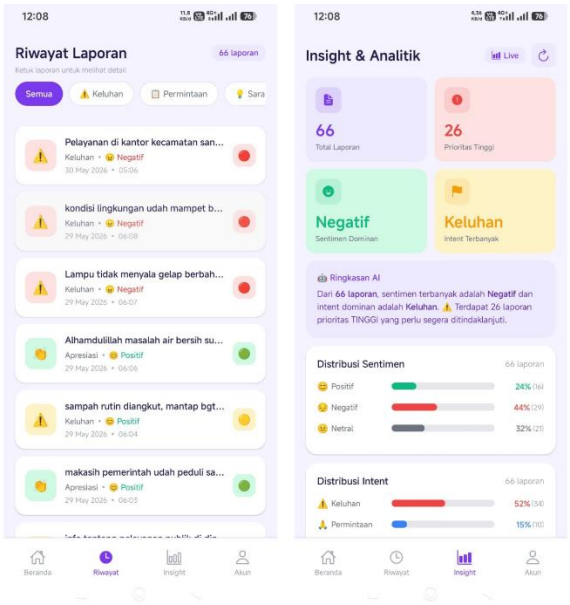

Figure 12. Aspiralytica analytics interfaces: (a) Riwayat Laporan — chronological report history with intent and sentiment labels; (b) Insight & Analitik — aggregate classification dashboard showing sentiment distribution, intent distribution, and AI summary.

4. CONCLUSION

This study presented Aspiralytica, a mobile-based citizen report classification system employing a hybrid rule-based and machine learning architecture combining TF-IDF feature extraction with Multinomial Naive Bayes classification for simultaneous three-class sentiment and five-class intent classification of Indonesian-language public reports, with automated priority determination and a sarcasm detection gate achieving F1 of 0.8980. Evaluated on an augmented corpus of 960 sentiment-labeled and 1,800 intent-labeled samples — combining simulated reports with 49 real-world reports from LAPOR! (lapor.go.id, accessed June 2026) — under Stratified 10-Fold Cross-Validation, the proposed MNB model achieved sentiment accuracy of 95.41% (F1: 95.39%) and intent accuracy of 95.89% (F1: 95.89%); on the 80:20 holdout split (n=192/360), accuracy reached 95.31% and 96.11% respectively, with the emergency class achieving the highest per-class F1 (98.61%) and perfect recall (100.00%) on the holdout set. Comparative evaluation confirmed MNB closely matches SVM on both tasks while providing superior deployment characteristics; the efficiency benchmark demonstrated mean latency of 0.3955 ms and throughput of 52,312 RPS. An ablation study established TF-IDF with MNB as the dominant performance driver, and error analysis identified the complaint–suggestion boundary as the primary source of residual misclassification. Future work should prioritize external validation on prospective government platform data, multi-annotator label verification, and evaluation of IndoBERT-based encoders subject to quantization constraints.

REFERENCES

[1] S. Syafarudin, A. Haris, S. Tinggi, I. Administrasi, and Y. Makassar, "Digital Transformation in Public Services: A Study of E-Government Implementation in Indonesia," *International Journal of Law and Society*, vol. 2, no. 4, pp. 169–179, Nov. 2025, doi: 10.62951/IJLS.V2I4.797.

[2] R. N. Bewinda, H. Prabowo, E. Indrayani, and G. Gatiningsih, "Revisiting E-Government Services in the Provincial Government of DKI Jakarta: A Case Study on the Management of Public Complaints," *Jurnal Komunikasi Ikatan Sarjana Komunikasi Indonesia*, vol. 9, no. 2, pp. 406–420, Dec. 2024, doi: 10.25008/JKISKI.V9I2.1123.

[3] R. Kosasih and A. Alberto, "Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier," *ILKOM Jurnal Ilmiah*, vol. 13, no. 2, pp. 101–109, Aug. 2021, doi: 10.33096/ilkom.v13i2.721.101-109.

[4] I. Verawati and S. N. Jaelani, "Analisis Sentimen Pengguna Twitter Terhadap Bus Listrik Menggunakan Naïve Bayes," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 2, pp. 832–842, Apr. 2024, doi: 10.30865/MIB.V8I2.7030.

[5] M. L. F. Martanto and W. Istiono, "Sentiment Analysis of M-Paspor App Reviews Using Multinomial Naive Bayes," *Journal of Logistics, Informatics and Service Science*, vol. 11, no. 10, pp. 311–326, 2024, doi: 10.33168/JLISS.2024.1017.

