TAMIKA: Jurnal Tugas Akhir Manajemen Informatika & Komputerisasi Akuntansi
Vol. 5 No. 1 (Juni 2025)
https://doi.org/10.46880/tamika.Vol5No1.pp110-115 e-ISSN: 2828-1276

USING AI IN TELEGRAM BOTS FOR EARTHQUAKE INFORMATION DELIVERY AND ANALYSIS

Ayu Adi Justicea✉, Rala Kurniawan Agustin, Suko, Darwin Harahap, Marzuki Sinambela
Undergraduate Program in Applied of Instrumentation Meteorology, Climatology and Geophysic
STMKG, Tangerang, Indonesia
Email: ayuadi.justicea@stmkg.ac.id

ABSTRACT

Indonesia is located in the Pacific Ring of Fire, making it vulnerable to earthquake activity. Earthquake events often require fast and accurate information delivery to reduce the impact of disasters. However, conventional systems often face obstacles such as information delays and lack of real-time impact analysis. Advances in artificial intelligence (AI) and machine learning (ML) provide opportunities to overcome these problems. AI is able to process seismic data efficiently, detect earthquake patterns, and predict their impact. By utilizing Telegram bots as a medium, earthquake information can be delivered quickly through real-time notifications, which are easily accessible to the public. The development of AI-based Telegram bots aims to provide earthquake information automatically, including location, magnitude, and potential impact, as well as analyze historical data to predict seismic activity trends. This solution is expected to increase public awareness and response in dealing with earthquake disasters..

Keyword: AI, Machine Learning, Telegram, Earthquake.

INTRODUCTION

Indonesia is one of the countries that has a high risk of earthquakes due to its location in the Pacific Ring of Fire zone. This geological condition causes Indonesia to experience thousands of earthquakes every year, both on a small and large scale, which can have a serious impact on people's lives. According to data from the Meteorology, Climatology and Geophysics Agency (BMKG), more than 11,000 earthquakes were recorded throughout 2023, most of which were not felt by the public, but still had the potential to damage infrastructure in the affected areas (BMKG, 2023).

In such situations, the speed and accuracy of information is of key importance to improving community preparedness for earthquakes. Slow or inaccurate information delivery can increase the risk of loss of life and materials. Information technology offers practical solutions to overcome this challenge, especially through internet-based applications that are easily accessible to the wider community.

One innovation in information technology is the development of Telegram Bot, an automation program based on the Telegram platform. Telegram bots can be used to deliver real-time earthquake information, allowing users to get instant notifications directly to their devices. When coupled with Artificial Intelligence (AI) capabilities, these bots can become smarter by analyzing earthquake data, predicting potential impacts, and providing relevant recommendations for disaster mitigation.

The application of AI in disaster mitigation has been proven effective by various studies. Xu et al. (2020) revealed that AI has the ability to process seismic and geospatial data quickly, making it easier to identify patterns of earthquake risk and impact. In addition, the use of AI in earthquake early warning systems is growing, as described by Li and Zhang (2018) who explain how AI is used in earthquake warning systems to provide faster responses. In recent years, many studies have utilized machine learning to predict earthquakes and their impacts. Güney et al. (2019) in their review describe various machine learning techniques used to improve earthquake prediction accuracy. Meanwhile, Zhang and Wang (2019) showed how the integration of deep learning can improve the classification of seismic events, which is crucial for improving the accuracy of earthquake information provided to the public.

The use of Telegram bots in the delivery of earthquake information is an innovation that allows for easier and faster access. The Telegram Bot API, which allows the development of bots to automatically provide information, has become an effective tool to increase public awareness of potential disasters. Telegram bots can send real-time earthquake notifications, allowing people to immediately know the location, magnitude, and impact, so they can take necessary actions. For example, Chatterjee et al. (2021) conducted a case study on the use of chatbots in

improving awareness and response to disasters, including earthquakes.

Barua and Kumar (2020) proposed the use of machine learning to predict tsunami risks following earthquakes, which could aid in better disaster response planning. Carvalho et al. (2021) added that the application of machine learning in tsunami warning systems provides the advantage of quicker response times. This suggests that the use of AI in disaster mitigation could be more holistic, encompassing tsunami prediction after an earthquake.

Saini and Soni (2020) provide a further review of the application of AI in disaster management, highlighting the importance of efficient and rapid data processing to reduce risk. They also noted that AI technology will be key in improving disaster response in the future.

