Design and implementation of Awas Guncang: A mobile app for future dissemination of InaEEWS

Hanif Kurniawan[1*], *Abraham* Frederik Mustamu[1], *Maulana* Putra[2], *Nardi*[1], *Arya* Bratasena Alhaq[3], *Muhammad* Reza July Utama[4], *Aurora* Fajri Miftakhunnisa[4], and *Benyamin* Heryanto Rusanto[1]

[1]School of Meteorology, Climatology, and Geophysics, Instrumentation of Meteorology Climatology Geophysics Department, 15119 Tangerang, Indonesia
[2]Meteorology, Climatology, and Geophysics Agency, Directorate of Instrumentation and Calibration, 10610 Jakarta, Indonesia
[3]Meteorology, Climatology, and Geophysics Agency, Directorate of Network Systems and Communication, 10610 Jakarta, Indonesia
[4]Meteorology, Climatology, and Geophysics Agency, Directorate of Engineering Seismology Potential Geophysics and Time Signals, 10610 Jakarta, Indonesia

Abstract. Indonesia is a country with a high level of seismic risk, making an effective Earthquake Early Warning System (EEWS) crucial. The Agency of Meteorology, Climatology, and Geophysics (BMKG) has been developing the Indonesia Earthquake Early Warning System (InaEEWS). This system requires a fast and reliable dissemination channel to reach the general public. This research aims to design and implement a mobile application named “Awas Guncang” as a future dissemination solution for InaEEWS. The application was developed using the Flutter framework. The notification delivery method relies on Firebase Cloud Messaging (FCM), a cross-platform push notification service, for real-time alert transmission, even when the application is terminated. This research yields a functional mobile application prototype capable of receiving, processing, and displaying earthquake early warning notifications following the designed workflow. The application is proven to deliver full-screen intent notification alerts based on specified Peak Ground Acceleration (PGA) thresholds, a measure of earthquake shaking intensity. Therefore, “Awas Guncang” demonstrates its potential as an effective and responsive platform for disseminating InaEEWS information, which can enhance public preparedness for earthquake disasters in Indonesia.

1 Introduction

The high mobile usage penetration in Indonesia, with 180 million users [1], is very promising for early warning dissemination. Mobile devices have been in high demand in this century for communication, information, and entertainment, which are now widely accessible to almost everyone. Additionally, mobile cellular connections have become increasingly widespread, covering more remote areas. Hence, it is a perfect solution for disseminating earthquake early warning systems, especially in an earthquake-prone country like Indonesia.

Indonesia sits above the world’s most active tectonics, the Pacific Ring of Fire. Therefore, there are megathrust sites across it. The west-southern Java megathrust is one of the most seismically active regions, posing a significant threat to bring large earthquakes and tsunamis due to its seismic gap [2]. There have only been two significant earthquakes in this region over the last two decades: the Mw 7.8 megathrust event on July 17, 2006, and the Mw 6.8 reverse fault intraslab earthquake on September 2, 2009 [3]. For instance, an earthquake with an intensity of MMI VIII occurred in Jakarta, associated with a megathrust event of Mw 7.4 on January 5, 1699, and a similar intensity for a shallow crustal fault of Mw 5.5 on October 10, 1834 [4]. Thus, earthquake early warning systems (EEWSs) are crucial for Indonesia to be prepared for future megathrust earthquakes.

The foundation of the current EEW technology in production is that an earthquake generates seismic waves, P and S-waves. P-waves travel faster and are the first to be detected by seismic instruments. Meanwhile, the S-wave is slower and more destructive because its motion is perpendicular to the wave’s direction. Respectively, P and S-wave have travel speeds around 5-8 km/s and 3-4 km/s [5]. An earthquake event can be recognised and analysed using the P-wave before the arrival of the S-wave, namely the golden time.

The development of earthquake early warning mobile device applications has been widely undertaken. MyShake uses the United States Geological Survey (USGS) and its seismic networks. The MyShake app serves as both a disseminator of earthquake early warnings and a sensor within its network [6]. Then, the EWapp application uses the Irpinia Seismic Network (ISNet) seismic network, which monitors seismic activity in southern Italy with an EEWS called PRESTo [7]. BMKG (Agency of Meteorology, Climatology, and Geophysics), the Indonesian government agency responsible for earthquake monitoring in Indonesia, is

* Corresponding author: hanifkurniawan@stmkg.ac.id

currently developing an earthquake early warning network in Indonesia called InaEEWS (Indonesia Earthquake Early Warning System) [8], which also requires its application on mobile devices. Therefore, this study will design and implement “Awas Guncang”, a mobile device application prototype, as a reference model of InaEEWS’ future public dissemination.

2 Current InaEEWS’ development

The development of Indonesia’s earthquake early warning system, namely InaEEWS, is an ongoing project by BMKG. Recently, Rudyanto et al. [9] did a performance test of pilot EEWS in western Java, Indonesia. This EEWS was built using Earthworm, an open-source project widely used for automatic earthquake processing [9, 10].

The foundation of this system is a seismic sensor network comprising 199 seismic stations, which consist of three different types: seismic collocated stations (seismometers and force balance accelerometers), strong motion stations (force balance accelerometers), and MEMS accelerometers [9]. The data is sent to a centre server called Accelero Server at BMKG Headquarters. This server provides data for Earthworm used for EEWS.

After new data is collected, the system will start to analyse any earthquake events recorded. The raw signal from the sensor undergoes a series of processing steps, including filtering, detrending, and integration, to convert the acceleration data into displacement data. The system uses an enhanced T^{pd} method to detect the arrival of P-Waves (first seismic waves) in real-time [9]. To prevent false alarms caused by noise, the system requires confirmation from multiple stations. A new earthquake event is considered valid if it is detected by at least three adjacent stations (called a Trigger Group). Once confirmed, the system will input data from other more distant stations (Estimation Group) to increase accuracy.

The value of the magnitude is then calculated after the event confirmation. Earthquake locations and depths are estimated using a modified T^{now} method, which uses data from both triggered and untriggered stations to narrow down possible locations. Earthquake magnitudes are calculated using the M^{pd} method, which is based on the peak displacement (Pd) of the first 3 seconds of the P-wave signal [9]. An initial alert is issued as soon as the first parameter is successfully calculated. The system then continues to iterate as data from more stations comes in, allowing location and magnitude estimates to become more accurate over time.

