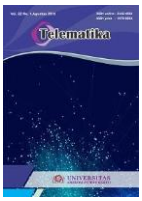

Available online at :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

Telematika

Accreditated SINTA "2" Kemendiktisaintek, No. 10/C/C3/DT.05.00/2025

UNIVERSITAS AMIKOM PURWOKERTO

Toward a Modular, Low-Latency Architecture with BERT-based Big Media Data Analysis

Widyawan[1], Handoko Wisnu Murti[2], Guntur Dharma Putra[3], Eddy Nurmanto[4], Achmad Affandi[5]

[1.3]Department of Electrical Engineering and Information Technology, Faculty of Engineering
[2.4]Semesta Data Digital
[5]Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology
[1.2]Universitas Gadjah Mada, Yogyakarta, Indonesia
[2,4]Yogyakarta, Indonesia
[5]Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
E-mail: widyawan@ugm.ac.id[1], handokowm@semesta.vc[2], gdputra@ugm.ac.id[3], dito@semesta.vc[4], affandi@its.ac.id[5]

ARTICLE INFO

History of the article:
Received June 26, 2025
Revised August 18, 2025
Accepted August 25, 2025

Keywords:
big media data
modular architecture
latency
stream processing

Correspondence:
E-mail: widyawan@ug.ac.id

ABSTRACT

The significant growth of digital and social media platforms has introduced massive streams of unstructured media data. However, current big data approaches are not specifically tailored to the high volume and velocity of media data, which consists of unstructured and lengthy full-text messages. This study proposes a modular and stream-oriented big data architecture for media data. The proposed architecture consists of data crawlers, a message broker, machine learning modules, persistent storage, and analytical dashboards, with a publish-subscribe communication pattern to enable asynchronous, decoupled data processing. The system integrates IndoBERT, a transformer-based model fine-tuned for the Indonesian language, enabling real-time semantic tagging within the streaming pipeline. The proposed solution has been implemented as a prototype using open-source technologies in an on-premise cluster. As such, the primary novelty is the successful integration and operationalization of a large, transformer-based language model (IndoBERT) within a low-latency streaming pipeline. The experimental results underscore the feasibility of deploying scalable, vendor-neutral media analytics platforms for institutions with high sensitivity to privacy and cost. Architectural quality is quantitatively evaluated through Martin's Instability Metric and Coupling Between Objects (CBO), confirming high modularity across components. The system demonstrates an end-to-end latency of 3.121 seconds, a deep learning latency of 2.333 seconds, and processes 32,102 messages per day, making an explicit trade-off where the 2.333-second deep learning inference provides advanced semantic depth. This study presents a reference architecture for scalable, intelligent real-time media analytics systems that support public sector and academic deployments, requiring data privacy and control over infrastructure.

INTRODUCTION

The proliferation of the Internet and digital technology has transformed newspapers into digital news platforms, which has fundamentally shifted readers' behavior from traditionally paper-based consumption toward digital engagement. As of the second quarter of 2024, around 96.2 percent of global users accessed the Internet via mobile phones (Ani Petrosyan, 2025). Boosted by the ubiquity of mobile devices, readers have the luxury of accessing content anytime, anywhere, with the touch of a finger and almost real-time updates (Guo, 2024; Elisa Shearer, 2021). The emergence of Web 2.0 platforms has also fundamentally

shifted web content creation from centralized publisher control to user-generated content (Marx & Cheong, 2023).

The exponential growth of digital news and the widespread integration of social media present both significant opportunities and complex challenges for academic researchers and industry practitioners. With over five billion active users worldwide (Statista, 2024), these platforms generate massive volumes of unstructured data at unprecedented velocities. This data deluge stems from users integrating personal information, behavioral patterns, and daily interactions within digital environments. The resulting data volumes, characterized as "big data," have become a focal point of contemporary research initiatives (Rahul et al., 2023).

Big data is characterized by the 5V's – volume, velocity, variety, veracity, and value – which are the five main innate characteristics that define its nature and potential (Shahnawaz & Kumar, 2025). Volume and velocity are generated from massive daily digital news and social media. Big data variety encompasses datasets in structured, semi-structured, and unstructured formats across diverse domains, including healthcare informatics (Muhunzi et al., 2024), astronomical research (Faaique, 2023), social web analytics (Zhang & Song, 2022), and geoscience applications (Vance et al., 2024). Social media artifacts – including microblog posts, user comments, status updates, and product reviews – represent primary contributors to big data ecosystems, generating massive volumes of unstructured data that drive analytics and decision-making processes (Sang et al., 2024).

To create value for digital news and social media data, information must be systematically extracted, pre-processed, and analyzed (G & Annabel, 2025). Big data analysis encompasses three primary objectives: descriptive, predictive, and prescriptive. Descriptive analysis answers the question of 'what happened', and predictive analysis answers the 'what is likely to happen' scenario. Prescriptive analysis goes beyond prediction by suggesting "what should be done" (Roy Debashish and Srivastava, 2022). Statistical methods and machine learning algorithms play a fundamental role in extracting insights from data (Nti et al., 2022).

However, the fundamental challenge extends beyond data acquisition, processing, and analysis. Big Media Data platform needs to encompass the design of modular and low-latency infrastructure capable of ingesting, processing, storing, analyzing, and visualizing large-scale, multi-source media data streams from hundreds or thousands of digital news outlets and social platforms.