RESEARCH METHODS

Artificial Intelligence

Artificial Intelligence (AI) is a branch of computer science that focuses on developing systems that can perform tasks that would normally require human intelligence. These tasks include pattern recognition, decision-making, and prediction based on data. According to Russell and Norvig (2020), AI is the ability of a system to think, learn, and act like a human by utilizing sophisticated algorithms and big data.

In the context of disaster mitigation, AI plays an important role, especially in analyzing complex seismic data and providing results in a short time. For example, AI systems can process earthquake data to identify the epicenter location and estimate the impact. This technology helps disseminate information faster and more accurately than manual methods (Xu et al., 2020).

AI also has several branches, such as machine learning, deep learning, and natural language processing (NLP), which can be integrated with Telegram bots to improve their analysis and information delivery capabilities. For example, NLP allows bots to respond to user questions in a more natural and informative way (Goodfellow et al., 2016).

Use of Telegram Bots in Information Delivery

Telegram is a cloud-based instant messaging application that provides bot development features through the Telegram Bot API. Telegram bots allow users to interact with automated systems designed to perform specific tasks, such as providing information, sending notifications, and presenting data visualizations. In the context of delivering earthquake information, Telegram bots have advantages in terms of speed, accessibility and personalization capabilities. By using bots, up-to-date earthquake information such as magnitude, location and depth can be delivered directly to users in real time. In addition, bots can include additional features such as location maps and earthquake trend graphs to make it easier for users to understand.

The application of AI in disaster management is expanding, with Santos et al. (2020) showing how AI-powered chatbots can change the way emergency response is conducted, improving coordination between authorities and communities. Bakhshi et al. (2021) further emphasized the role of Telegram Bots in real-time earthquake communication, mentioning that the integration of Telegram with AI systems can improve the speed and accuracy of information delivery. A study by Chatterjee et al. (2021) showed that the use of bots for disaster information dissemination increased public awareness by 40%. The real-time notification feature of Telegram bots allows people to get information immediately, so they can take the necessary steps quickly. The integration of AI in Telegram bots also enables predictive analysis to provide early warnings to users based on historical patterns of earthquakes.

Python Script for Telegram Bot.

The script contains the main functions in the Python script used to build an AI-based Telegram bot to provide earthquake information. This bot uses data from the BMKG API to deliver the latest earthquake information, location maps, tsunami predictions, and mitigation recommendations to users.

1. Get Earthquake Data
   Retrieves the latest earthquake data from the BMKG API. This function uses the `requests` library to access the BMKG API URL and returns earthquake information in JSON, or `None` if an error occurs.

```python
def get_earthquake_data():
    try:
        response = requests.get(BMKG_API_URL)
        if response.status_code == 200:
            data = response.json()
            return data['Infogempa']['gempa']
        return None
    except Exception as e:
        print(f"Error: {e}")
        return None
```

Fig. 1. Function to Get the Latest Earthquake Data

2. Create Map
   Create an earthquake location map using the `folium` library. This function displays earthquake

location markers based on latitude and longitude coordinates, and saves the map in the file `gempa_map.html`.

```
def create_map(lat, lon, lokasi):
    peta = folium.Map(location=[lat, lon], zoom_start=8)
    folium.Marker([lat, lon], popup=f"Lokasi: {lokasi}").add_to(peta)
    map_path = "gempa_map.html"
    peta.save(map_path)
    return map_path
```

Fig. 2. Function to Create Earthquake Map

3. Recommendation Mitigation
   Provides mitigation recommendations based on the magnitude and depth of the earthquake. Recommendations are grouped into four categories based on the magnitude scale.

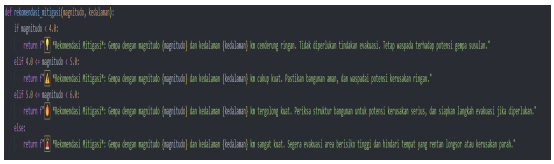

Fig. 3. Function to Create Earthquake Map

4. Clean Depth
   Cleans the depth string from the API and converts it to a float. This function is useful to ensure the depth data can be used in calculations or predictions.

```
def clean_depth(depth_str):
    """Membersihkan string kedalaman menjadi float."""
    try:
        return float(depth_str.replace(" km", "").strip())
    except ValueError:
        return None
```

