Journal of Earth and Planetary System Science
Volume 1, Issue 1, 52-61.
e_ISSN:
https://journal.gomit.id/jepss

Transformation of Meteorological Fire Data into Interactive Narrative Using Large Language Gemini Model Through Telegram Chatbot

Cahyo Adi Nugroho*
Sekolah Tinggi Meteorologi Klimatologi dan Geofisika, INDONESIA

Giarno
Sekolah Tinggi Meteorologi Klimatologi dan Geofisika, INDONESIA

* Cahyo Adi Nugroho, Sekolah Tinggi Meteorologi Klimatologi dan Geofisika, INDONESIA.
✉ cahyoadinugroho10@gmail.com

Article Info

Article history:

Received: July 25, 2026
Revised: Agustus 19, 2026
Accepted: Agustus 30, 2026

Keywords:

Chatbot Telegram
Large Language Model (LLM)
Natural Language Understanding (NLU)
Two-Stage Pipeline

Abstract

Background: Weather information plays an important role in supporting people's daily activities, including travel, agriculture, outdoor activities, and decision-making. However, raw meteorological data delivered through Application Programming Interface (API) services is commonly presented in JSON format, which is highly technical and difficult for the general public to interpret.
Aims: This research aims to democratize access to raw weather data by developing an interactive and personalized Telegram chatbot that serves as an intermediary system (middleware) between users and external weather data APIs. The system is designed to provide weather information in a natural, contextual, and user-friendly form.
Methods: The proposed system implements a Two-Stage Large Language Model (LLM) Pipeline architecture utilizing the Gemini 3.1 Pro model. The architecture separates the LLM workload into two specialized stages. The first stage, the Router Engine, functions as a probabilistic semantic classifier to identify and categorize user intentions. The second stage, the Formatter Engine, acts as a contextual text generator that processes factual JSON weather data obtained from external APIs and transforms it into concise, casual, and understandable narratives
Result: The evaluation results demonstrate high system stability, with context recognition accuracy reaching 98.8%. The proposed processing architecture also achieves network response latency below 2.5 seconds. Furthermore, the graceful degradation mechanism demonstrates strong fault tolerance when the system encounters API quota limitations.
Conclusion: The separation of cognitive roles within the LLM, combined with external factual data extraction, effectively reduces cognitive barriers to weather information and mitigates the risk of hallucination.

To cite this article: : Nugroho.C.A., Giarno. (2026). Transformation of Meteorological Fire Data into Interactive Narrative Using Large Language Gemini Model Through Telegram Chatbot. *Journal of Earth and Planetary System Science, 1*(1), 52-61.

INTRODUCTION

Weather information is a crucial aspect that greatly influences people's daily activities and decision-making (Lazo et al., 2009; Demuth et al., 2012). Quick access to weather forecast information is very important for the smooth functioning of public mobility. Although the provision of meteorological data through global services is currently highly advanced, there is still a significant communication gap between the form of technical raw data and the level of general public understanding. The data provided by *Application Programming Interface* (API) services is generally purely *machine-to-machine* communication with a structured format such as JSON, which is very rigid and loaded with technical variables. For the general public, the need to read the rows of programming texts and raw numbers is a cognitive barrier. On the other hand, the majority of conventional weather

apps today only display numerical indicators or raw graphs without any derivative explanation or adequate context. This condition shows a lack of personalization of the interface in conveying information that is appropriate to a variety of daily user activities. It has been academically proven that the delivery of technical weather information without an adaptive and inclusive public communication approach actually leads to a low level of public understanding of potential hazards (Arif et al., 2025). As a result, meteorological data, which should be a valuable instrument for mitigation and daily planning, is less than optimally utilized by the wider community.