While early big data processing frameworks like the Lambda and Kappa architectures provided foundational models for handling large datasets, they present a significant architectural gap when applied to the unique demands of modern media analytics. These paradigms were primarily designed for transactional, sensor, or log data—formats that are typically structured and concise (Penka et al., 2022; Pal et al., 2018). Their core designs fall short in three critical areas for media data: First, they are not inherently optimized for processing high-velocity streams of large, unstructured, full-text documents, which require more sophisticated parsing and analysis. Second, they lack native, by-design integration for computationally intensive deep learning models, which are essential for extracting nuanced semantic insights from text. Adding such modules to a Lambda or Kappa pipeline often results in a cumbersome, non-scalable bolt-on rather than a seamless component. Third, many recent implementations are tightly coupled with cloud-native ecosystems like Google Cloud Platform and AWS (Sandhu, 2022), creating vendor lock-in and posing significant barriers for organizations with stringent data privacy requirements or cost constraints, such as government agencies or academic institutions. This leaves a clear need for an

architecture that is not only stream-oriented and modular but also purpose-built for deep linguistic analysis in a secure, on-premise environment.

To address the gap, this study adapts a stream model designed explicitly for processing extensive media data. The proposed model encompasses acquisition, modular processing, persistence storing, and interactive analytical capability. NLP (Natural Language Processing) is also integrated into the model as a deep learning module to enhance its analytical depth. Furthermore, to evaluate the architectural quality of the model, this research utilizes the modularity metric measured by Martin's Instability Metric (Sas et al., 2022).

The integration of NLP into real-time data streams is an established approach. Prior implementations in streaming pipelines, however, have often prioritized processing speed over analytical depth, typically employing lightweight models like lexicon-based sentiment analysis or keyword extraction, especially for high-velocity sources like social media feeds. These methods, while fast, often lack deep semantic and contextual understanding. This study advances this area by integrating IndoBERT, a computationally intensive, transformer-based model tailored for the Indonesian language (Koto et al., 2020). Our approach thus distinguishes itself by enabling more nuanced text analysis—such as advanced topic modeling and named entity recognition—within a streaming context. The core challenge addressed here is the operationalization of a large language model in a low-latency environment, striking a novel balance between the demand for real-time processing and the need for high-fidelity semantic extraction from complex media data.

As proof of concept and to prevent cloud-vendor lock-in conditions, this study implements an open-source technology stack that can be deployed in an on-premise fashion. To evaluate the proposed solution, system performance is assessed based on latency and throughput in the delivery of media data.

The remainder of the paper is organized as follows: the research methods section presents the methodology used in the study. The result and discussion section will provide quantitative measurements and discussion. The conclusion and recommendation section will summarize the research and suggest possible further inquiries to enhance the field.

RESEARCH METHODS

This research adopts a design science methodology (Venable et al., 2016). Design science is a research paradigm in computer science, information systems, and software engineering that focuses on the creation and evaluation of purposeful artifacts, such as architecture, models, or systems, that solve real-world problems. This study involves a constructive approach, wherein a model for big data infrastructure is proposed, implemented, and empirically evaluated. The research methodology is detailed in Figure 1.

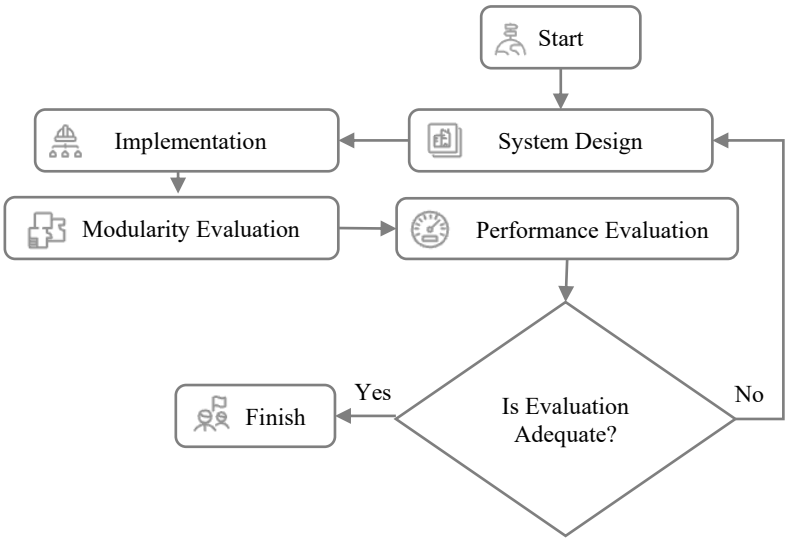

Figure 1. Research Methodology

1. System Design

The proposed model enhances the Kappa architecture for unstructured media-rich data. It integrates data acquisition, modular processing, storing, and interactive visualization. Deep learning modules will also be included.

The model consists of several components: crawler, publisher, message broker, deep learning module, subscriber, cluster database, and dashboard. The proposed model is illustrated in **Error! Reference source not found.**. The components of the model are detailed as follows:

a. Crawler

The crawler module is responsible for extracting content from external media sources, such as news websites, social media APIs, and RSS feeds from digital news sources. The crawler module is also responsible for pre-processing the data, such as removing noise and, importantly, parsing unstructured data into structured formats (e.g., JSON) later needed for downstream processing.