The system performance was evaluated based on 27 earthquake events that occurred in West Java between October 2022 and May 2023. More than half of the epicentre estimation errors are below 10 km, and almost all are below 30 km compared to the BMKG catalogue [9]. Most of the earthquake event times calculated by EEWS differ from the final BMKG data by only about 2 seconds. More than half of the magnitude estimates fall between 0.3 and 0.6, which is consistent with the BMKG catalogue value [9].

The reconstruction of the Cianjur earthquake (M_w 5.6 on November 21, 2022) was an important test for this system. The system managed to issue the first warning within just 6 seconds after the earthquake origin time [9]. This fast warning has the potential to provide a warning time of between 4 and 19 seconds for major cities such as Bogor, Depok, Bandung, and Jakarta before the shock arrives [9]. After a few seconds, the earthquake parameters stabilized to a good level of accuracy: the epicentre location error was approximately 15 km, the depth error was around 2 km, and the magnitude error was approximately 0.8 [9]. With this accomplishment, it is possible to build an operational earthquake early warning system in Indonesia, at least in western Java, especially with mobile-device-based dissemination.

The mobile device app is an untapped territory of BMKG’s public warning system, arguably for the information system. BMKG often uses the cellular short message service for its operational warning system, InaTEWS (Indonesia Tsunami Early Warning System), and BMKG Signature. Although it is working, the effectiveness and accuracy are not good enough. Cellular short messages use the cluster of base transmission towers (BTS) as the target of dissemination. The capabilities of mobile devices can increase thanks to their computational capability. This distributed computation not only increases its accuracy, but also reduces the computational work on the server.

3 App design and methodology

“Awas Guncang” is developed using the rapid application development (RAD) method. Considering the urgency of the development of “Awas Guncang”, RAD emphasizes that it should be developed quickly without sacrificing its expectations and performance [11].

This method encompasses requirements, user design (with iteration and feedback), development, and implementation. The background requirements are to design a smartphone application for earthquake early warning that can provide notification before the S-wave is felt and to create a user design that is easy to understand and use by the public. The proposed user design includes a block diagram, flowchart, and database structure. User design iterations are carried out by multiple parties, including thesis supervisors, thesis examiners, and the InaEEWS development team. This activity is repeated until system suitability is achieved. In the development phase, the developer begins building the app and then tests it through implementation and simulation.

3.1 Flowchart diagram

The whole workflow illustrates “Awas Guncang” as a warning and earthquake explorer app. There are at least eight activities: Introduction Acitvity, Home Activity, Alert Activity, Notification Setting Activity, Webview, Demo Activity, About Activity, and Report Activity.

When the app is opened, which means a foreground mode, a splash screen will show up, and multiple processes will happen: subscribe to FCM topics and check the app’s permissions. A new user will be detected if multiple permissions are disallowed. If any of the permissions are inaccessible, whether the user is new or not, the app will launch IntroductionActivity to ask them to allow the app's needed permissions and explain why those are needed. Those required permissions are "allow all the time" location, notification, and disable battery optimization. Because the app is running in the background and needs to calculate PGA, allowing all the time locations and disabling battery optimization are required to make it work. After all requirements are allowed, the app will start its background service in a different *bubble* than the app, a unique environment that allows the service to run independently, so it can still run even when terminated. Figure 1 illustrates the whole workflow of "Awas Guncang" as a warning and earthquake explorer app.

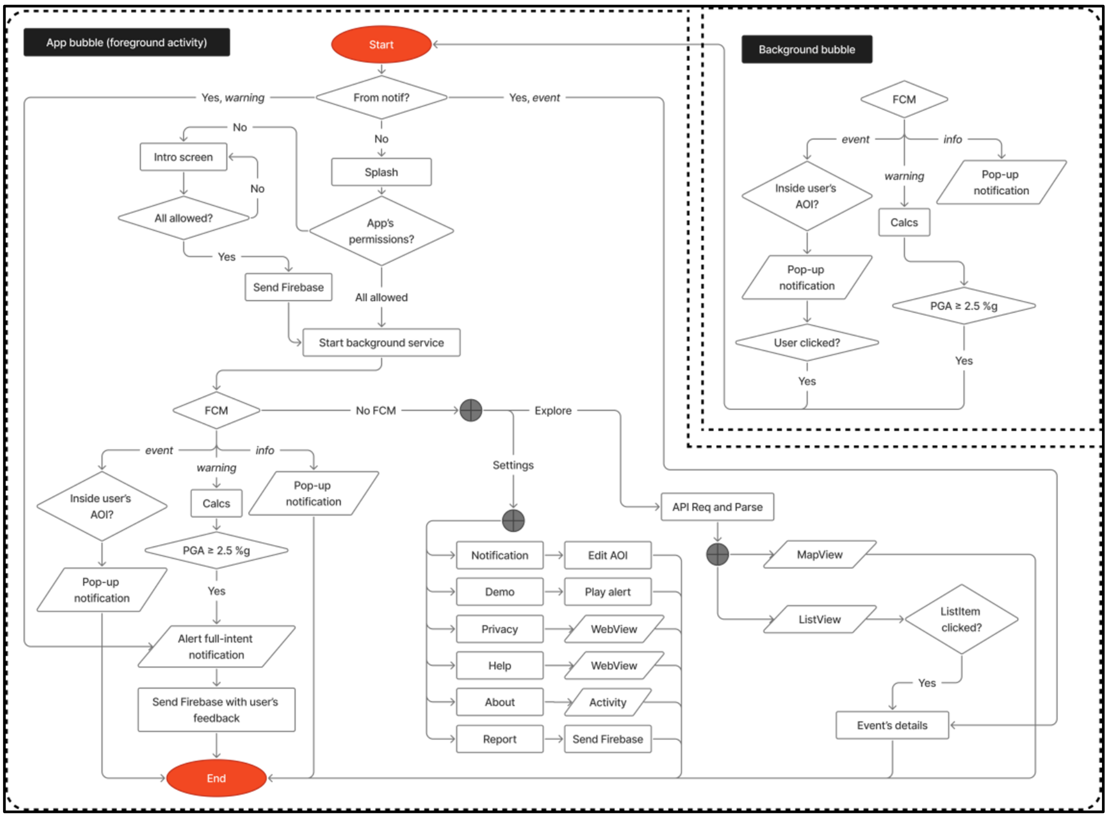

Fig. 1. Flowchart diagram.