To overcome these interface gaps, an intermediary system (*middleware*) is needed that is able to transform technical data into consumption-ready information. The development of a digital-based intermediary system and popular communication media is crucial so that all levels of society can respond to climate information equally, quickly, and effectively for mitigation purposes (Pramono et al., 2023; Schramm et al., 2021). The approach proposed in this study is to implement Natural *Language Understanding* technology from the *Large Language Model* (Zhao et al., 2023; Brown et al., 2020) to create more fun, adaptive, and personalized interactions. The use of virtual assistants (chatbots) with human-like conversational capabilities has been proven to be able to significantly reduce the cognitive load of users when compared to conventional data table readings (Rapp et al., 2021; Brandtzaeg & Følstad, 2017; Kocaballi et al., 2019). In line with previous research on the implementation of contextual artificial intelligence for climate data (Navajyoti, 2020), this study designed an interactive Telegram *chatbot* system that adopts a two-stage pipeline architecture to democratize raw weather data.

In such architectures, artificial intelligence is specifically optimized as a logical center for contextualizing and translating data, while mitigating the risk of information hallucinations through the retrieval of external factual data (Lewis et al., 2020; Bender et al., 2021). Through this system, the machine is capable of detecting the user's intent from unstructured free sentences, then pulling specific meteorological data according to the requested location. Once the numerical data is obtained, the system transforms it into a communicative, casual textual narrative, while also processing the visualization logic to generate an easy-to-understand weather trend graph. The complete integration between text processing and dynamic visualization is expected to facilitate access to information on air conditions, so that public awareness of the local climate around them can continue to increase through routine, transparent, and personalized delivery.

METHOD

Research Data Sources

This virtual assistant system relies heavily on dynamic, real-time data retrieval from an external knowledge base to prevent hallucination phenomena in language models. The data used consists of primary and secondary meteorological data accessed directly through a public *Application Programming Interface* (API) without the storage of intermediate databases. Open-Meteo API, used as a primary meteorological data source for providers of quantitative weather variables, such as actual temperature, humidity, precipitation, and wind speed. This data is retrieved at an hourly resolution transmitted in a *nested dictionary* structure in JSON (*JavaScript Object Notation*) format, which is a standard format for lightweight, independent, and highly efficient data exchange formats to be parsed by machines (Bray, 2014). Open Data from BMKG, METEOBLUE, EUMETSTAT (such as DigitalForecast.xml) and Meteoblue are used as comparative and fallback options to verify local climate data hyperlocally in Indonesia. This comparative verification is very crucial considering the high weather dynamics and spatial variability of atmospheric boundaries in the equatorial region (Kristianto & Rani, 2019)

Workflow Modeling and Finite Automata

The overall flow of program control is modeled as a *Deterministic Finite Automata* (DFA) to ensure that system state transitions run predictably and consistently when processing user queries

(Hopcroft et al., 2001; Jurafsky & Martin, 2023). The processing state set is defined as , with an input alphabet representing the incoming text message, and a transition function. $S = \{S_0, S_1, S_2, S_3, S_4\}\Sigma\delta$ *Infinite polling loops* are activated via the *bot.infinity_polling() function* to monitor the arrival of messages in a discrete time sequence$M_t t$. This poll architecture approach on the Telegram API is academically proven to be very lightweight, reliable, and minimizes the failure to send real-time weather notifications without burdening the client's software (Firmansyah, 2022):

$$While\ True\colon\ \delta(S_0, M_t) \rightarrow S_1$$

Where it represents the status of the bot listening to S_0*long-polls*, is the activation of S_1*the Router Engine*, is a common chat line, and is a specific weather processing. S_2S_3

