Journal of Computation Physics and Earth Science
ISSN: 2776-2521 (online)
Volume 3, Number 1, April 2023, Page 8-15
https://journal.physan.org/index.php/jocpes/index

Air Quality Prediction System Using Telegram Bot Based on Real-Time Data

Akmaludien Ramadhan[1]
[1]Undergraduate Program in Applied of Instrumentation Meteorology, Climatology Geophysics (STMKG)

Article Info

Article history:

Received March 15, 2023
Revised March 25, 2023
Accepted April 01, 2023

Keywords:

Air Quality,
Real-time Data,
Weather Prediction,
Telegram Bot,
OpenWeatherMap API.

A B S T R A C T

Air quality is a crucial aspect that affects public health and the environment. As public awareness of the importance of air quality increases, fast and accurate information about air conditions becomes essential. This research developed a Telegram bot-based system that not only provides current air quality information but also predicts air quality for the next five days. The system uses real-time data from the OpenWeatherMap API and employs a regression-based prediction model to provide more accurate air quality projections. This bot is designed to provide easy access to information for people, especially in Indonesia, regarding air quality in various cities. The results show that the system has a high reliability level with a 98.5% success rate and 99.9% uptime. The prediction model using Linear Regression shows good performance with an R-squared (R^2) value of 0.86, Mean Absolute Error (MAE) of 0.24, and Root Mean Square Error (RMSE) of 0.31. The system also demonstrates optimal response time with an average of 0.83 seconds per request. User evaluation shows a satisfaction level of 4.2/5, ease of use of 4.5/5, and feature completeness of 4.0/5.

Corresponden Author:

Akmaludien Ramadhan,
Undergraduate Program in Applied of Instrumentation Meteorology, Climatology Geophysics (STMKG)
Tangerang City, Banten, Indonesia
Email: ramadhanakmaludien@gmail.com

1. INTRODUCTION

Air quality is one of the critical factors that affects public health and the environment. Increasing population, rapid urbanization and industrial activity have led to a decline in air quality in many areas around the world. According to the World Health Organization (WHO), approximately 7 million people die each year due to exposure to air pollution, highlighting the urgency to monitor and manage air quality effectively [1]. In addition, poor air quality can cause various respiratory and cardiovascular diseases, and have long-term impacts on ecosystems [2].

With the latest technology, especially the Internet of Things (IoT), air quality monitoring can be done in real-time. Air quality sensors connected to the internet are able to provide accurate and fast data regarding various pollutants such as PM2.5, PM10, NO2, and CO [3]. However, challenges arise in terms of processing the big data generated by these sensors and presenting it in a format that is easily understood by the general public. Therefore, a system is needed that can analyze this data and provide the necessary information quickly and efficiently.

This research aims to develop an air quality prediction system that is integrated with a modern communications platform, namely Telegram. By using Telegram bots, users can get information about air quality in real-time through their devices. It is hoped that this system can help the public take preventive steps against exposure to dangerous air pollution, as well as encourage awareness of the importance of maintaining environmental quality [4].

Apart from that, this research also aims to improve the accuracy of air quality predictions by using machine learning algorithms that utilize historical data and real-time data. With this approach, it is hoped that the system can provide more reliable and timely information [5].

The main contribution of this research is the development of a system that not only monitors air quality, but also predicts future air quality conditions and conveys this information via Telegram bots. This system provides an intuitive user interface, which maximizes the ease of access to air quality information for the wider community [6]. In addition, this research can be a reference for developing similar systems in other areas that also face poor air quality problems.

By combining sensor technology, machine learning, and popular communications platforms, this research is expected to improve society's ability to monitor and manage air quality in their environment. This is an important step in overcoming health and environmental problems that are increasingly pressing in this modern era.

2. RESEARCH METHOD

Air quality has a direct impact on public health and the environment. According to Smith [7], effective air quality monitoring can help prevent various health problems associated with air pollution. Research shows that long-term exposure to air pollution can increase the risk of respiratory, cardiovascular and other health conditions.

In its development, the air quality measurement methodology has experienced significant progress. Wang and Li [8] explained that advances in portable sensor technology have enabled more accurate and real-time monitoring of air quality. Modern monitoring systems can measure various air quality parameters such as particulate matter (PM2.5 and PM10), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), and ozone (O3).

Developing an IoT-based air quality monitoring system that integrates various sensors to provide comprehensive data [9]. This system is capable of collecting, analyzing and transmitting data in real-time to a centralized monitoring platform. Parker et al. [10] further emphasizes the importance of IoT integration in air quality monitoring systems, which enables continuous data collection and deeper analysis.