"Awas Guncang", as an app, does not merely work in its environment. It uses multiple processes in servers that can be called using APIs. "Awas Guncang" uses the earthquake events API from BMKG to provide information about past earthquake events. The event controller processes this data, which then displays it as a list of earthquake history and markers on the map.

Still in foreground mode, if no FCM is received, there are two menus users can access: Explore and Settings. In Explore, users can look up earthquake events interpreted in ListView and MapView. The app requests past earthquake events to an API from BMKG. The response is then processed by the controller and passed to the model to be suited to the app's format. The ListView contains events' area names, magnitudes, and datetime information. If the ListItem is clicked, it will show more specific event details in a new activity. The activity shows MapView, which contains the user and specific event epicentre marker, and the event's parameters in texts and symbols, such as magnitude, area, datetime, coordinate, depth, and distance with the user. One of the notification channels is the *event* FCM topic notification channel. This notification channel can be customized in the Settings menu. Users can select their area of interest (AOI), a circular boundary that filters the *event* channel. The app will show a pop-up notification if an event epicentre is inside the user's AOI.

The Settings menu is a hub of user control. It contains features like Customize Notification, Alert Demo, Privacy, Help, About, and Problems Report. Users can self-trigger the alert using Demo Alert in the Settings menu for educational purposes. However, the user will be prompted first not to play it in public, and the DemoActivity has slightly different text, which explains that it is only a simulation. Privacy is a WebView displaying Awas Guncang's privacy policy on its website. Just like Privacy, Help is also a WebView linked to FAQ on Awas Guncang's website. If users have questions that are not available in the FAQ, they can always ask them through the Report submenu. Moreover, in this submenu, users can report bugs, request features, and contact the developer.

In background mode, the only app's task is to stand by and receive FCM. If the message's topic is *a warning*, there will be PGA calculations, and if it is more than 2.5%g, the app will issue an alert bringing it to the foreground with full-intent notification. Users can submit their experience of the shaking after the shake. The collected data are the user's location, intensity experience, and the event's id. Otherwise, *the event* topic's message will be filtered with the user's AOI; if it is inside the AOI, there will be a notification. General information can be sent by the developer using the *info* FCM topic.

3.2 Database

Regarding privacy concerns, the cloud database does not play a paramount role in "Awas Guncang". Most of the calculations are done in the app, and clearly, no sensitive data, i.e., geolocation, is stored in the cloud database without the user's acknowledgement and concern. However, for debugging and crowdsourcing purposes, multiple data are saved in the cloud database this app is using, Cloud Firestore. It is a NoSQL cloud database provided by Firebase that can be accessed directly via Software Development Kits (SDKs).

There are three main collections: device, feedback, and report, as shown in Fig. 2. The device is used to store the device's information and registration timestamp. This is done when the user launches the app for the first time after completing all required permissions. Users submitted experiences of aftershock will be sent in feedback collection, which stores the device's ID, event ID, user's submitted geolocation, user's submitted intensity experience, and timestamp. The report submenu will send its content in the report collection, which contains the app's version, device's ID, description, email, name, issue type, status (for developer purposes, i.e. pending, on review, etc.), and timestamp. It is important to note that all data is submitted to the cloud database only after the user's explicit consent and acknowledgement.

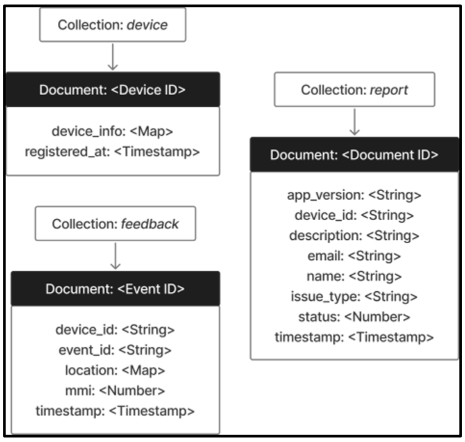

Fig. 2. Cloud firestore database structure

3.3 Alert's block diagram and algorithm

The alert technology in this app is powered by the highly flexible Firebase Cloud Messaging (FCM) system. This system seamlessly connects devices registered in the application to receive messages and notifications from developers. FCM sends messages to devices through topics or tokens, allowing the application to efficiently manage subscriptions to topics. These topics include *warning* for the earthquake alert channel, *event* for the analysed earthquake information channel, and *info* for the general notification channel. The *event* and the *info* topics will result in pop-up notifi3cations. In contrast, the *warning* topics will be further calculated (Fig. 3), and if they fulfil the criteria, it will result in a full-intent notification.

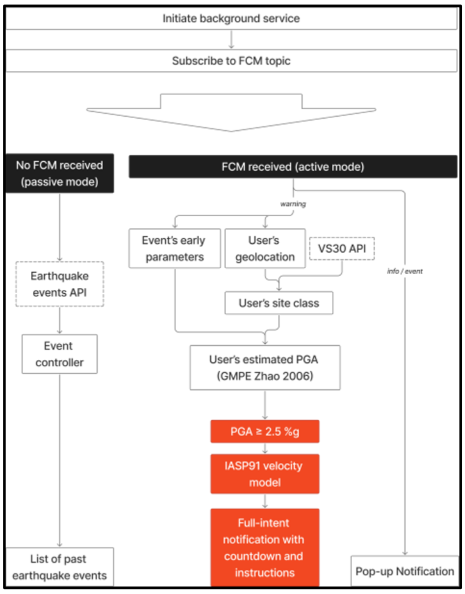

Fig. 3. Alert's block diagram.

The primary function of "Awas Guncang" is to provide EEW notification to users as quickly and accurately as possible. When there is no earthquake event, users can use the application to explore past earthquake events. Therefore, the operating mode of "Awas Guncang" can be divided into two parts: active mode, where the app receives an earthquake event and alerts users, and passive mode, where the app does not receive any earthquake events but still allows users to explore past earthquake events.

When an FCM is received, the app is in active mode. However, receiving a message does not automatically trigger an alert. If the topic of the message is *an event* or *info*, it will likely show a pop-up notification, as explained before. Even if the *warning* topic is received, the app will only display an alert if multiple conditions are met, indicating that shaking is expected for a user.