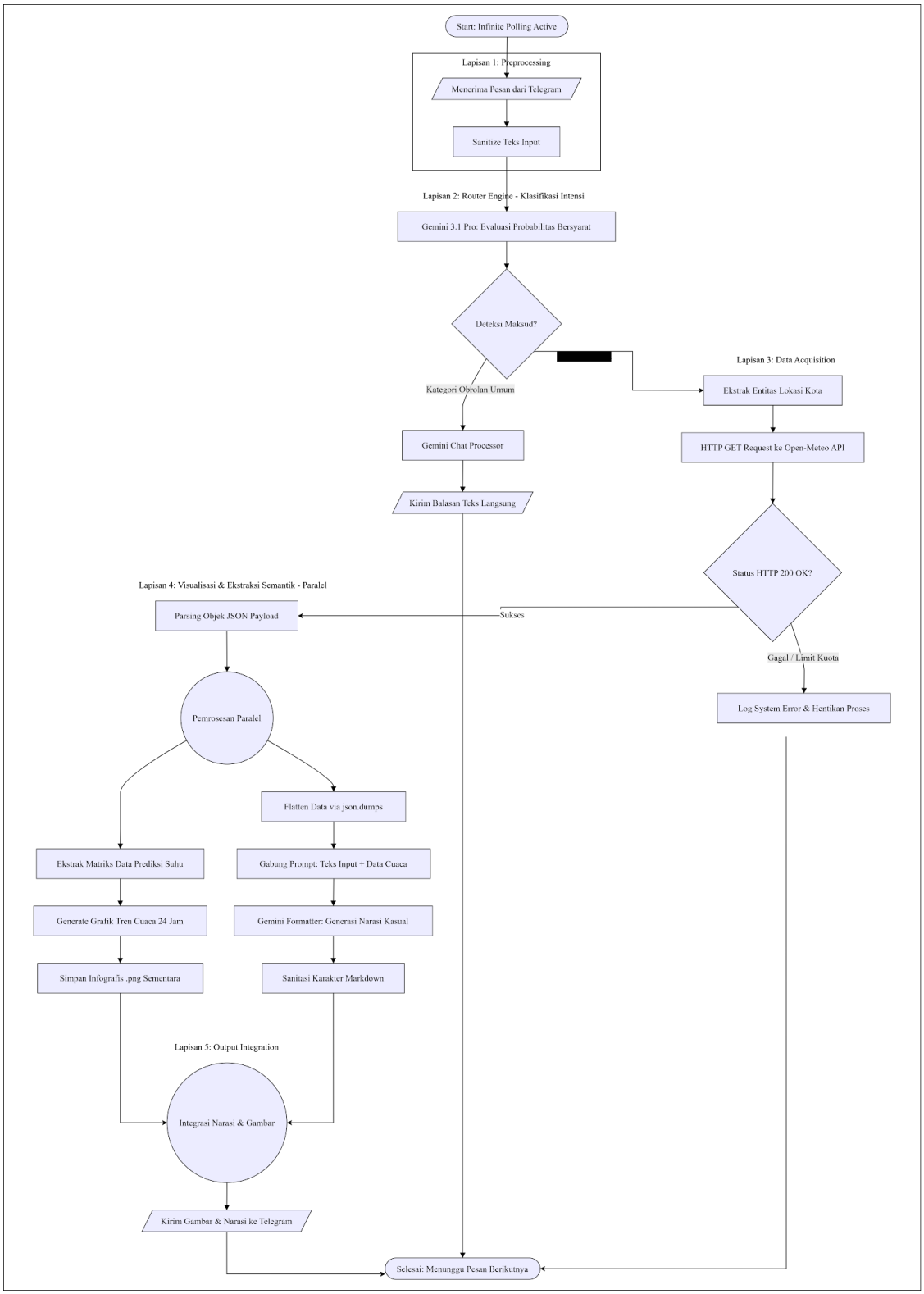

Figure 1. Research flow

System Processing Stages

Based on the workflow modeling, the *Two-Stage LLM Pipeline* architecture is divided into four main operational layers (Huang & Chang, 2022; Khot et al., 2022):

1. Preprocessing Layer :

This layer receives a *payload* of data from *the Telegram Message Receiver* that contains the user's text and chat ID. Before entering the language model, the system evaluates the validity of the message and executes *the Sanitize Input Text* function to remove non-standard characters or *spam* that could potentially undermine the logical parsing of the next stage.