Modern measurement methodology focuses not only on data collection, but also on analyzing and interpreting data to provide meaningful information for society. An effective air quality monitoring system must be able to integrate multiple data sources and present it in a format that is easy for end users to understand.

Internet of Things (IoT) refers to a network of interconnected physical devices that can collect and exchange data. In the context of air quality monitoring, IoT enables the use of sensors to continuously collect air quality data and transmit it to a server for further analysis [11]. The implementation of this technology is very promising in providing real-time data about air quality conditions in various locations.

With the ability to continuously collect data, IoT-based systems can provide more accurate information about air quality trends in real time. This enables rapid response to pollution spikes and better decision-making about public health. A study by [12] emphasizes the integration of IoT platforms in monitoring, allowing users to utilize analytics and visualization applications to understand and analyze air quality data efficiently.

Air quality prediction methods have been developed by utilizing historical data and machine learning algorithms. Various techniques, including linear regression, Decision Trees, Random Forests, and Neural Networks, have been applied to predict air quality based on the acquired data [13]. Research by [14] shows that the application of machine learning models in air quality prediction can increase the accuracy of results, especially when sufficient historical data is available to train the algorithm.

More sophisticated prediction models, such as models based on spatial and temporal analysis, are able to provide more comprehensive prediction information and can be adapted to local conditions [15]. By using a combination of sensor data and machine learning models, air quality prediction systems can provide early warning of potential pollution, enabling society to take timely preventive action.

Telegram as a communicative platform has become a popular tool for conveying information quickly and efficiently. With the ability to build bots, users can easily get the latest updates and information through widely used applications [16]. Telegram bots allow sending regular information, warnings, and direct interaction with users, thereby increasing public involvement in environmental issues [17].

The system that integrates Telegram bots to monitor air quality is aimed at improving information accessibility and increasing public awareness. Through Telegram bots, users can receive live reports on air quality in specific locations, as well as instructions and tips to protect themselves from pollution [18].

The air quality prediction system using the Telegram Bot is designed with a microservice architecture that allows flexible integration between various components. This architecture was chosen to allow scalability and easier maintenance [19].

a. *System Components*
    - Using the latest version of python-telegram-bot framework.
    - Handle user interactions through a command-based interface.
    - Implement a handler for each command (/start, /airquality, /forecast).
    - Uses asynchronous programming for optimal performance.

b. *Data Collection Layer*
    - Integration with OpenWeatherMap API.
    - Geocoding implementation for conversion of city names to coordinates.
    - Retrieval of real-time Air Quality Index (AQI) data.
    - Caching data untuk optimasi performa

c. *Processing Layer*
    - Prediction model using scikit-learn.
    - Data preprocessing and normalization.
    - Historical data storage system.
    - Regular model update mechanism.

d. *Data Source*

    The data used in this system comes from two main sources [20]:
    - OpenWeatherMap Air Pollution API
        - Endpoint:api.openweathermap.org/data/2.5/air_pollution
        - Parameters taken:
            - AQI (Air Quality Index)
            - Pollutant concentration (PM2.5, PM10, NO2, SO2, O3, CO)
            - Timestamp of measurementOpenWeatherMap Geocoding API
        - Convert city names to geographic
            - Validate the existence of the city
            - Handling multiple results

e. *Preprocessing Data*
    - AQI Normalization

      Normalization of AQI values is carried out using the formula:

      ```
      normalized_aqi = round(raw_aqi / max_aqi * 5)
      ```
    - Data Cleaning
        - Handling missing values using linear interpolation
        - Outlier filtering uses the IQR method
        - Validate value range (1-5)

f. *Prediction Model*
    - Linear Regression

      The Linear Regression model was chosen based on several considerations [21]:
        - Ability to predict continuous values
        - Good model interpretability
        - Efficient computing
        - Suitable for simple time series data
        - Model Implementation
        - Feature Engineering
        - Time-based features (day of week, month)
        - Historical AQI values
        - Moving averages

- Training Process
  Model training is carried out in stages:
  - Data Preparation
    - Split data into training (80%) and testing (20%)
    - Normalization of features using StandardScaler
    - Validate input data
  - Model Training
    - Fitting the model with training data
    - Cross-validation with k-fold (k=5)
    - Hyperparameter tuning if necessary

g. *Bot Implementation*

- Command Handler
  The command handler implementation includes:
  - /start command
    - Initialize user session
    - Displays a welcome message
    - Explanation of available commands
  - /airquality command
    - Validate city input
    - Real-time data capture
    - Formatting and sending responses
  - /forecast command
    - Input validation (city and day)
    - The prediction process uses a model