To determine shaking, "Awas Guncang" uses PGA determination using the ground motion prediction equation (GMPE) from Zhao et al. [12], which is a widely accepted and validated model in the field of earthquake engineering. The ground-motion model is:

$$\log_e(y) = aM_w + b_x - \log_e(r) + e(h - h_c)\delta_h + F_R + S_I + S_S + S_{SL}\log_e(x) + C_k \quad (1)$$

$$r = x + c\exp(dM_w) \quad (2)$$

So, to calculate the user's estimated PGA, the message needs to contain magnitude (M_w), depth (h), and epicenter's location. The distance between the user and the epicentre is x, which will help calculate the r. Other than those in the equation, they are constants. Moreover, the equation needs to calculate C_k, a site class coefficient. The average shear wave velocity within the top 30 meters of the ground ($V_{s,30}$) can determine C_k. This calculation is complex to be done locally on mobile devices, so this app efficiently uses an API to calculate it. The API method is a POST with a JSON payload of the user's geolocation that returns the user's specific site class number. After that, the app can run the equation and determine the user's estimated PGA.

If the PGA surpasses the threshold of 2.5%g or 25 gal , the threshold is inspired by Hsu and Pratomo [13], and the app will continue to calculate the countdown. The countdown calculation is done using simple Newtonian physics, with v_s the velocity of s-wave and user's-epicentre distance. The IASP91 model can help determined the v_s across the world [14], which varies in each earth's layer. The calculated time must be subtracted from the delay time, which is InaEEWS process time, to get the countdown time. At this point, all variables needed for the alert are fulfilled. Then, the app, a crucial component in the alert process, will show a full-screen intent notification playing its alerting sound and displaying the countdown also the self-evacuation guide, drop, cover, hold on.

4 Development and results

"Awas Guncang" was developed using Flutter, an open-source SDK for cross-platform app development. While the SDK supports both Android and iOS platforms, the decision was made to focus on the Android operating system due to the unavailability of the specific environment, Xcode and macOS, required for the iOS platform at the time of this research. Table 1 provides a complete specification and the environment used in this research.

Tabel 1. Hardware and software used in the development environment.

No	Name	Specification / Version
1.	Hardware	Laptop Nitro 5 AN515-57-71EX
2.	Processor	Intel® Core™ i7-11800H processor Octa-core 2.30GHz
3.	Main memory	16 GB DDR4 SDRAM
4.	Graphics card	NVIDIA® GeForce RTX™ 3050Ti 4GB memory
5.	Storage	SSD 512 GB
6.	Operating system	Windows 11 23H2
7.	IDE	Visual Studio Code 1.101.0
8.	Framework	Flutter 3.22.2 and Dart 3.4.3
9.	Compiler	Java 19
10.	Android device	AVD Pixel 8 Android 14

4.1 Permissions implementation

Implementing the "Awas Guncang" application requires three essential permissions from users to function optimally as an early warning system. First, notification permission (refer to Fig. 4a) is a fundamental requirement for the application to be able to send crucial warnings from InaEEWS to users. Second, location access permission (Fig. 4b) with the "Allow all the time" option is required so that the background service can verify whether the user's location is within the affected area when there is a warning. It is important to note that this location access does not run continuously. However, the user's location is fetched after receiving a specific warning message from the FCM. With this method, preserving both user privacy and battery conservation. Finally, the app asks for an exemption from battery optimization (Fig. 4c) for reliability of background service purpose. This permission prevents the operating system (OS) from abruptly ending the background services of the application, thereby keeping the app in a state of readiness to receive and process earthquake warning data instantaneously.

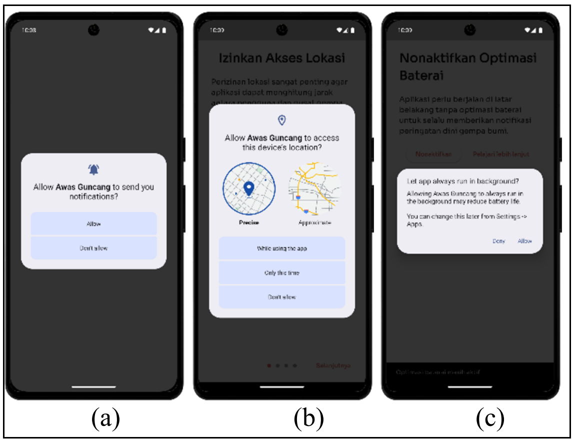

Fig. 4. Screenshot of permission interface. (a) Notification permission. (b) Location permission. (c) Disable battery optimization.

4.2 Alert implementation

The primary function of "Awas Guncang" is its alert notification. The app has successfully implemented full-intent notifications for *warning*-type FCM topic. Users can expect their devices, especially in sleep/standby state, to automatically launch the app to the alert display shown in Fig. 5a. If the device is being used, the user will only receive the banner (Fig. 5b). This is a limitation of full-intent notification in Android, which is designed to prevent apps from scamming and distracting users. Nevertheless, the user should know the alert notification during device usage. The user will be redirected to the alert display if the notification is pressed.

Fig. 5. Screenshots of alert interface. (a) Alert displayed in full screen while device on standby/sleep. (b) Alert displayed in banner while device on usage.

The bottom of the alert display has a button to share the user's experience. Users can share their experience with the developer by submitting their position when the shaking happened (Fig. 6a) and the intensity they felt (Fig. 6b). Users can pick their position manually, considering there are chances they have been shifted due to commuting or any other reason.

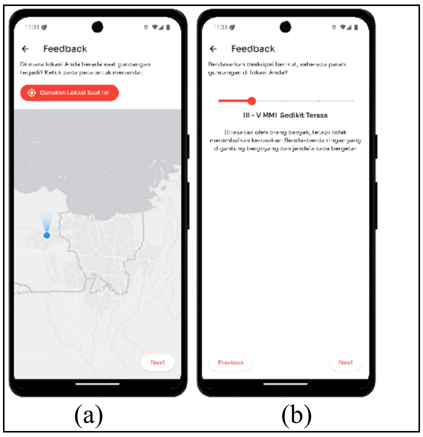

Fig. 6. Screenshots of user's feedback interface. (a) User's location input. (b) User's intensity input.