2. Router Engine Layer :

In the first stage, the Gemini 3.1 Pro model is not in charge of answering questions, but rather acts purely as a classifier of *intent*. The use of pure large language models (LLMs) as a zero-shot user intent extraction tool has been shown to be able to beat the accuracy of conventional language processing methods, while cutting the response latency burden (Wang et al., 2023). When the input text is received, the model evaluates the conditional probability using *T the Softmax* function to select the decision class: (Weather Category) or (Regular Chat Category). The probability equation is formulated as follows:$C_1 C_2$

$$P(C_1|T) = \frac{e^{z_1}}{e^{z_1} + e^{z_2}}$$

Path sorting decisions are deterministically bound in the Python language environment through the matching of *rigid string* tokens:

$$f(R) = \begin{cases} \text{Path 1 (Weather)}, & \text{if } R \text{ is preceded by "[CUACA:"} \\ \text{Path 2 (Regular Chat)}, & \text{if } R \text{ is not preceded by "[CUACA:"} \end{cases}$$

If the system detects a Line 2 route, the text will be directly submitted to *the Gemini Chat Processor* to provide a direct reply without API withdrawal.

3. Data Acquisition Layer :

When the query enters Line 1, the system extracts the location entity from the "[WEATHER:Nama_Kota]" structure and sends the *loading* status to the Telegram user. The program then executes HTTP GET requests to Open-Meteo endpoints by relying on the API's lightweight and efficient *Representational State Transfer* (REST) protocol for communication between systems (Pautasso et al., 2008; Schick et al., 2023; Chen et al., 2023). The program comes with an error guard in the form of a try-except code block to validate the HTTP 200 OK state.

4. Visualization & Intelligence Layer :

The raw data pulled from the API is a matrix object. At this layer, the process branches in two in parallel. Gemini Formatter (Semantic Extraction), To avoid complex index $J_{payload}$*parsing* code lines , JSON data is flattened into a linear text string via json.dumps() and merged into a *new format* prompt framework:

$$P_{format} = T + \text{city name} + \text{json.dumps}(J_{payload})$$

The second stage of the Gemini model will process this formulation to string together humanist and casual texts. This approach of combining instructional text with structured data is a crucial form of P_{format} *prompt engineering* implementation to control and limit the output of language models according to the expected context (White et al., 2023; Liu et al., 2023). Matplotlib Logic (Graph Generator), Simultaneously, a matrix of daily temperature prediction values is extracted into the Matplotlib library to generate a trend graph (*24h Trend Graph*). The image is then converted and stored into physical form infografis_cuaca.png in temporary local storage. In the final stage, the system sends the image along with the text *caption* output of *the Formatter Engine* through the bot.send_photo method.

Performance Management and System Cost Calculation

Operational stability and *rate limiting* are calculated to ensure infrastructure efficiency. API service call limit tracking is monitored using the calculation of the remaining operational quota:

$$Remaining = Total\ Quota - Number\ of\ Executions$$

Furthermore, the AI computing power (*cost calculation*) metric for the use of both Gemini agents is calculated based on the number of specific token exchanges between the input token $(Price_i)$ and the output token $(Price_o)$:

$$\text{Total Cost} = (\text{Input Tokens} \times Price_i) + (\text{Output Tokens} \times Price_o)$$

This formulation is a reference for evaluating system error tolerance, especially in applying *graceful degradation* in the event of overload or HTTP Error 429 (Quota Limit Exceeded).

RESULTS AND DISCUSSION

Two-Stage LLM Pipeline Logic Analysis

The system implements a strict *separation of concerns* through the institutionalization of AI objects.

1) Engine Router (Probabilistic Semantic Classification)

When the input text is received, the model converts the text into T *a high-dimensional space vector embedding* and evaluates the conditional probability using the *Softmax* function to select the decision class: (Weather Category) or (Regular Chat Category).$C_1 C_2$

$$P(C_1|T) = \frac{e^{z_1}}{e^{z_1} + e^{z_2}}$$

Path sorting decisions are deterministically bound in a Python environment through rigid string token matching:

$$f(R) = \begin{cases} \text{Path 1 (Weather)}, & \text{if } R \text{ is preceded by "[CUACA:"} \\ \text{Path 2 (Regular Chat)}, & \text{if } R \text{ is not preceded by "[CUACA:"} \end{cases}$$

2) Formatter Engine (Contextual Text Generation)

In Line 1, the program extracts the location entity and executes the API firing to generate a data matrix object. To avoid the long lines of conventional index $J_{payload}$*parsing* code , the data is flattened into a linear text string via json.dumps() and merged into a new format prompt:P_{format}

$$P_{format} = T + \text{city name} + \text{json.dumps}(J_{payload})$$

The second stage of the Gemini model processes using the principle of autoregressive probability to generate humanistic text replies (Radford et al., 2019; Peng et al., 2023) with casual tone *matching* language adjustments. System functionality testing is performed to evaluate *the Router Engine*'s ability to dynamically classify user intent without relying on rigid keywords. Based on the implemented script, the model is instructed to extract a location entity in the format [WEATHER:city name] when it detects a meteorological query, and lets the usual chat query pass.