Task:

- Content extraction from different sources (i.e., digital news and social media)
- Data acquisition from social media API (e.g., Twitter /X API) or scraping HTML
- Parse publication date, titles, author, news content, and news outlet

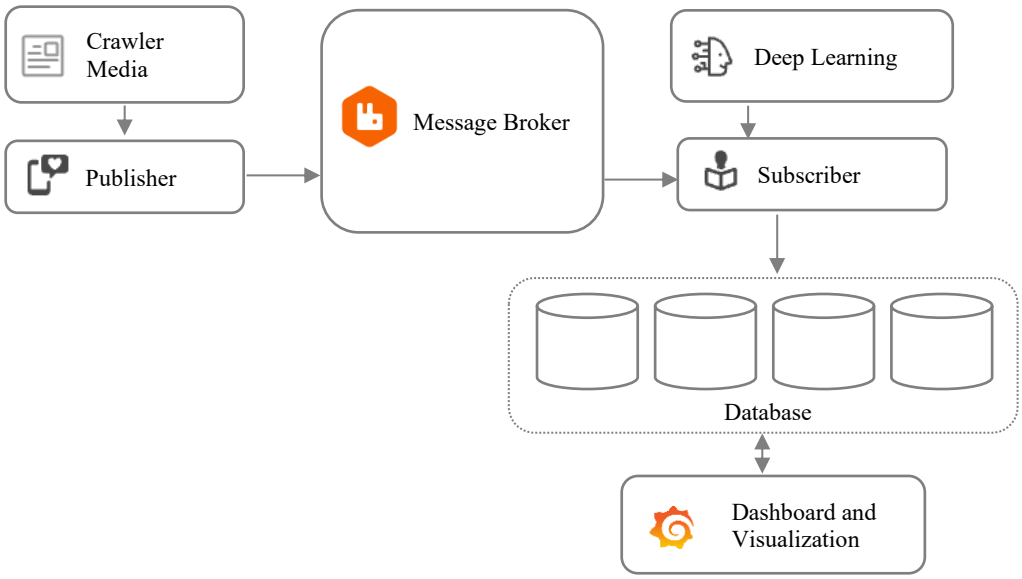

Figure 2 Proposed architecture for Big Media Data

b. Publisher

The publisher module is the message originator. It invokes data from the crawler, generates a message, and sends it to the message broker. In a queue-based system, such as RabbitMQ, it sends the message to a specific queue or exchange.

c. Message Broker

The message broker serves as a central communication layer between components, implementing a publish-subscribe pattern (Saleh et al., 2025). Choosing the publish-subscribe (pub-sub) pattern is a foundational architectural decision in the proposed model. The aim is to provide a modular and scalable platform.

In the pub-sub pattern, loose coupling between components is emphasized. The crawlers publish data without caring which modules will process it. The subscriber will choose and listen to relevant data from the broker. Publisher and subscriber are decoupled; they are unaware of each other's existence. The design will offer benefits, including maximizing modularity.

Pub-sub also enables an asynchronous workflow. For instance, the crawler does not have to wait for deep learning to finish processing data or for database insertion, improving overall latency. Tools such as Kafka (Apache, 2025), RabbitMQ (RabbitMQ, 2025), and MQTT are well-known message brokers that can be utilized.

d. Subscriber

The subscriber module is the consumer of the data stream coming from the message broker. It acts as an adhesive between data streaming and the rest of the system. The subscriber can also perform additional tasks, such as preparing index mapping in the database. Subscriber can invoke deep learning modules in their workflow.

Task:

- Read data from a message queue
- Prepare a map (schema) of the index (table) in the storage engine
- Invoke the deep learning module
- Passing enriched data to the database API

e. Deep Learning

The Deep Learning (DL) module enriches media content with intelligent analysis (Hector et al., 2024). This may include natural language processing (NLP) tasks such as sentiment analysis. The DL module operates on each streamed message received from the broker. This study mainly focuses on Indonesian language media; therefore, a BERT-based model (Devlin et al., 2018) with an Indonesian language focus is utilized. IndoBERT is a monolingual BERT-based language model specifically pre-trained for the Indonesian language. It was introduced in (Koto et al., 2020).

Task:

- Indonesian language tokenization
- Run stream inference on a pre-trained IndoBERT model
- Append analytical data to articles or social media messages.

f. Database

The database stored clean and enriched media content for later analysis and visualization. The database should be able to store large volumes of time series media data. A database capable of horizontal scaling and replication is preferred to ensure high availability and performance (Sundarakumar et al., 2023). The database may also include full-text search and retrieval, allowing flexible queries across content from digital news and social media. Elasticsearch, Splunk, and Apache SOLR are database engines capable of fulfilling those requirements (DB-Engines, 2025).

g. Dashboard and Visualization

The dashboard and visualization components provide a user interface for analyzing enriched media data. The dashboard should be able to interface with and query various types of database engines, allowing flexibility and interoperability for diverse kinds of horizontal and NoSQL databases. The visualization should be able to build and show a time series chart, allowing users to observe trends and patterns in the media over time. Apache Superset, Grafana, and Kibana are some known dashboards capable of business intelligence or time series analysis (Sangeeta Rani, 2025).

h. IndoBERT Integration

Integrating IndoBERT into a streaming pipeline required a dedicated serving stack to balance accuracy and latency. The model is deployed via a Flask-based REST API, decoupled from the pipeline for scalability. High-throughput is attained by distributing inference requests through API load balancing with HAProxy. This approach ensures efficient resource utilization and enables the system to achieve near-real-time performance, with an average latency of 2.333 seconds per message.

2. Implementation

As a proof of concept, this study implements the model in a prototype deployed on an on-premise server cluster. There are four hardware servers, each with the following specifications: AMD EPYC 7252 8-Core Processor, 188 GB RAM, and 2 TB storage. The operating system is Linux Ubuntu 22.04.1 LTS, and they are connected via 1 Gbps LAN. The servers are hosted in the University's data center.