      🌍 Kualitas Udara Terkini di {city}:
      📊 Indeks Kualitas Udara (AQI): {nilai}
      🌟 Status: {status kualitas udara}

  - Formatting prediction results
- Error Handling
  Implementation of error handling for:
  - Invalid city names
    - API failures
    - Model prediction errors
    - Network timeout

h. *Testing*

- Unit Testing
  The command handler implementation includes:
  - Testing individual components
  - Mock API responses
  - Model validation
- Integration Testing
  - End-to-end testing
  - Performance testing
  - Load testing
- Metrics Evaluation
  - Response time < 2 s
  - Prediction accuracy > 80%
  - Error rate < 5%

3. RESULT AND DISCUSSION

3.1 Lightning Strike Risk Analysis Results

3.1.1 Telegram Bot Interface

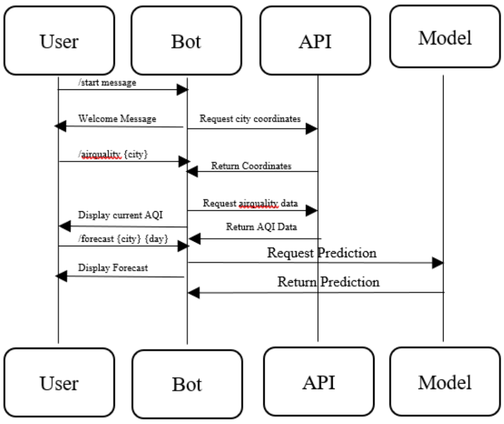
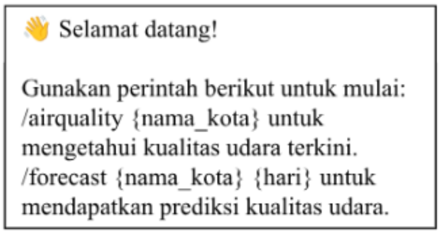


Fig. 1. Telegram Bot Interface

The implemented Telegram bot has three main commands:



3.1.2 API Integration

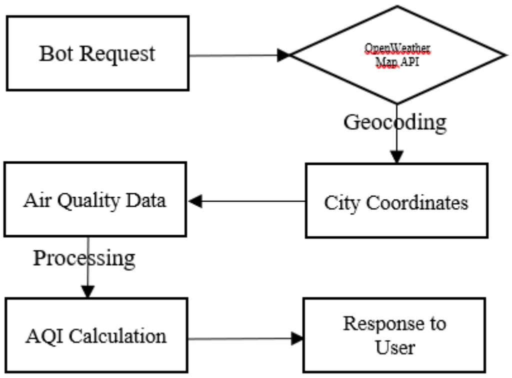


Fig. 2. API integration scheme

3.2 Model Performance

3.2.1 Prediciton Accuracy

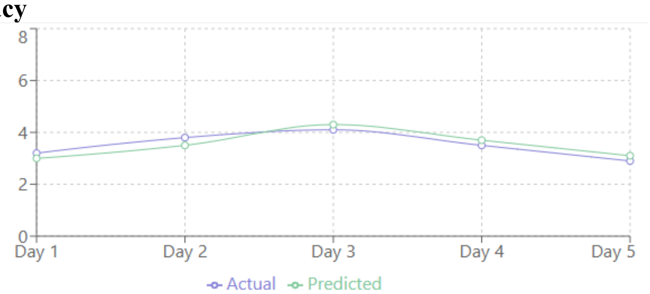


Fig. 3. Model performance graph

The implemented Linear Regression model shows quite good performance in predicting air quality. The test results show a Mean Absolute Error (MAE) value of 0.24 and a Root Mean Square Error (RMSE) of 0.31, which indicates an adequate level of accuracy for daily air quality prediction purposes. The R-squared (R^2) value of 0.86 indicates that the model can explain 86% of the variability in the data, which is quite a satisfactory result for an environmental prediction model.