Fig. 4. Function to Clean Depth String to Float

5. Train Tsunami Model
   Train a Logistic Regression model for potential tsunami prediction. Training data includes magnitude, depth and tsunami label. The model is stored in the file `tsunami_predictor.pkl`.

```
def train_tsunami_model():
    data = [
        [5.5, 10, 1], [6.0, 30, 1], [4.5, 50, 0], [5.8, 15, 1], [7.0, 100, 1],
        [3.5, 60, 0], [5.2, 20, 0], [6.5, 70, 1], [4.0, 40, 0], [6.8, 25, 1]
    ]

    X = [d[:2] for d in data]
    y = [d[2] for d in data]

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

    model = LogisticRegression()
    model.fit(X_train, y_train)

    with open('tsunami_predictor.pkl', 'wb') as file:
        pickle.dump(model, file)

    return model
```

Fig. 5. Function to Train the Tsunami Prediction Model

6. Load Tsunami Model
   Loads the tsunami prediction model from the file `tsunami_predictor.pkl`. If the file is not found, this function trains a new model.

```
def load_tsunami_model():
    try:
        with open('tsunami_predictor.pkl', 'rb') as file:
            model = pickle.load(file)
        return model
    except FileNotFoundError:
        print("Model tidak ditemukan, melatih model baru...")
        return train_tsunami_model()
```

Fig. 6. Function to Load the Tsunami Prediction Model

7. Predict_tsunami
   Uses the prediction model to determine the tsunami potential based on earthquake magnitude and depth. Returns 1 if a potential tsunami exists, and 0 otherwise.

```
def predict_tsunami(magnitudo, kedalaman, model):
    scaler = StandardScaler()
    input_data = scaler.fit_transform([[magnitudo, kedalaman]])
    prediction = model.predict(input_data)
    return prediction[0]
```

Fig. 7. Function to Predict the Tsunami

8. Start Command
   Greets the user when the bot is run for the first time. This function provides a list of commands that can be used by the bot.

```
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text("👋 *Halo! Saya adalah bot informasi gempa.*\n\n"
                                    "Gunakan perintah berikut untuk mendapatkan informasi gempa terkini:\n"
                                    "📍 /gempa - Menampilkan info gempa terkini 📊\n"
                                    "🗺️ /peta - Menampilkan peta lokasi gempa terbaru 🗺️\n"
                                    "⚠️ /mitigasi - Rekomendasi mitigasi gempa 🛠️\n"
                                    "🌊 /shakemap - Menampilkan gambar shakemap (peta guncangan) 🌎\n"
                                    "🌊 /tsunami - Prediksi potensi tsunami berdasarkan gempa 🌊")
```

Fig. 8. Command to Start the Bot

9. Earthquake Command
   Retrieves the latest earthquake data from the BMKG API and displays information such as location, magnitude, depth, time, tsunami potential, and whether the earthquake was felt by the public.

```
async def gempa(update: Update, context: ContextTypes.DEFAULT_TYPE):
    data = get_earthquake_data()
    if data:
        wilayah = data.get('Wilayah', 'Data tidak tersedia')
        magnitudo = data.get('Magnitude', 'Data tidak tersedia')
        kedalaman = data.get('Kedalaman', 'Data tidak tersedia')
        waktu = data.get('DateTime', 'Data tidak tersedia')
        potensi_tsunami = data.get('PotensiTsunami', 'Tidak ada potensi tsunami')
        dirasakan = data.get('Dirasakan', 'Tidak ada data dirasakan')

        pesan = (
            f"🌍 *Informasi Gempa Terkini*\n"
            f"📍 *Lokasi*: {wilayah}\n"
            f"🌐 *Magnitudo*: {magnitudo} Mw\n"
            f"🔴 *Kedalaman*: {kedalaman}\n"
            f"🕒 *Waktu*: {waktu}\n"
            f"⚠️ *Potensi Tsunami*: {potensi_tsunami}\n"
            f"✳️ *Dirasakan*: {dirasakan}"
        )
        await update.message.reply_text(pesan, parse_mode="Markdown")
    else:
        await update.message.reply_text("❌ *Maaf, data gempa terkini tidak dapat diambil.*")
```