Along with the component's requirement mentioned beforehand, this study also prioritizes open-source software, especially with thriving community support. The list of modules and their corresponding technology stack can be seen in Table 1. The module is deployed in containers and managed by Docker.

Table 1 Modules and Their Corresponding Technology Stack

No.	*Module*	*Technology Stack*
1	Crawler Digital News	Python, JSON, newspaper3k library
	Crawler Social Media	Python, JavaScript, social media API, JSON
2	Message Broker	RabbitMQ, JSON
3	Publisher, Subscriber	Python
4	Deep Learning	Python, Hugging Face and IndoBERT-base model libraries
5	Database	Elasticsearch in a 4-machine cluster
6	Dashboard	Grafana

3. Modularity Evaluation

Modularity means the system is built from independent, interchangeable components (modules), each with a specific responsibility. In the proposed model, the elements include crawlers, queues, deep learning (DL), databases, and dashboards, which are separate modules. Each module can be scaled, replaced, and tested independently.

Modularity is often treated as a *design quality*—a desirable *architectural property* rather than a direct functional requirement. To quantify, modularity can be measured by coupling. Coupling measures the degree to which one module is dependent on others. Lower is better, and each module should be able to perform its task without relying heavily on another. This research adopts a threshold of the CBO (Coupling Between Objects) metric, as defined by Shatnawi (2010), where a value below 9 is considered acceptable or low coupling.

Furthermore, modularity can be measured by Martin's Instability Metric (Martin & October, 1997), comparing afferent coupling and efferent coupling:

$$I = \frac{Fan_{out}}{Fan_{out} + Fan_{in}} \quad (1)$$

Where:

- I: instability level
- Fan_{out}: how many modules does this module depend on (afferent coupling)
- Fan_{in}: how many modules used this module (efferent coupling)

4. Performance Evaluation

System performance is measured in terms of latency and throughput. Latency was measured using direct timestamp sampling, a method commonly used in distributed stream processing systems (Carbone et al., 2015; Karimov et al., 2018). Each message was instrumented with a timestamp at the moment of ingestion and again upon being processed by end modules. Direct timestamp sampling is given by:

$$W = \frac{1}{N}\sum_{i}^{N} W_i \quad (2)$$

Where:

- W: latency

- N: number of measurements
- W_i: message's timestamp difference between modules measured at index i

Throughput is the rate at which a system processes data. In this study, it refers to the number of articles (or messages) successfully processed by a system component per unit of time (Karimov et al., 2018). A general formula of throughput is given by:

$$Througput = \frac{M}{T} \quad (3)$$

Where:

- M: number of messages or articles
- T: interval of measurement

5. Experimental Setting

Latency and throughput are measured in the prototype of the Big Media Data system. The message and its timestamp are collected from the crawler, message broker, subscriber, and databases. The crawler collected articles from 6000 Indonesian digital news outlets for 30 days.

RESULTS AND DISCUSSION

This study evaluated the modularity of the proposed design, described in **Error! Reference source not found.**, in terms of coupling. Coupling measures the degree to which a module is dependent on others. The instability matrix in Equation (1) is used to quantify the module's stability. A summary of the modules and their respective coupling and instability metrics is presented in Table 2.

Table 2 Summary of modules' coupling

No.	*Module*	*Depends on/afferent coupling* Fan_{out}	*Used by/efferent coupling* Fan_{in}	*Instability* I
1	Crawler	2 (external media API, Publisher)	0	1
2	Publisher	1 (Message Broker)	1 (Crawler)	0,5
3	Message Broker	0	2 (Subscriber, Publisher)	0
4	DL	1 (DL external library)	1 (Subscriber)	0.5
5	Subscriber	2 (Message Broker, DL)	1 (Database)	0,67
6	Database	1 (Subscriber)	1 (Dashboard)	0.5
7	Dashboard	1 (Database)	0	1

Overall, both afferent and efferent coupling between modules remains low, below the threshold of 9, due to the message-passing mechanism employed. All components communicate by exchanging data in JSON format, which promotes loose coupling.

Crawlers publish only to the message broker via the publisher; no other modules directly depend on it; therefore, it is highly decoupled. Instability is 1; it can be modified without affecting others. The module dashboard also has the same characteristics. It reads data only from the database; no other module uses it, and it is highly decoupled. The value of one means it can be dynamically changed based on the analytical necessity of the user.

The message broker has an Instability of zero, which means it is stable. As a foundational component that allows message passing and asynchronous communication, it is a desired quality of such a component or module. It is a requirement that such modules never change, as they must be stable; otherwise, all dependent modules would require maintenance whenever changes occur. Furthermore, the modules of DL,

publisher, subscriber, and database also exhibit low coupling. In comparison, the median Instability is 0.5. It means that the modules are in a balanced dependency profile, not overly dependent or heavily dependent upon them.

Latency is measured by capturing message arrival, assigning a timestamp, and calculating the time difference with subsequent modules, as shown in Equation (2). The result of system latency or end-to-end latency measurement is illustrated in Figure 2. It shows the latency between when an article was crawled and when it was written to the Elasticsearch database. Bert-based Deep Learning latency measurement is presented in Figure 3.