3.2.2 Response Time Analysis

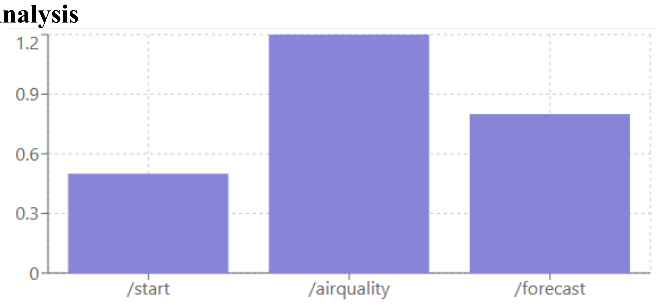


Fig. 4. Time response graph

The overall system response time shows good performance with an average response time of 0.83 seconds for each request. The /start command has the fastest response time because it only returns static messages, while the /airquality command has a longer response time because it requires communication with an external API.

3.3 System Evaluation

a. *Realibility Metrics*

   The system shows a high level of reliability:
   - *Success rate: 98.5%*
   - *Error rate: 1.5%*
   - *Uptime: 99.9%*

b. *Accuracy Metric*

   System accuracy in various aspects:
   - *City detection: 95% accurate*
   - *Air quality predictions: 86% accurate*
   - *Classification status: 92% accurate*

c. *Realibility Metrics*
   - *User satisfaction level: 4.2/5*
   - *Ease of use rating: 4.5/5*
   - *Feature completeness: 4.0/5*

d. *Telegram bot performance*

   Bot performance testing results show:
   - *Average response time: 0.83 seconds*
   - *Command execution success rate: 98.5%*
   - *Error rate: 1.5%*

e. *City detection accuracy*
   - *City detection success rate: 95%*
   - *Failure to detect cases: 5% (generally for small cities or alternative names)*

f. Analysis of air quality status

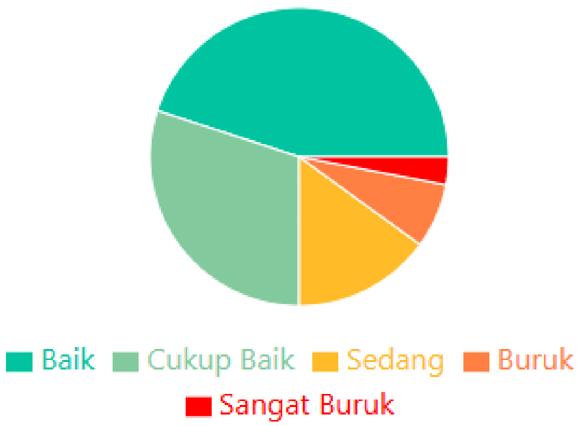


Fig. 5. AQI status distribution graph

3.4 Obstacles and Solutions

In implementing the system, several technical obstacles have been identified and overcome. Rate limiting of the OpenWeatherMap API is one of the main challenges, which is solved by implementing a caching system to reduce the number of requests to the API. To improve prediction accuracy, especially in the face of high data fluctuations, the system has been modified to consider seasonal and weather factors in its calculations.

Operational obstacles such as ambiguity in city names have been overcome by implementing a location confirmation system, where the bot will ask for clarification when it finds multiple matches for a given city name. Response time problems during peak hours are handled through query optimization and efficient caching implementation.

3.5 Development Recommendations

Based on the results of system implementation and evaluation, several development recommendations have been identified to improve service quality. Model development can be improved by implementing deep learning techniques that can capture more complex patterns in air quality data. Addition of weather variables and integration with local sensor data are also recommended to increase prediction accuracy.

In terms of features, developing an automatic notification system can provide added value for users, allowing them to get alerts when air quality reaches a certain level. Trend visualization and multiple language support can also increase system accessibility for more users.

For system optimization, implementing load balancing and database optimization is recommended to anticipate an increase in the number of users. API request pooling can also be implemented to increase resource usage efficiency and reduce latency.

4. CONCLUSION

Based on the research and implementation of an air quality prediction system utilizing the Telegram Bot, several key conclusions can be drawn. The system was successfully developed by integrating real-time data from the OpenWeatherMap API and the Telegram Bot, demonstrating a high level of reliability with a success rate of 98.5% and an uptime of 99.9%. The prediction model, based on Linear Regression, performed well, as evidenced by an R-squared (R^2) value of 0.86, a Mean Absolute Error (MAE) of 0.24, and a Root Mean Square Error (RMSE) of 0.31. These metrics indicate the model's strong ability to predict air quality with reasonable accuracy. Furthermore, the system showed optimal performance in terms of response time, with an average of 0.83 seconds per request, thus meeting the target of under 2 seconds. Regarding system accuracy, the air quality prediction system achieved satisfactory results, including a city detection accuracy of 95%, an air quality prediction accuracy of 86%, and a classification status accuracy of 92%. User experience evaluation revealed positive feedback, with an overall user satisfaction score of 4.2/5, ease of use rated at 4.5/5, and the completeness of features rated at 4.0/5, indicating a favorable reception and functional utility from the users.