Fig. 9. Command for Handle the Earthquake

10. Map Command

Creates a map of the location of the latest earthquake using data from the BMKG API and sends an HTML file containing the map to the user.

```
async def peta(update: Update, context: ContextTypes.DEFAULT_TYPE):
    data = get_earthquake_data()
    if data:
        lat, lon = map(float, data['Coordinates'].split(","))
        lokasi = data['Wilayah']
        map_path = create_map(lat, lon, lokasi)
        await update.message.reply_document(document=open(map_path, 'rb'), filename="gempa_map.html")
    else:
        await update.message.reply_text("❌ *Tidak ada data gempa terbaru untuk peta.*")
```

Fig. 10. Command for Handle the Earthquake Map

11. Mitigation Command

Provides mitigation recommendations based on the latest earthquake magnitude and depth data.

```
async def mitigasi(update: Update, context: ContextTypes.DEFAULT_TYPE):
    data = get_earthquake_data()
    if data:
        try:
            magnitudo_terakhir = float(data['Magnitude'])
            kedalaman_terakhir = data['Kedalaman']
            rekomendasi = rekomendasi_mitigasi(magnitudo_terakhir, kedalaman_terakhir)
            await update.message.reply_text(f"⚠️ *Rekomendasi Mitigasi Gempa*\n"
                                            f"Magnitudo: {magnitudo_terakhir}\n"
                                            f"Kedalaman: {kedalaman_terakhir} \n\n"
                                            f"{rekomendasi}")
        except ValueError as e:
            await update.message.reply_text(f"❌ *Terjadi kesalahan dalam memproses data gempa: {e}*")
    else:
        await update.message.reply_text("❌ *Tidak ada data gempa untuk rekomendasi mitigasi.*")
```

Fig. 11. Command for Handle the Mitigation

12. Shakemap Command

Retrieve and display an earthquake shaking map (shakemap) if available in the earthquake data.

```
async def shakemap(update: Update, context: ContextTypes.DEFAULT_TYPE):
    data = get_earthquake_data()
    if data:
        # Periksa jika shakemap URL tersedia
        shakemap_url = data.get('ShakemapURL', None)
        if shakemap_url:
            await update.message.reply_photo(shakemap_url)
        else:
            await update.message.reply_text("❌ *Shakemap (peta guncangan) tidak tersedia untuk gempa ini.*")
    else:
        await update.message.reply_text("❌ *Tidak ada data gempa terbaru untuk shakemap.*")
```

Fig. 12. Command for Handle the Earthquake Shakemap

13. Tsunami Command

Uses a prediction model to determine whether the latest earthquake has tsunami potential. This function provides an evacuation warning if a potential tsunami is found.

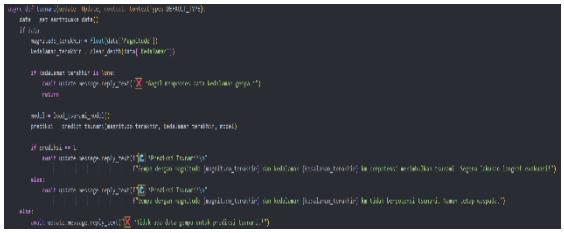

Fig. 13. Command for Handle the Tsunami

14. Main

The main function for running the Telegram bot. It registers all command handlers such as `/start`, `/earthquake`, `/map`, and others, and starts polling to listen for commands from the user.

```
def main():
    application = Application.builder().token(TOKEN).build()

    application.add_handler(CommandHandler("start", start))
    application.add_handler(CommandHandler("gempa", gempa))
    application.add_handler(CommandHandler("peta", peta))
    application.add_handler(CommandHandler("mitigasi", mitigasi))
    application.add_handler(CommandHandler("shakemap", shakemap))
    application.add_handler(CommandHandler("tsunami", tsunami))

    application.run_polling()

if __name__ == "__main__":
    main()
```

Fig. 14. Main Function to Running the Bot

Earthquake Analysis with Telegram Bot

The telegram bot will display up-to-date earthquake information by clicking a specific command to display it. Before starting to run the bot, the script that has been created must be run with a command prompt or Visual Studio Code. After clicking run, the bot will respond according to the existing command.

Fig. 15. Telegram Bot View

In the display section of the telegram bot called InfoGempa with the username @gempaSTMKG_bot and its description.