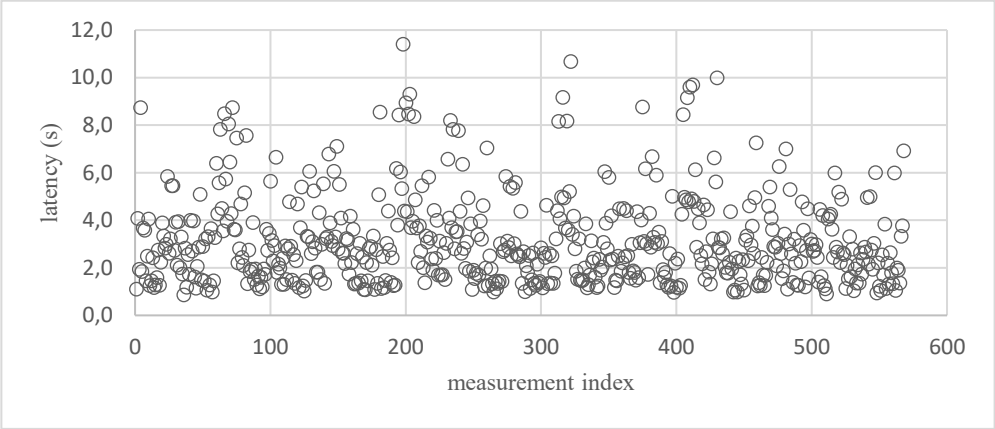

Figure 2 End-to-end latency measurement

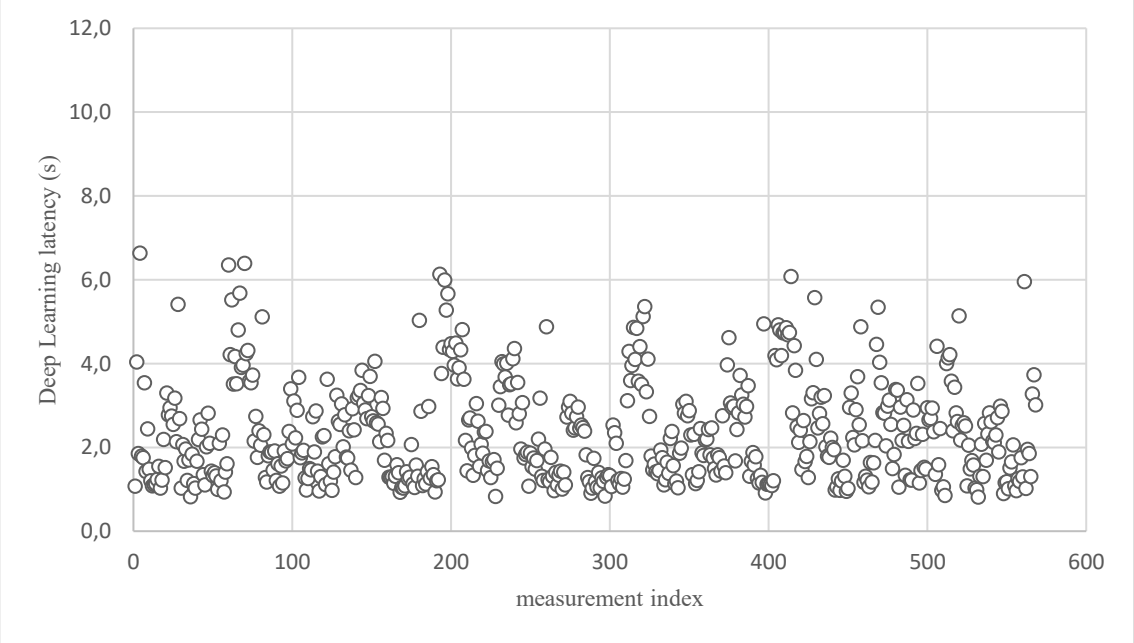

Figure 3 IndoBERT Deep Learning latencies

A summary of the latencies is given in Table 3. End-to-end latency is averaged at 3.121 seconds. Furthermore, it shows 0.787 s, 3.120 s, 0.047 s, and 2.333 s for inter-module latency, message broker latency, and deep learning latency, respectively. The measurement reveals a system with low latency, supporting high throughput in large-scale media data analysis. The message broker, as a core component, maintains sub-second latency - an essential characteristic for handling high-velocity data (Dobbelaere & Esmaili, 2017). The deep learning module uses IndoBERT, an Indonesian-specific transformer model.

Given its size and complexity, some latency during inference per message is expected and reflects its advanced language understanding.

Table 3 Summary of latency measurement

No.	*Module*	*Latency*	*Note*
1	Crawler - Database	3.121 s	End-to-end latency
2	Crawler - Subscriber	0,787 s	Inter-module latency, before invoking DL
3	Publisher - Database	3.120 s	Inter-module latency, after invoking DL
4	Publisher - Subscriber	0.047 s	Message Broker latency
5	Deep Learning	2.333 s	BERT-based inference latency

Throughput is measured by the number of successfully processed messages, from crawler to database module, per unit of time, as described in Equation (3). The crawler collected messages from news outlets; the daily throughput for 30 days is shown in Figure 4. The average throughput is 32,102 messages per day or 22.30 messages per minute.

The system achieves near-real-time analytics with an end-to-end latency of 3.121s, largely due to IndoBERT inference (2.333s), reflecting a trade-off between speed and analytical depth. While lightweight models offer faster keyword extraction, this architecture enables richer semantic analysis crucial for topic modelling and sentiment tasks. Its efficiency is further shown by the message broker's sub-second latency (0.047s). The system processes 32,102 messages per day—equivalent to 22 full-text articles per minute from 6,000 news outlets—demonstrating strong capacity for large-scale media monitoring. Thanks to its modular design, throughput can be scaled horizontally, ensuring long-term viability for real-world deployment.