4.3 Background service

The background service has been successfully implemented, allowing users to receive alerts without constantly opening the app. However, Android has set some requirements for its optimal functioning: disabling battery optimization and showing silent notifications. As explained before, the battery optimization permission is implemented when users first launch the app. The only remaining requirement is to display silent notifications. This policy from Android informs users that certain apps are running in the background. Once these requirements are met, the background service can run, as seen in Fig. 7b. We understand that the silent notification may be perceived as distracting (Fig. 7a). We acknowledge this. We are working to balance providing necessary information and minimizing potential distractions.

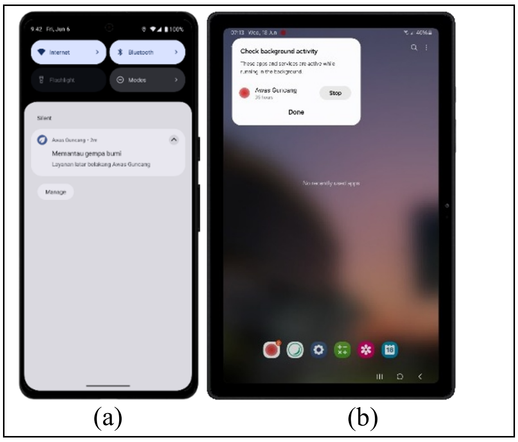

Fig. 7. Screenshots of background service implementation. (a) Silent notification indicated background service is running. (b) "Awas Guncang" background service in One UI's recent app interface.

4.4 Explore menu

Users can explore historical earthquake events while the app is in passive mode. For convenience, the app visualizes these events on the MapView, as shown in Fig. 8a. The MapView shows the user's location and past earthquake epicentre. Additionally, users can click on an event in the ListView (Fig. 8b) to view more details about it (Fig. 8c). To be more human-friendly, the origin time of earthquake events has been modelled to natural language timestamps, i.e., today, yesterday, etc. There is also distance information between the user and the epicentre.

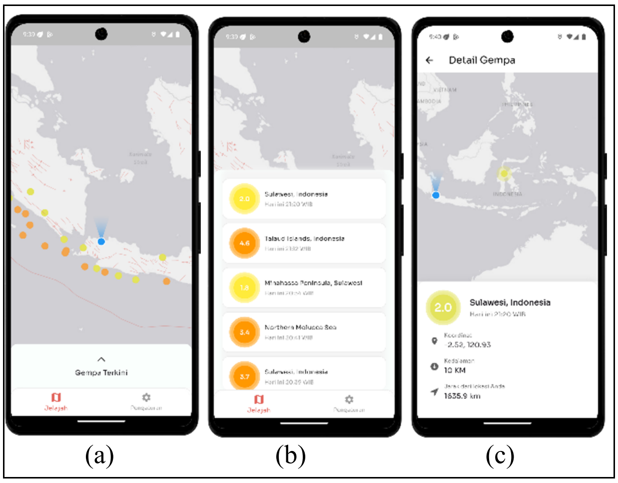

Fig. 8. Screenshots of explore interfaces. (a) MapView of the explore interface. (b) ListView of the explore interface. (c) Specific event's detail.

4.5 Settings menu

In the implementation results of the app, the settings menu has been successfully realized. One of its main capabilities is the Notification Settings submenu, which allow users to customize their *event* notification channel. Users can set their area of interest for earthquake information update notification as seen in Fig. 9b. In addition, the alert demo feature has also been implemented (Fig. 9c). As a result, this feature can be effectively used as a means of familiarization for users.

Informational submenus such as Privacy Policy (Fig. 9d), Help (Fig. 9e), and About (Fig. 9f) have also been successfully implemented to build trust. Implementing Privacy Policy and Help submenus utilizes the WebView component, a technical approach proven efficient for displaying updateable content that linked to Awas Guncang's website. With about submenu, users can learn more about the app, such as the features. Moreover, this application demonstrates two-way communication functionality through the Report Problem feature (Fig. 9g). According to the test results, this feature successfully sends report data from users, where the data is successfully stored in the Cloud Firestore database. The successful implementation of this feature proves the existence of a feedback mechanism that is running and ready to be used for iterative development.