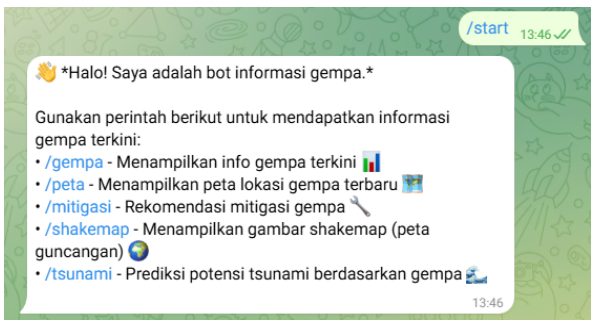

Fig. 16. Command Start on Telegram Bot

When we type the start command, the bot will respond according to Figure 16, which gives the option to provide what information. There are several

parameters in the form of earthquake information, maps of the location of the epicenter, earthquake mitigation, maps of shaking from the impact of the earthquake, and predictions of tsunami potential based on the information provided using the Ilogistic regression method.

Fig. 17. Earthquake Command on Telegram Bot

When we click or type the earthquake command, it will output the latest earthquake information. It provides information such as location, magnitude, depth, time, tsunami potential, and the area felt on the MMI scale.

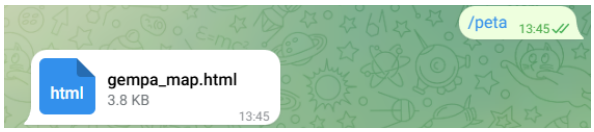

Fig. 18. Map Command on Telegram Bot

When we perform the map command, it will display the location of the earthquake based on the latest earthquake information. The map will be sent in html form and will be attached as shown in Figure 19.

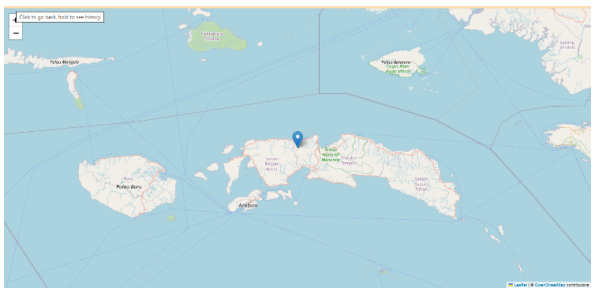

Fig. 19. Earthquake Location Map

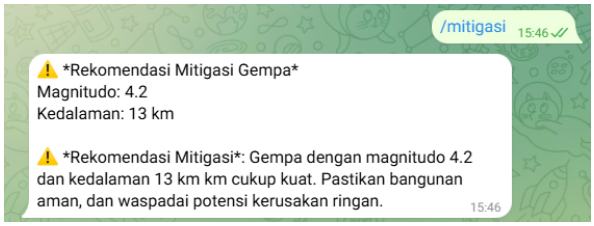

Fig. 20. Mitigation Command on Telegram Bot

Displays recommendations for earthquake mitigation displayed on earthquake information with magnitude and depth analysis. In addition, it displays recommendations for mitigation due to earthquakes.

Here is the link from the telegram bot: https://t.me/gempaSTMKG_bot

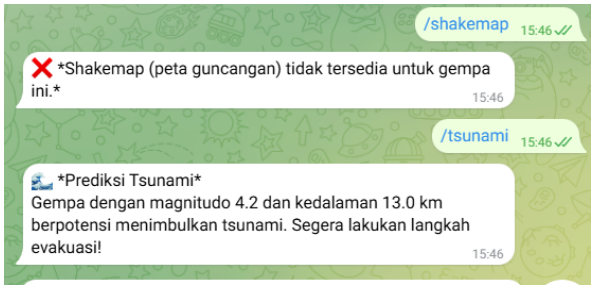

Fig. 21. Shakemap and Tsunami Command

The shakemap command will display the shaking map if available, but in Figure 2.7 it is not available from the BMKG API. Then the tsunami command will provide prediction information on whether the earthquake that occurred had the potential to cause a tsunami or not. Here, Machine Learning is used to predict using the regression method based on magnitude and depth scale.