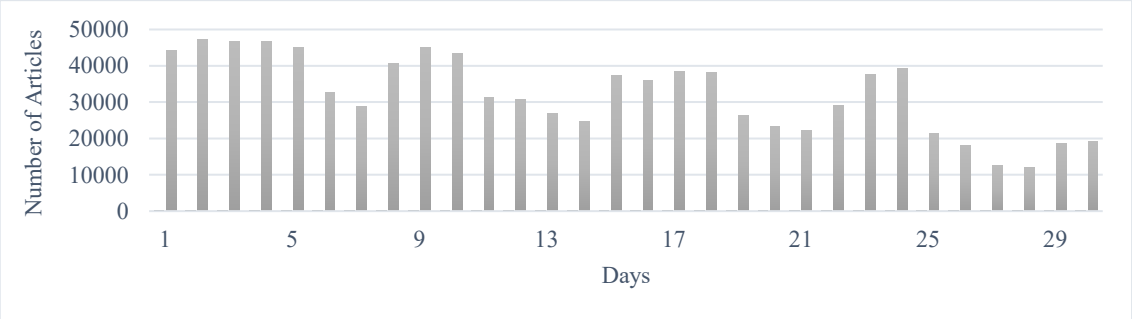

Figure 4 Daily throughput of media data

The significance of these performance metrics, particularly the latency, lies in the deliberate architectural trade-off made to prioritize analytical depth over raw processing speed. While the 2.333-second latency from the IndoBERT module constitutes roughly 75% of the total 3.121-second end-to-end latency, this is not a performance bottleneck but a strategic investment in high-quality insight generation. A comparative baseline using simpler NLP techniques, such as keyword extraction or lexicon-based sentiment analysis, could achieve sub-second latencies but would offer a superficial understanding of the text.

Therefore, the 2.333-second processing time is the cost of transforming raw data into meaningful intelligence. For the target application of strategic media monitoring, where the goal is to understand complex narratives and trends, the value of this enriched analysis far outweighs the need for instantaneous, sub-second processing. The system remains firmly in the "near-real-time" category, delivering deep insights fast enough for effective decision-making.

Most of the modules are back-end service-type applications. However, the RabbitMQ message broker provides a GUI (Graphical User Interface) for queueing analysis and administrative purposes. The GUI can be seen in Figure 5.

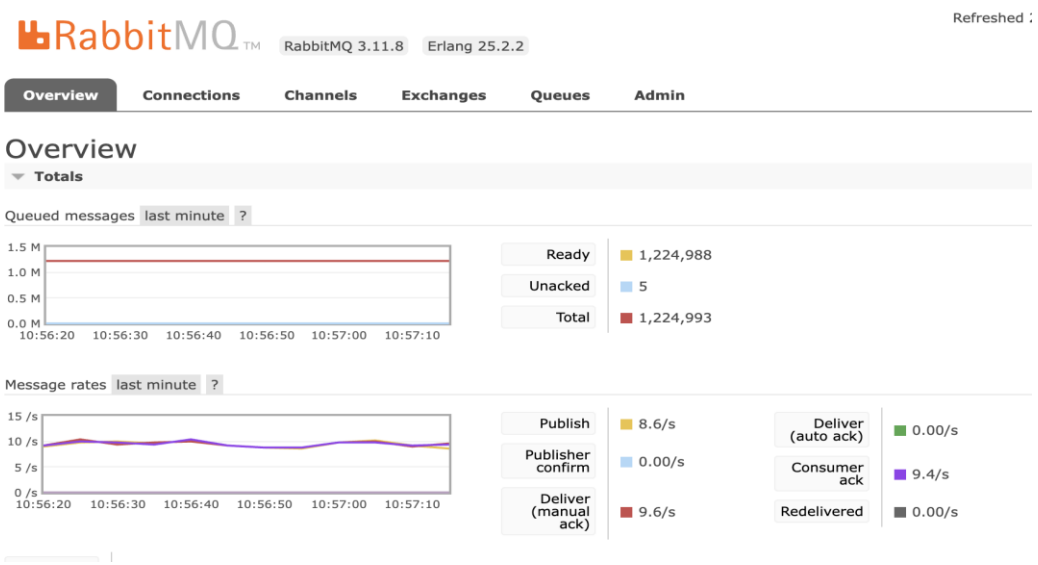

Figure 5 RabbitMQ browser-based management UI

A prototype dashboard was developed using the open-source visualization platform Grafana (Grafana Labs, 2025). The dashboard leverages Grafana's ability to interface with various database engines, most notably Elasticsearch, which provides advanced full-text query capabilities for media data. It also supports time-series analysis out of the box, a feature crucial for media analysis. For academic purposes, the dashboard is publicly available at https://xplore.pustakadata.id/. A screenshot of the dashboard is presented in Figure 6, showing various panels: a time series picker, data filtering, descriptive analytics, and DL chart visualizations.