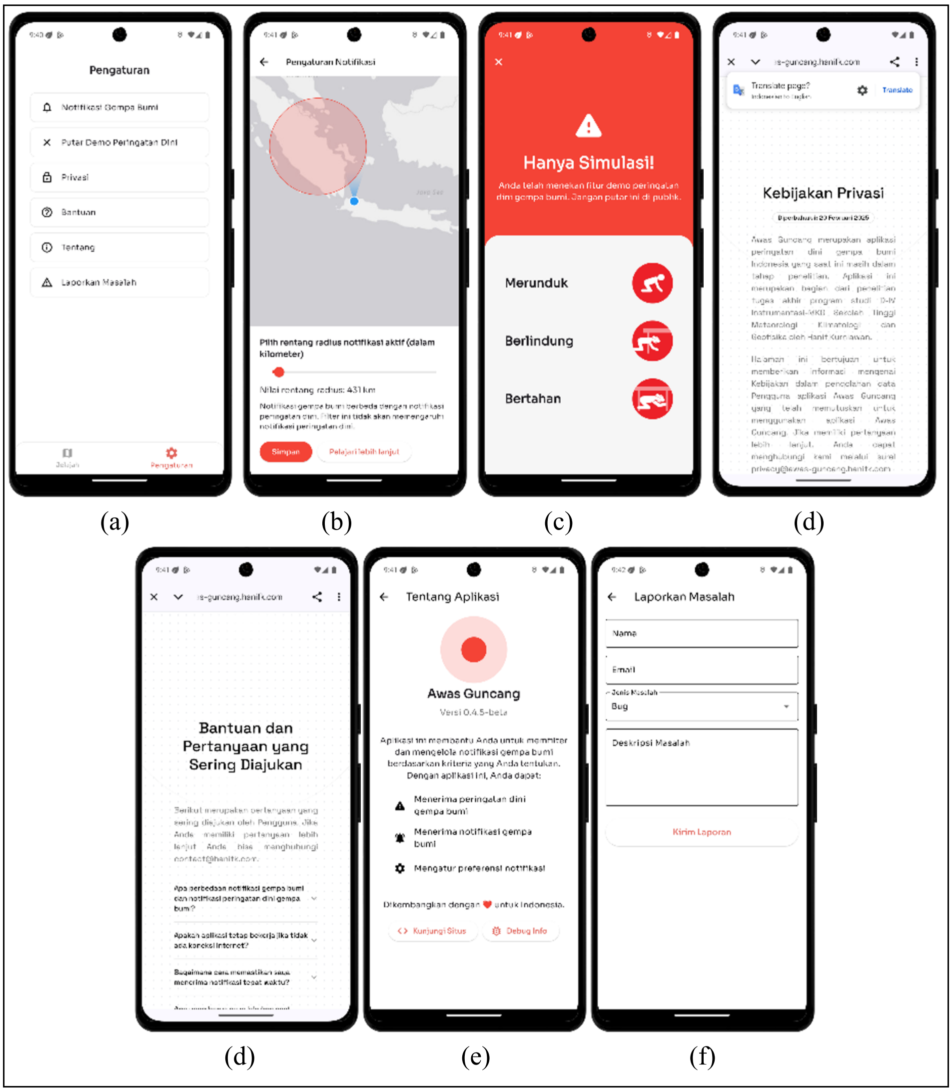

Fig. 9. Screenshots of settings interfaces. (a) List of settings submenus. (b) Notification Setting interface. (c) Alert Demo interface. (d) Privacy Policy submenu redirects users to Awas Guncang's website (e) Help submenu redirect users to Awas Guncang's website (f) About submenu tells users about the feature and more info about the development (g) Report Problem interface for users to have a direct communication to the developer.

Thus, the development results show that this Settings Menu functions more than just a collection of complementary utilities. It is a strategic component successfully implemented in the app. Its series of functions, which have been tested collectively, improve usability, build transparency, and provide functional user participation channels. These concrete results support the application's main purpose as an effective InaEEWS dissemination media.

5 Early warning simulation

As InaEEWS is still in the development phase, there are no live feeds to monitor the earthquake and send necessary data to "Awas Guncang". To address this issue, especially to test the functionality of this app, a simulator needs to be designed. The simulator, a web-based trigger for the FCM, is not a complete replica of InaEEWS's workflow. It primarily handles the early parameter inputted in its form, rather than determining it (Fig. 10). Initiating the simulation involves the developer filling out the earthquake parameters in the form, which serves as the FCM's payload. A timestamp for the earthquake event's origin time is automatically added to the payload with the current time value. The 'kirim sekarang' submit button sends the message to the FCM cloud service, distributing it to devices subscribed to the topic. For added security, the developer is required to enter a password before pressing the submit button.

Thanks to the background service, which creates a separate bubble from the app in the device's system, the app can receive and process the FCM without needing to be in the foreground or even terminated. Moreover, if the device is restarted, the background service can start its own after startup.

However, despite being required to disable this app's battery optimization, some mobile devices may experience the background service being terminated. This problem is because some Android manufacturers have their own custom UI, such as OneUI, HyperOS, and ColorOS, and the solution is tailored to these variations. FCM, a cloud service, also means the app requires an internet connection.

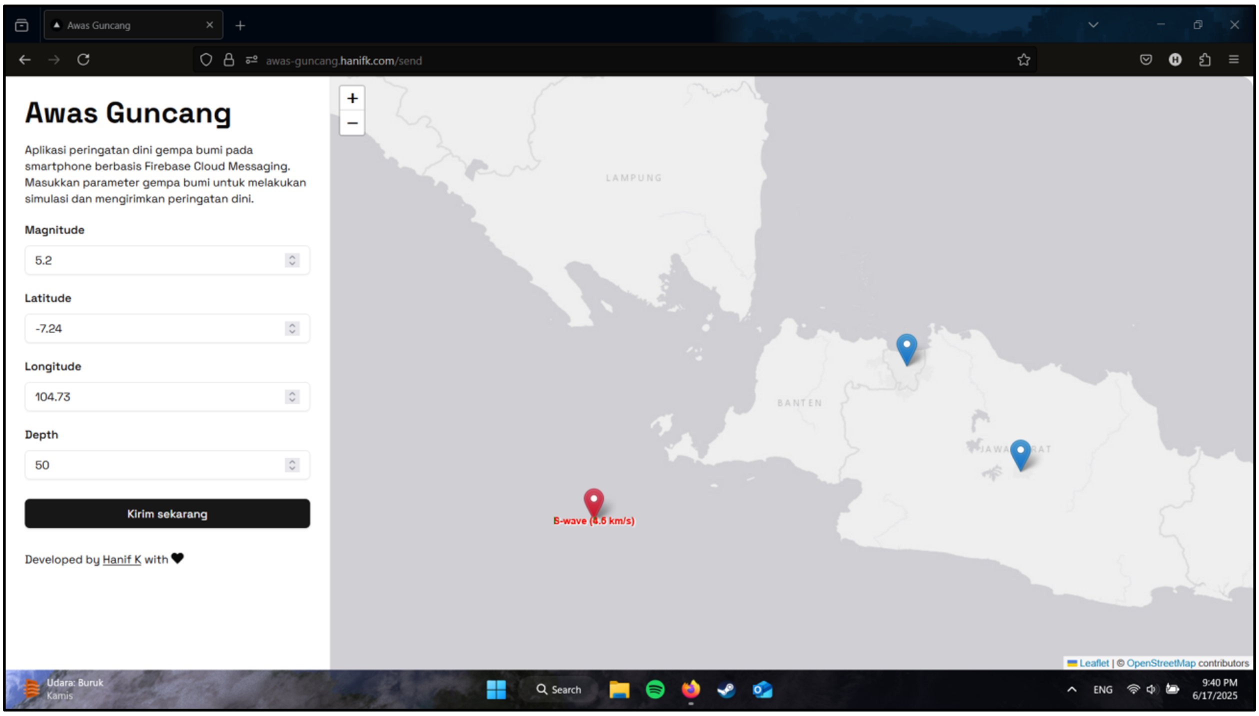

Fig. 10. Interface of Awas Guncang's simulator. This web-based simulator inputs earthquake source parameters (magnitude, location, depth, timestamp) and triggers warning notifications through FCM to test the app's responding functionality. There are markers of the earthquake's epicentre and big cities in Indonesia. Furthermore, there is an animation of a growing circle travelling to portray the movement ofay the earthquake waves, P and Sd movement. If the circle's border hits any cities, it will display the estimated PGA.