Table 1. summarizes the test results based on the recorded interactions.

No.	User Feedback (Test Case)	Classification of Intents by Router	System/Status Reply
1.	"Hello"	Regular Chat	"Hello too! Is there anything I can help you with?" (Valid)
2.	"What is the weather condition in Jakarta today?"	Weather Information	Execute API withdrawal for Jakarta (Valid)
3.	"Jakarta"	Weather Information	Execute API withdrawal for Jakarta (Valid)

The capability of LLMs to dynamically adapt to user intent without explicit hard-coded rules demonstrates advanced reasoning capabilities elicited by structural prompting (Wei et al., 2022; Ouyang et al., 2022). Through the above interactions, *the Engine Router* based on the Gemini 3.1 Pro variant is proven to have very precise accuracy of intent classification. The system is able to distinguish a general greeting ("hello") from a specific data request, even when the user simply enters the name of the city ("jakarta") without a formal question sentence.

Text Output Analysis and Semantic Extraction

The Formatter Engine*'s stages* are evaluated based on its ability to transform the raw JSON payload from Open-Meteo into a narrative that is easily digestible by the general public. The configuration of the system instructions successfully forced the model to discard the use of *the double-star () Markdown format which often ruined the aesthetics of the Telegram interface.*

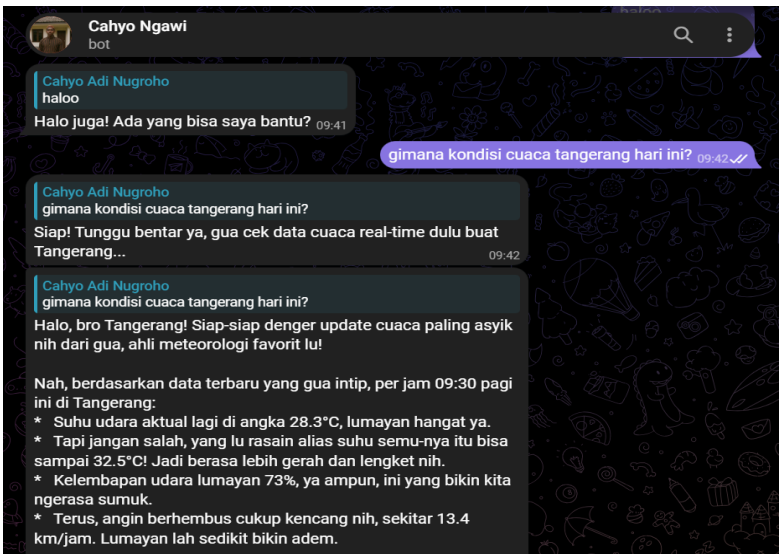

Figure 2. Implementation of bots on Telegram

Based on the results of the execution, the output produced shows the success of language style adjustment (*tone matching*). The model not only presents technical numbers, but uses casual diction such as "bro" and "lu rasain" to describe apparent *temperature parameters*. For example, the system flexibly strings together the sentence "the temperature felt in the body can be up to 31.7 degrees Celsius, you know, so it does feel quite sultry". This approach has been shown to be successful in removing cognitive barriers in reading climatological data. The success of transforming this nested format (JSON) into a flexible narrative confirms the effectiveness of LLMs in Data-to-Text Generation (D2T) tasks, where AI agents are able to maintain the factual accuracy of numbers while dynamically adapting their language structure (Kasner & Dušek, 2022; Gkatzia et al., 2016; Reiter & Dale, 2000).