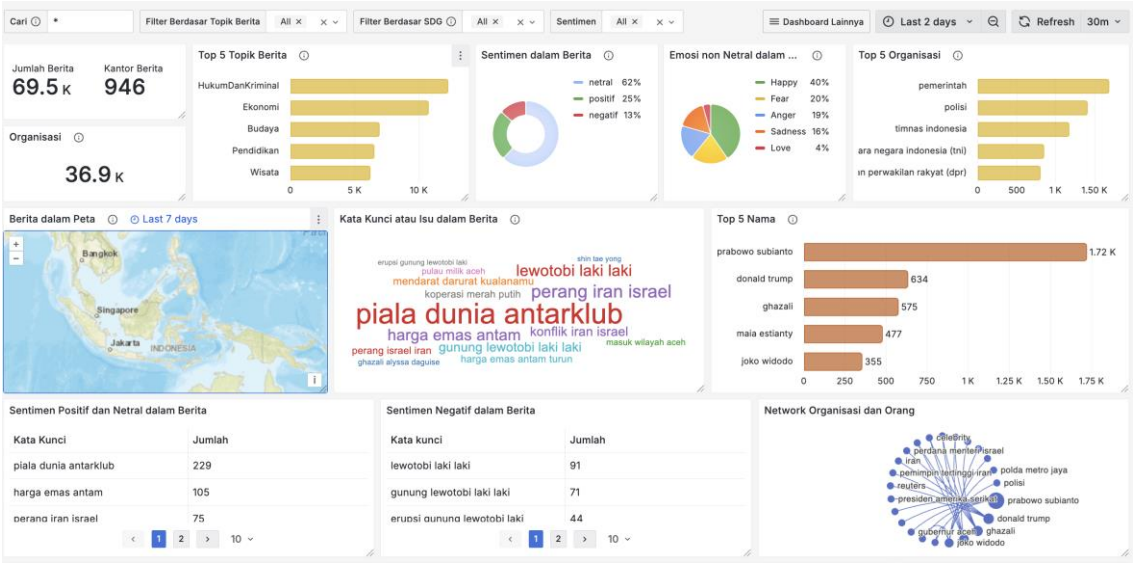

Figure 6 User interface of the Grafana-based dashboard

CONCLUSIONS AND RECOMMENDATIONS

This study has adapted a streaming model for processing unstructured media data. The proposed model components include a crawler for data acquisition, a message broker, deep learning, a database for

persistence storage, and an analytical dashboard. Even though there are additional modules compared to early architecture - DL, database, dashboard - all module coupling remains low, below the CBO threshold of 9.

The modular architectural quality is enhanced by message-passing mechanisms and asynchronous communication enabled by the message broker. The crawler and dashboard modules have an instability metric of 1, indicating they are highly decoupled; they can dynamically change without affecting others. The message broker has an instability of 0, indicating a very stable component, a desirable characteristic of a central communication module.

As a proof of concept, this study has implemented a prototype deployed in an on-premise data center. An open-source technology stack, as shown in Table 1, is utilized. End-to-end latency, which is the time it takes for an article to be crawled, stream-processed by DL, and stored in the database, is averaged at 3.121 seconds. IndoBERT latency is 2.333 seconds. The average throughput is 32,102 articles per day.

Future research could explore architectural support for multimodal fusion. Focusing on synchronizing and processing of video, audio, image, and text streams. This includes standardized reference architecture that defines reusable blueprints (e.g., ingestion layer, storage layer, processing layer, visualization) and cross-platform reference architectures deployable on GPU-supported cloud-native and on-premise systems.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the research funding provided by the Ministry of Higher Education, Science, and Technology through the PUPT program and support from the Indonesia Research and Education Network (IDREN).

REFERENCES

Ani Petrosyan. (2025, February 6). *Share of users worldwide accessing the Internet in 3rd quarter 2024, by device*. Statista. https://www-statista-com.ezproxy.ugm.ac.id/statistics/1289755/internet-access-by-device-worldwide/

Apache. (2025). *Apache Kafka*. https://kafka.apache.org/

Carbone, P., Katsifodimos, A., Kth, †, Sweden, S., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). *Apache Flink™: Stream and Batch Processing in a Single Engine*.

DB-Engines. (2025). *DB-Engines Ranking of Search Engines*. https://db-engines.com/en/ranking/search+engine

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, *abs/1810.04805*. http://arxiv.org/abs/1810.04805

Dobbelaere, P., & Esmaili, K. S. (2017). Industry paper: Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations. *DEBS 2017 - Proceedings of the 11th ACM International Conference on Distributed Event-Based Systems*, 227–238. https://doi.org/10.1145/3093742.3093908

Elisa Shearer. (2021, January 12). *More than eight-in-ten Americans get news from digital devices By*. PewResearch. https://www.pewresearch.org/short-reads/2021/01/12/more-than-eight-in-ten-americans-get-news-from-digital-devices/

Essaidi, A., & Bellafkih, M. (2023). A New Big Data Architecture for Analysis: The Challenges on Social Media. *IJACSA) International Journal of Advanced Computer Science and Applications*, *14*(3). www.ijacsa.thesai.org

Faaique, M. (2023). Overview of Big Data Analytics in Modern Astronomy. *International Journal of Mathematics, Statistics, and Computer Science*, *2*, 96–113. https://doi.org/10.59543/ijmscs.v2i.8561

G, K., & Annabel, S. P. (2025). A survey on big data classification. *Data and Knowledge Engineering*, *156*. https://doi.org/10.1016/j.datak.2025.102408

Grafana Labs. (2025). *Grafana*. https://grafana.com/

Guo, M. (2024). Predictors of Mobile News Consumption through News Applications (Apps): The Impacts of Audience Characteristics, Media Usage, and Motivations. *Journalism and Media*, *5*(3), 1071–1084. https://doi.org/10.3390/journalmedia5030068

Hector, D.-L., Chavoya, A., & Hernandez-Ochoa, M. (2024). The Role of Machine Learning in Big Data Analytics: Current Practices and Challenges. In F. and M. G. J. and D.-L. H. Mora Manuel and Wang (Ed.), *Development Methodologies for Big Data Analytics Systems: Plan-driven, Agile, Hybrid, Lightweight Approaches* (pp. 47–74). Springer International Publishing. https://doi.org/10.1007/978-3-031-40956-1_2

Karimov, J., Rabl, T., Katsifodimos, A., Samarev, R., Heiskanen, H., & Markl, V. (2018). Benchmarking distributed stream data processing systems. *Proceedings - IEEE 34th International Conference on Data Engineering, ICDE 2018*, 1519–1530. https://doi.org/10.1109/ICDE.2018.00169

Koto, F., Rahimi, A., Lau, J. H., & Baldwin, T. (2020). *IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP*. https://arxiv.org/abs/2011.00677

Martin, R. C., & October. (1997). *OO Design Quality Metrics*. https://api.semanticscholar.org/CorpusID:18246616

Marx, J., & Cheong, M. (2023). Decentralised Social Media: Scoping Review and Future Research Directions. *ACIS 2023 Proceedings*.