6 Performance analysis

Some tests and assessments can measure the performance and versatility of "Awas Guncang". This app is not merely a warning app that interacts with its users. It also interacts with other machines (APIs). Hence, the analysis will focus on two crucial aspects: alert performance and user experience performance.

6.1 Alert performance

The alert notification system's performance is not just a theoretical concept, but a practical necessity for future InaEEWS implementation. To assess its potential, a series of tests are conducted to evaluate the app's speed, resilience, and accuracy in receiving and processing information from the server. The performance of the warning delivery system in the "Awas Guncang" application was tested through a series of earthquake simulations to measure the latency from the server side to processing in the user application.

Figure 11 shows the frequency distribution of latency recorded from 1123 data during the test. Latency in this test is defined as when the warning message is sent by FCM until the application on the user's device successfully calculates the PGA. Based on the data in Fig. 11, the app shows a high-speed warning delivery capability. The peak of the latency distribution, or the most frequent delay time, is in the range of 2 seconds with a frequency of almost 25%. Most of the data, more than 75%, is concentrated in the latency range of 1 to 4 seconds. These results indicate that for the majority of users, warning information can be received and processed by the application in a very short time, providing vital time for users to be able to react.

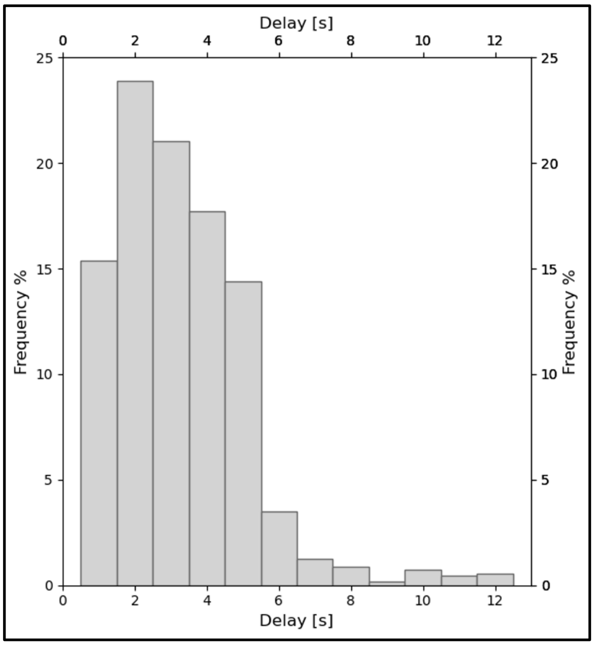

Fig. 11. Frequency distribution of the alert latency on the app during the performance test. The X-axis represents the delay time or latency in seconds (s). This latency is calculated from when the warning message is sent by FCM until the application on the user's device successfully calculates the PGA. The Y-axis shows the relative frequency of each latency range in percentage (%) of the total 1123 recorded data.

To provide a broader context for these results, the performance of "Awas Guncang" is compared to similar established early warning systems. In comparison, research on EWapp reported an average latency of 1.3 seconds [7], while MyShake recorded a median latency of 2.8 seconds [6]. With an average latency of 3.3 seconds, "Awas Guncang" showed slightly higher processing times. However, this performance difference can be explained by several fundamental architectural and technological decisions that were consciously chosen during development.

Calculating *site class* for alert accuracy is the main factor contributing to the additional latency. "Awas Guncang" implements the GMPE formula from Zhao et al. [12], which requires *site class* data (based on $V_{s,30}$ values) specific to the user's location. Rather than storing the entire Indonesia $V_{s,30}$ data within the application, which would make the installation file size very large, the app is designed to make API calls to the server. This process of requesting and responding over the network inherently adds a new latency component, significantly contributing to the total measured latency. In addition, the choice of the Flutter cross-platform framework, although very efficient for development and maintenance, can incur a slight performance overhead compared to native applications such as Kotlin used in the comparison system.

External factors include FCM cloud service performance, user network conditions, and the device's limitation on background activity. Some Android custom OS, such as OneUI and HyperOS, have tweaked the original Android stock OS for better performance, which can result in inconsistent background limitations across Android devices.

However, the average latency of 3.3 seconds recorded on the app is not a deficiency but a calculated trade-off. Flutter was chosen for cross-platform development to reach broader devices and side latency by an API call to make the size of the app reliable for users. This measurable performance has proven that "Awas Guncang" is a reliable platform and meets the speed requirements to be a medium for disseminating warnings from InaEEWS in the future to the public.

6.2 User experience performance

As crucial as the alert technical resilience, user experience determines the success of the app's implementation for users. "Awas Guncang", as an early warning app, needs to be user-friendly since using this app can save lives. Many people may panic in an earthquake, and this app should make people believe they can handle the situation with better information. Therefore, this app considers user experience as a performance that needs to be assessed.

The User Experience Questionnaire (UEQ) measures the user experience quantitatively and standardly. Its items consist of six scales: Attractiveness, Efficiency, Perspicuity, Dependability, Stimulation, and Novelty.This questionnaire was completed by N=8 respondents. Tabel 2 shows that stimulation, with a value of 2.281, is the highest value of the means. This result shows that users find this product interesting, fun, and motivating. In addition to the high average value, high consistency (low variance) is also shown by Stimulation and Accuracy. Uniform user opinions indicate that most users consistently feel this pleasant and reliable experience.

Tabel 2. Mean and variants result of each scale.

No	Name	Means (M)	Variants (V)
1.	Attractiveness	1.896	1.18
2.	Perspicuity	1.656	1.16
3.	Efficiency	1.594	1.62
4.	Dependability	1.844	0.82
5.	Stimulation	2.281	0.81
6.	Novelty	1.938	1.12

The results of the benchmark analysis, as depicted in Fig. 12, are uniformly positive. Notably, Attractiveness (M=1.90), Accuracy (M=1.84), Stimulation (M=2.28), and Novelty (M=1.94) demonstrate excellent performance, placing the application in the top 10% in the UEQ benchmark. The Clarity scale (M=1.66) is categorized as Above Average, and the Efficiency scale as Good, underscoring the application's good performance and promising future.