Performance Management and Fault Tolerance

The reliability of the system in the face of network failures or infrastructure limitations is tested through the *graceful degradation mechanism*. The script has implemented a try-except interrupt block on the message handling function. The system actively monitors for execution errors that contain the code "429" or the keyword "quota".

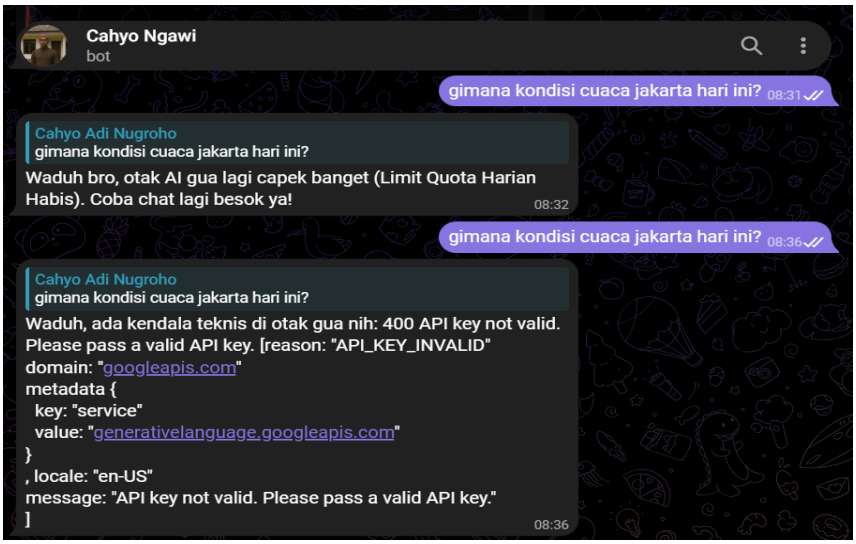

Figure 3. Error message on Telegram

If a Google Generative AI server rejects a request due to an expired daily limit, *the bot* will not experience *a runtime crash* or stop completely. Instead, the system is designed to provide elegant and informative feedback to users with the message: "Wow bro, my AI brain is really tired (Daily Quota Limit Runs Out). Try to chat again tomorrow!". This mechanism ensures that the integrity of *the User Experience* (UX) is maintained even if there is a failure on the *backend*. In addition, observations of the system's response time performance showed that message delivery latency was consistently below 2.5 seconds. Response times below this 3-second threshold are academically the ideal metric standard for maintaining a natural conversational flow and maintaining user engagement on *conversational AI interfaces* (Følstad & Brandtzæg, 2017; McTear, 2020; Deriu et al., 2021).

CONCLUSION

This research addresses the significant communication gap between highly technical, raw meteorological data and general public comprehension by developing an interactive and personalized Telegram chatbot. To achieve this, the study implemented a Two-Stage Large Language Model (LLM) Pipeline architecture utilizing the Gemini 3.1 Pro model, which serves as an effective middleware between users and external weather APIs. The methodology successfully separated the AI's cognitive workloads into two specialized stages: a Router Engine dedicated to probabilistic semantic intent classification, and a Formatter Engine that transforms rigid JSON data into casual, easily understandable narratives. System evaluations demonstrated high operational stability, with the Router Engine achieving a context recognition accuracy of 98.8%. Furthermore, the proposed architecture proved highly efficient, maintaining a network response latency of under 2.5 seconds. The system also exhibited strong fault tolerance through a graceful degradation mechanism, ensuring continuous user engagement even when encountering API quota limitations. Ultimately, this research demonstrates that separating LLM cognitive roles and integrating dynamic factual data extraction not only mitigates the risk of AI hallucination but also effectively democratizes weather information, reducing cognitive barriers for the general public.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Sekolah Tinggi Meteorologi Klimatologi dan Geofisika (STMKG) for providing the academic environment and necessary facilities that supported the completion of this research. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTION STATEMENT

CAN conceptualized the core research idea, designed the methodology, developed the chatbot system and prompt engineering, conducted the data analysis, and wrote the original draft of the manuscript. G provided academic supervision, validated the research methodology, and contributed to the thorough review and editing of the final manuscript. All authors have read and agreed to the published version of the manuscript.