CONCLUSION

The use of Artificial Intelligence (AI) technology in the development of Telegram bots for earthquake information has proven its benefits in improving the speed, accuracy and accessibility of information. The Telegram bot utilizes data from the BMKG API to deliver information such as earthquake location, magnitude, depth, tsunami potential, and mitigation recommendations. In addition, the integration of machine learning, specifically Logistic Regression, allows the bot to predict tsunami potential based on the latest earthquake data (Bakhshi et al., 2021; Zhang & Li, 2019). With features such as earthquake location maps, data visualization, and predictive analysis, this solution is able to help people understand earthquake risks and take appropriate mitigation steps. The application of this bot is expected to increase public awareness and preparedness for earthquake disasters, while reducing the risk of loss of life and material. Overall, the development of AI-based Telegram bots is an innovative step that can support disaster mitigation in Indonesia, especially by utilizing information technology for real-time information dissemination (Santos et al., 2020; Becerra & García, 2020).

Further development of the Telegram bot could include adding features such as integration with tsunami early warning systems and socio-economic impact analysis, which would provide more

information to users. To improve the accuracy of tsunami predictions, machine learning models can be trained using a wider range of earthquake data (Liu & Zhang, 2020). In addition, it is important to conduct educational programs to improve public understanding of the use of bots and effective disaster mitigation measures (Shah & Tiwari, 2021). Collaboration with BMKG and other relevant institutions will also strengthen the accuracy of the data used and expand the reach of the benefits of these bots to the wider community.

DAFTAR PUSTAKA

Bakhshi, S., Mohammadi, A., & Mohammadi, S., (2021). Real-time earthquake communication through Telegram bots: A case study, *Journal of Disaster Management and Mitigation*, 14(3), 112-121.

Barua, D., & Kumar, A., (2020). Machine learning-based tsunami risk prediction following earthquakes, *Journal of Geophysics and Disaster Management, 22*(1), 33-45.

Becerra, J., & García, L., (2020). Applications of AI and IoT in disaster management and mitigation, *Journal of Information Technology in Disaster Management, 18*(3), 147-160.

Carvalho, J., Silva, L., & Costa, M., (2021). Enhancing tsunami early warning systems using machine learning for faster response times, *Disaster Response and Recovery*, 30(2), 57-72.

Chatterjee, S., Das, A., & Ray, S., (2021). Improving disaster awareness through chatbots: A case study of earthquake information dissemination, *International Journal of Disaster Studies*, 18(4), 213-229.

Chatterjee, S., et al., (2021). Using Chatbots for Disaster Awareness and Response: A Case Study, *Journal of Disaster Studies, 15*(3), 45-60.

Güney, G., & Aksoy, T., (2019). Machine learning techniques for earthquake prediction: A review, *Journal of Seismology and Earthquake Engineering, 28*(1), 101-115.

Liu, Y., & Zhang, J., (2020) Improving tsunami prediction using machine learning and earthquake data, *Journal of Earth Science and Technology, 29*(2), 188-200.

Li, J., & Zhang, H., (2018). AI in earthquake warning systems: Accelerating response times for disaster management, *Journal of Earthquake Research, 25*(4), 98-110.

Russell, S. J., & Norvig, P., (2020). Artificial Intelligence: A Modern Approach, 4th ed., Pearson.

Saini, P., & Soni, M., (2020). Artificial Intelligence for Disaster Management: A Review, *International Journal of Computational Intelligence Systems, 13*(5), 809-825.

Santos, D., Ferreira, T., & Costa, P., (2020). AI-powered chatbots for disaster management: Improving emergency response through better coordination, *Disaster Risk Reduction and Management Journal, 12*(3), 134-146.

Shah, S., & Tiwari, S., (2021). Public education programs for effective disaster mitigation: The role of AI-powered bots, *Journal of Public Safety and Technology, 17*(2), 105-118.

Xu, B., Liu, W., & Zhang, Z., (2020). The role of AI in earthquake data processing for disaster mitigation, *Journal of Geophysical Research, 22*(5), 78-91.

Zhang, L., & Wang, Q., (2019). Deep learning in earthquake classification and its impact on public awareness, *Seismological Research Letters, 90*(6), 1750-1762.

Zhang, W., & Li, H., (2019). Machine Learning in Earthquake Prediction and Tsunami Risk Assessment, *Seismological Research Letters, 90*(4), 1225-1236.

Badan Meteorologi, Klimatologi, dan Geofisika (BMKG), "Data Gempa Terkini," 2023. [Online]. Tersedia: https://bmkg.go.id. [Diakses: 16-Des-2024].

Telegram, "Telegram Bot API Documentation," 2023. [Online]. Tersedia: https://core.telegram.org/bots/api. [Diakses: 16-Des-2024].