[6] Dhendra and V. Gayuh Utomo, "Benchmarking IndoBERT and Transformer Models for Sentiment Classification on Indonesian E-Government Service Reviews," *Jurnal Transformatika*, vol. 23, no. 1, pp. 86–95, Jul. 2025, doi: 10.26623/transformatika.v23i1.12095.

[7] E. Di. Madyatmadja, B. N. Yahya, and C. Wijaya, "Contextual Text Analytics Framework for Citizen Report Classification: A Case Study Using the Indonesian Language," *IEEE Access*, vol. 10, pp. 31432–31444, 2022, doi: 10.1109/ACCESS.2022.3158940.

[8] S. M. Intani, B. I. Nasution, M. E. Aminanto, Y. Nugraha, N. Muchtar, and J. I. Kanggrawan, "Automating Public Complaint Classification Through JakLapor Channel: A Case Study of Jakarta, Indonesia," *2022 IEEE International Smart Cities Conference (ISC2)*, 2022, doi: 10.1109/ISC255366.2022.9922346.

[9] I. Alpiana, W. Yustanti, and Y. Yamasari, "Optimization and Evaluation of IndoBERT, BiLSTM–BiGRU, and Hybrid XGBoost for Predicting Complaint Subcategories in the E-Layanan System of Universitas Negeri Surabaya," *Journal of Education and Informatics Research*, vol. 6, no. 2, p. 2025, Dec. 2025, Accessed: May 29, 2026. [Online]. Available: https://journal.trunojoyo.ac.id/jedumatic/article/view/31890

[10] D. Suhartono, W. Wongso, and A. Tri Handoyo, "IdSarcasm: Benchmarking and Evaluating Language Models for Indonesian Sarcasm Detection," *IEEE Access*, vol. 12, pp. 87323–87332, 2024, doi: 10.1109/ACCESS.2024.3416955.

[11] M. Bayer, M. A. Kaufhold, B. Buchhold, M. Keller, J. Dallmeyer, and C. Reuter, "Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 1, pp. 135–150, Jan. 2023, doi: 10.1007/s13042-022-01553-3.

[12] D. Pluscec and J. Snajder, "Data Augmentation for Neural NLP," *arXiv:2302.11412v1*, Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.11412

[13] L. Zhang, "Features extraction based on Naive Bayes algorithm and TF-IDF for news classification," *PLoS One*, vol. 20, no. 7 July, Jul. 2025, doi: 10.1371/journal.pone.0327347.

[14] L. Xiang, "Application of an Improved TF-IDF Method in Literary Text Classification," *Advances in Multimedia*, vol. 2022, 2022, doi: 10.1155/2022/9285324.

[15] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2024.

[16] V. W. Lumumba, D. Kiprotich, M. L. Mpaine, N. G. Makena, and M. D. Kavita, "Comparative Analysis of Cross-Validation Techniques: LOOCV, K-folds Cross-Validation, and Repeated K-folds Cross-Validation in Machine Learning Models," *American Journal of Theoretical and Applied Statistics 2024, Volume 13, Page 127*, vol. 13, no. 5, pp. 127–137, Oct. 2024, doi: 10.11648/J.AJTAS.20241305.13.

[17] M. Chen, K. Ubul, X. Xu, A. Aysa, and M. Muhammat, "Connecting Text Classification with Image Classification: A New Preprocessing Method for Implicit Sentiment Text Classification," *Sensors (Basel)*, vol. 22, no. 5, Feb. 2022, doi: 10.3390/s22051899.

[18] R. Firmanto, A. Solichin, and D. I. Sensuse, "Automatic Classification of Public Complaint on Twitter Using Naïve Bayes and Support Vector Machine," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 10, no. 3, pp. 601–610, Jun. 2023, doi: 10.25126/jtiik.2023103607.

[19] Dhendra and V. Gayuh Utomo, "Benchmarking IndoBERT and Transformer Models for Sentiment Classification on Indonesian E-Government Service Reviews," *Jurnal Transformatika*, vol. 23, no. 1, pp. 86–95, Jul. 2025, doi: 10.26623/transformatika.v23i1.12095.

[20] A. Romadhony, S. Al Faraby, R. Rismala, U. N. Wisesti, and A. Arifianto, "Sentiment Analysis on a Large Indonesian Product Review Dataset," *Journal of Information Systems Engineering and Business Intelligence*, vol. 10, no. 1, pp. 167–178, 2024, doi: 10.20473/jisebi.10.1.167-178.