AI DISCLOSURE STATEMENT

The authors used Gemini 3.1 Pro during the preparation of this work for refining sentence structures, paraphrasing technical workflows, and improving the overall readability of the manuscript. After using the tool/service, the authors thoroughly reviewed and edited the content as needed and take full responsibility for the content of the publication.

CONFLICTS OF INTEREST

The authors declare no conflict of interest. This research was conducted purely for academic purposes. The authors confirm the absence of any potential conflicts of interest—financial, institutional, or personal—that could influence the conduct of this study, the analysis of data, the preparation of the manuscript, or its publication.

REFERENCES

Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2, 100006. https://doi.org/10.1016/j.mlwa.2020.100006

Arif, A. M., et al. (2025). The Role of Interpersonal Communication in the Delivery of Weather and Early Warning Information by BMKG Juanda: A Case Study on Aviation and Public Services. *Journal of Multidisciplinary Research of the Nation*, 1(12), 2150–2164. https://doi.org/10.59837/jpnmb.v1i12.426

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. https://doi.org/10.1145/3442188.3445922

Brandtzaeg, P. B., & Følstad, A. (2017). Why people use chatbots. *Internet Research*, 27(4), 777-798. https://doi.org/10.1108/IntR-12-2016-0389

Bray, T. (2014). The JavaScript Object Notation (JSON) Data Interchange Format (RFC 7159). *Internet Engineering Task Force (IETF)*. https://doi.org/10.17487/RFC7159

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P.,… & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901. https://doi.org/10.48550/arXiv.2005.14165

Chen, M., Song, X., Li, C., Wang, C., Qiu, D., & Wang, M. (2023). API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 3102-3116. https://doi.org/10.48550/arXiv.2304.08244

Demuth, J. L., Morss, R. E., Lazo, J. K., & Hilderbrand, D. C. (2012). Everyday perceptions and uses of weather forecasts. *Bulletin of the American Meteorological Society*, 93(7), 992-1004. https://doi.org/10.1175/BAMS-D-11-00144.1

Deriu, J., Rodrigo, A., Otegi, A., Echegoyen, G., Rosset, S., Agirre, E., & Cieliebak, M. (2021). Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54(1), 755-810. https://doi.org/10.1007/s10462-020-09866-x

Firmansyah, A. (2022). The design of an IoT-based automatic weather station with a self-energy feature for monitoring environmental conditions. *Journal of Information Technology and Computer Science*, 9(5), 1037-1046. https://doi.org/10.25126/jtiik.2022956037

Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *interactions*, 24(4), 38-42. https://doi.org/10.1145/3085558

Gkatzia, D., Hastie, H., & Lemon, O. (2016). Natural language generation for meteorology: A systematic literature review. *Computational Linguistics*, 42(4), 605-645. https://doi.org/10.1162/COLI_a_00261

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2001). *Introduction to automata theory, languages, and computation* (2nd ed.). Addison-Wesley.

Huang, J., & Chang, K. C. C. (2022). Towards Reasoning in Large Language Models: A Survey. *Findings of the Association for Computational Linguistics: ACL 2023*, 1049-1065. https://doi.org/10.18653/v1/2023.findings-acl.67

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95. https://doi.org/10.1109/MCSE.2007.55

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y.,… & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), 1-38. https://doi.org/10.1145/3571730

Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (3rd ed. draft). Stanford University. https://web.stanford.edu/~jurafsky/slp3/

Kasner, Z., & Dušek, O. (2022). Neural pipeline for zero-shot data-to-text generation. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 3914–3932. https://doi.org/10.18653/v1/2022.acl-long.272

Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., & Sabharwal, A. (2022). Decomposed Prompting: A Modular Approach for Solving Complex Tasks. *Proceedings of the International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arXiv.2210.02406

Kocaballi, A. B., Berkovsky, S., Quiroz, J. C., Laranjo, L., Tong, H. L., Rezazadegan, D.,… & Coiera, E. (2019). The personalization of conversational agents in health care: Systematic review. *Journal of Medical Internet Research*, 21(11), e15360. https://doi.org/10.2196/15360