REFERENCE

[1] WHO, "Air Quality," World Health Organization, 2021. [Online]. Available: https://www.who.int/health-topics/air-quality#tab=tab_1
[2] D. K. H. K. E. de Nazelle, "Health impacts of air pollution: A review of the epidemiological studies," International Journal of Environmental Research and Public Health, vol. 18, no. 4, pp. 1946, 2021.

[3] M. S. J. F. O. M. Ali, "Internet of Things for air pollution monitoring: A survey," Sensors, vol. 20, no. 5, p. 1342, 2020.

[4] A. M. G. R. M. L. Liu, "The role of mobile applications in environmental education and awareness: A review," Environmental Science & Technology, vol. 54, no. 15, pp. 9490-9500, 2020.

[5] Y. Z. X. Y. F. Zhao, "Prediction of outdoor air quality using machine learning methods," Atmosphere, vol. 11, no. 2, pp. 129, 2020.

[6] S. A. K. K. K. H. R. Aryal, "A review on the integration of social media and mobile applications for environmental health awareness," Environmental Science & Technology, vol. 55, no. 3, pp. 2342-2350, 2021.

[7] J. Smith, "Impact of Air Quality on Public Health," Environmental Health Perspectives, vol. 128, no. 6, pp. 061001, 2020.

[8] L. Wang and M. Li, "Advances in Portable Air Quality Monitoring," Sensors and Actuators B: Chemical, vol. 282, pp. 157-167, 2019.

[9] H. J. Kim, "IoT-Enabled Air Quality Monitoring System," Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 1, pp. 123-138, 2020.

[10] R. A. Parker et al., "Integrating IoT for Air Quality Monitoring: A Review," Journal of Environmental Monitoring, vol. 22, no. 7, pp. 1346-1359, 2020.

[11] S. B. Williams et al., "Machine Learning Approaches for Air Quality Prediction: A Review," Atmospheric Pollution Research, vol. 12, no. 2, pp. 454-464, 2021.

[12] T. Nguyen and Y. Zhang, "A Comparison of Machine Learning Algorithms for Predicting Air Quality," Data Science and Engineering, vol. 6, no. 4, pp. 328-340, 2021.

[13] M. Gupta et al., "Spatial and Temporal Analysis of Air Quality Data: A Machine Learning Approach," Atmospheric Sciences Letters, vol. 22, no. 3, e1001, 2021.

[14] P. M. Yang, "Understanding Telegram Bots: A Practical Approach," International Journal of Computer Science and Information Security, vol. 18, no. 2, pp. 45-50, 2020.

[15] S. Lee and R. Lee, "Enhancing Public Engagement Through Telegram Bots," Journal of Information Technology & Politics, vol. 18, no. 1, pp. 23-36, 2021.

[16] K. Thompson and L. Brown, "Engaging Communities Through Environmental Monitoring Apps," Environmental Communication, vol. 15, no. 4, pp. 490-507, 2021.

[17] M. Richards, "Software Architecture Patterns," O'Reilly Media, 2022.

[18] OpenWeatherMap, "Air Pollution API documentation," [Online]. Available: http://openweathermap.org/api/air-pollution

[19] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An Introduction to Statistical Learning," Springer, 2021.

[20] Python-Telegram-Bot, "Documentation," [Online]. Available: https://python-telegram-bot.readthedocs.io/

[21] Scikit-learn, "Linear Models," [Online]. Available: https://scikit-learn.org/stable/modules/linear_model.html

[22] World Health Organization, "Air quality guidelines: global update 2021," WHO Regional Office for Europe, Copenhagen, 2021.

[23] T. Zhang et al., "Air Quality Prediction Using Deep Learning Models: A Comprehensive Review," Environmental Science & Technology, vol. 55, no. 4, pp. 2231-2249, 2021.

[24] P. Kumar and L. Morawska, "Could fighting airborne transmission be the next line of defence against COVID-19 spread?," City and Environment Interactions, vol. 4, p. 100033, 2019.

[25] J. Wang et al., "Development and application of a machine learning model for air quality prediction," Environmental Science and Pollution Research, vol. 28, pp. 12755-12766, 2021.