Muhunzi, D., Kitambala, L., & Mashauri, H. L. (2024). Big data analytics in the healthcare sector: Opportunities and challenges in developing countries. A literature review. In *Health informatics journal* (Vol. 30, Issue 4). https://doi.org/10.1177/14604582241294217

Nti, I. K., Quarcoo, J. A., Aning, J., & Fosu, G. K. (2022). A mini-review of machine learning in big data analytics: Applications, challenges, and prospects. In *Big Data Mining and Analytics* (Vol. 5, Issue 2, pp. 81–97). Tsinghua University Press. https://doi.org/10.26599/BDMA.2021.9020028

Pal, G., Li, G., & Atkinson, K. (2018). Multi-agent big-data lambda architecture model for E-commerce analytics. *Data*, *3*(4). https://doi.org/10.3390/data3040058

Penka, J. B. N., Mahmoudi, S., & Debauche, O. (2022). An Optimized Kappa Architecture for IoT Data Management in Smart Farming. *Journal of Ubiquitous Systems and Pervasive Networks*, *17*(2). https://doi.org/10.5383/juspn.17.02.002

RabbitMQ. (2025). *RabbitMQ, One broker to queue them all*. https://www.rabbitmq.com/

Rahul, K., Banyal, R. K., & Arora, N. (2023). A systematic review on big data applications and scope for industrial processing and healthcare sectors. *Journal of Big Data*, *10*(1). https://doi.org/10.1186/s40537-023-00808-2

Roy Debashish and Srivastava, R. and J. M. and K. M. S. (2022). A Complete Overview of Analytics Techniques: Descriptive, Predictive, and Prescriptive. In T. and H.-P. D. and S. T. P. and A. S. Jeyanthi P. Mary and Choudhury (Ed.), *Decision Intelligence Analytics and the Implementation of Strategic Business Management* (pp. 15–30). Springer International Publishing. https://doi.org/10.1007/978-3-030-82763-2_2

Saleh, A., Morabito, R., Dustdar, S., Tarkoma, S., Pirttikangas, S., & Lovén, L. (2025). Towards Message Brokers for Generative AI: Survey, Challenges, and Opportunities. *ACM Comput. Surv.* https://doi.org/10.1145/3742891

Sandhu, A. K. (2022). Big Data with Cloud Computing: Discussions and Challenges. *Big Data Mining and Analytics*, *5*(1). https://doi.org/10.26599/BDMA.2021.9020016

Sang, V. M., Thanh, T. N. P., Gia, H. N., Quoc, D. N., Long, K. Le, & Yen, V. P. T. (2024). Impact of user-generated content in digital platforms on purchase intention: the mediator role of user emotion in the electronic product industry. *Cogent Business & Management*, *11*(1), 2414860. https://doi.org/10.1080/23311975.2024.2414860

Sangeeta Rani. (2025). Tools and techniques for real-time data processing: A review. *International Journal of Science and Research Archive*, *14*(1), 1872–1881. https://doi.org/10.30574/ijsra.2025.14.1.0252

Sas, D., Avgeriou, P., & Uyumaz, U. (2022). On the evolution and impact of architectural smells—an industrial case study. *Empirical Software Engineering*, *27*(4). https://doi.org/10.1007/s10664-022-10132-7

Shahnawaz, M., & Kumar, M. (2025). A Comprehensive Survey on Big Data Analytics: Characteristics, Tools and Techniques. In *ACM Computing Surveys* (Vol. 57, Issue 8, pp. 1–33). Association for Computing Machinery. https://doi.org/10.1145/3718364

Shatnawi, R. (2010). A quantitative investigation of the acceptable risk levels of object-oriented metrics in open-source systems. *IEEE Transactions on Software Engineering*, *36*(2), 216–225. https://doi.org/10.1109/TSE.2010.9

Statista. (2024). *Number of social media users worldwide from 2017 to 2028*.

Sundarakumar, M. R., Mahadevan, G., Natchadalingam, R., Karthikeyan, G., Ashok, J., Manoharan, J. S., Sathya, V., & Velmurugadass, P. (2023). A comprehensive study and review of tuning the performance on database scalability in big data analytics. *Journal of Intelligent & Fuzzy Systems*, *44*(3), 5231–5255. https://doi.org/10.3233/JIFS-223295

Vance, T. C., Huang, T., & Butler, K. A. (2024). Big data in Earth science: Emerging practice and promise. *Science*, *383*(6688), eadh9607. https://doi.org/10.1126/science.adh9607

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89. https://doi.org/10.1057/ejis.2014.36

Zhang, H., & Song, M. (2022). How Big Data Analytics, AI, and Social Media Marketing Research Boost Market Orientation. *Research-Technology Management*, *65*(2), 64–70. https://doi.org/10.1080/08956308.2022.2022907