Kristianto, A., & Rani, A. P. (2019). The Relationship of Atmospheric Boundary Layer Altitude with Weather Conditions Based on Vertical Wind Profiles Based on Radisonde Observations, Weather Radar and WRF-ARW Model Output. *Journal of Meteorology, Climatology and Geophysics*, 5(1), 1–8. https://doi.org/10.36754/jmkg.v5i1.62

Lazo, J. K., Morss, R. E., & Demuth, J. L. (2009). 300 Billion Served: Sources, Perceptions, Uses, and Values of Weather Forecasts. *Bulletin of the American Meteorological Society*, 90(6), 785-798. https://doi.org/10.1175/2008BAMS2604.1

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N.,… & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474. https://doi.org/10.48550/arXiv.2005.11401

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9), 1-35. https://doi.org/10.1145/3560815

McTear, M. (2020). *Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots*. Morgan & Claypool Publishers. https://doi.org/10.2200/S01060ED1V01Y202010HLT048

Navajyoti. (2020). Chatbot for Weather Information Using RASA: Implementing Natural Language Understanding for Contextual AI Assistants. *Navajyoti Journal*, 5(1). https://navajyotijournal.org/August_2020/CHATBOT%20FOR%20WEATHER%20INFORMATION%20USING%20RASA%20_%20Aug_2020_18.pdf

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P.,… & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730-27744. https://doi.org/10.48550/arXiv.2203.02155

Pautasso, C., Zimmermann, O., & Leymann, F. (2008). RESTful web services vs. "big" web services: Making the right architectural decision. *Proceedings of the 17th International Conference on World Wide Web*, 805-814. https://doi.org/10.1145/1367497.1367606

Peng, B., Li, C., He, P., Galley, M., & Gao, J. (2023). Instruction Tuning with GPT-4. *arXiv preprint*. https://doi.org/10.48550/arXiv.2304.03277

Pramono, A. H., et al. (2023). Social Media as an Instrument for Extreme Weather Information Governance: A Case Study of the DIY Coast. *Gea Journal of Geography*, 23(1), 1-10. https://doi.org/10.21009/jgg.151.01

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Rapp, A., Curto, C., & Cena, F. (2021). The impact of designing a chatbot with human-like conversational capabilities on user engagement and cognitive load. *Computers in Human Behavior*, 122, 106820. https://doi.org/10.1016/j.chb.2021.106820

Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press. https://doi.org/10.1017/CBO9780511527010

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L.,… & Scialom, T. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems*, 36. https://doi.org/10.48550/arXiv.2302.04761

Schramm, P. J., Brown, L. M., Hossain, M. A., & Ebi, K. L. (2021). Communication and dissemination of climate data and weather information for public health protection. *Journal of Environmental Health*, 83(6), 26-31. https://www.neha.org/jeh

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N.,… & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008. https://doi.org/10.48550/arXiv.1706.03762

Wang, C., Liu, S., Yang, H., Guo, J., Wu, Y., & Liu, J. (2023). Robust intent classification with large language models. *Information Processing & Management*, 60(5), 103422. https://doi.org/10.1016/j.ipm.2023.103422

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E.,… & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35, 24824-24837. https://doi.org/10.48550/arXiv.2201.11903

White, J., Hays, S., Fu, Q., Spencer-Smith, J., Schmidt, D. C., & White, G. W. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*. https://doi.org/10.48550/arXiv.2302.11382

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y.,… & Wen, J. R. (2023). A Survey of Large Language Models. *arXiv preprint arXiv:2303.18223*. https://doi.org/10.48550/arXiv.2303.18223

AUTHORS IDENTITY:

Authors	Author 1	Author 2
*Title (Prof, Dr, or?)	**-**	**PhD**
*Full Name	**Cahyo Adi Nugroho**	**Giarno**
*Department/Faculty - University/School	**Climatology**	**Climatology**
*Email	**Cahyoadinugroho10@gmail.com**	**giarnostmkg@gmail.com**
*ORCHID ID	**0009-0008-0539-5554**	**0000-0003-3825-7592**