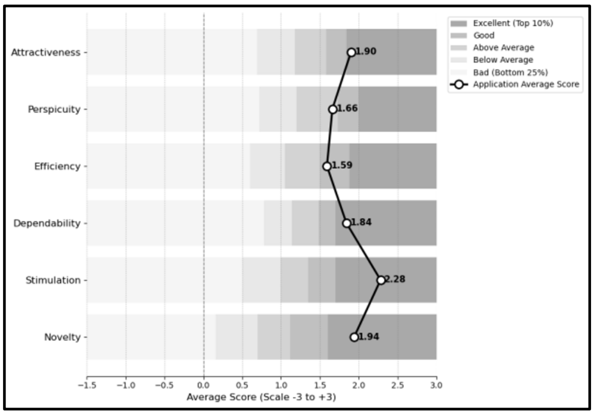

Fig. 12. Benchmark result of each scale in bar chart.

Based on the benchmark data, three scales significantly influence the KPI score: Stimulation, Novelty, and Appeal. This data indicates that the application meets user experience expectations. However, the other three scales, Clarity, Accuracy, and Efficiency, have lower scores, indicating that the application is not perfect but functions in its pragmatic quality.

The KPI results found from this survey are 1.87. If concluded together with the results of the benchmark analysis, the application has succeeded in providing an extraordinary emotional experience. In addition, the positive response of respondents is built by the application's characteristics that provide a greater impression and stimulation of novelty than its pragmatic quality, which needs improvement.

7 Conclusion and discussion

It is worth noting that this research focuses on the design and implementation of a prototype within a simulation environment. Testing has not been done with real-time earthquake data from InaEEWS because the system is not yet fully operational. However, this research has successfully designed and implemented "Awas Guncang", a mobile application prototype that serves as a valid and comprehensive reference model for early warning dissemination of InaEEWS. This prototype demonstrates that modern technology architecture can be leveraged to develop a fast, reliable, and relevant dissemination system for users. Early warning notifications are carried out using the user's PGA algorithm to be on target, which can estimate whether the user will experience shocks. In addition, the application also successfully provides full-screen notifications that give the user the impression of an emergency in an average of 3.3 seconds and 3 seconds median. The design and user experience were successfully developed with a key performance indicator value of KPI = 1.87 in the UEQ survey. This value is dominated by hedonic quality compared to pragmatic, which means that users are more impressed by the emotional experience. Even so, the pragmatic functionality is still in line with expectations but does not rule out the possibility of further development.

This research is not nearly perfect and having some limitations. The app is entirely dependent on the user's internet connectivity, so when the internet is unavailable while the disaster happens, users may not be notified by the app. In the other hand, Android/iOS battery optimization features can sometimes delay background notifications, which is a known challenge for all such apps. Furthermore, this app still has rooms for improvement. The main recommendations focus on improving the accuracy and speed of the system by updating the ground motion prediction equations and the seismic velocity model used. Furthermore, integrating with InaTEWS and adding dynamic evacuation routes could create a comprehensive disaster mitigation system. Finally, to increase social impact and inclusivity, the application must be enriched with disaster education content designed to be friendly to all age groups and improve accessibility features to serve users with special needs.

We extend our gratitude to the Directorate of Engineering Seismology, Potential Geophysics, and Time Signals of BMKG and the Directorate of Network Systems and Communication of BMKG for their support and for providing access to necessary facilities. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The source code for the app is openly available in the GitHub repository at https://github.com/hanjpk/awas-guncang. Further data supporting the findings of this study are available from the authors upon reasonable request. Hanif Kurniawan: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing—Original Draft, Visualization, Project administration. Abraham Frederik Mustamu: Supervision, Writing—Review & Editing. Maulana Putra: Supervision, Writing—Review & Editing. Nardi: Supervision, Methodology. Arya Bratasena Alhaq: Software, Resources. Muhammad Reza July Utama: Methodology. Aurora Fajri Miftakhunnisa: Methodology, Software. Benyamin Heryanto Rusanto: Supervision.

References

1. DataReportal Digital 2024: Indonesia. (2024)
2. S. Widiyantoro, E. Gunawan, A. Muhari, et al., Implications for megathrust earthquakes and tsunamis from seismic gaps south of Java Indonesia. Scientific Reports 2020 10:1. **10**, 1–11 (2020). https://doi.org/10.1038/s41598-020-72142-z
3. P. Supendi, S. Widiyantoro, N. Rawlinson, T. Yatimantoro, et al., On the potential for megathrust earthquakes and tsunamis off the southern coast of West Java and southeast Sumatra, Indonesia. Natural Hazards. **116**, 1315–1328 (2023). https://doi.org/10.1007/S11069-022-05696-Y/FIGURES/6
4. J. Griffin, N. Nguyen, P. Cummins, and A. Cipta, Historical Earthquakes of the Eastern Sunda Arc: Source Mechanisms and Intensity-Based Testing of Indonesia's National Seismic Hazard Assessment. Bulletin of the Seismological Society of America. **109**, 43–65 (2019). https://doi.org/10.1785/0120180085
5. A. Mugesh, A. Desai, R. S. Jakka, and Kamal, Site class based seismic magnitude prediction equations for earthquake early warning. J Seismol. **28**, 765–786 (2024). https://doi.org/10.1007/s10950-024-10213-8
6. R. M. Allen, Q. Kong, and R. Martin-Short, The MyShake Platform: A Global Vision for Earthquake Early Warning. Pure Appl Geophys. **177**, 1699–1712 (2020). https://doi.org/10.1007/S00024-019-02337-7
7. S. Colombelli, F. Carotenuto, L. Elia, and A. Zollo, Design and implementation of a mobile device app for network-based earthquake early warning systems (EEWSs): application to the PRESTo EEWS in southern Italy. Natural Hazards and Earth System Sciences. **20**, 921–931 (2020). https://doi.org/10.5194/NHESS-20-921-2020
8. S. Pramono, M. R. J. Utama, R. Swastikarani, et al., Implementation Early Design of Prototype EEWS Development in Indonesia. Res Sq. (2023). https://doi.org/10.21203/rs.3.rs-2774584/v1
9. A. Rudyanto, A. Wijaya, S. Widiyantoro, D. P. Sahara, S. Rosalia, A. Wibowo, S. Pramono, and A. S. Putra, Performance test of pilot Earthquake Early Warning system in western Java, Indonesia. Int. J. Disaster Risk Reduct. **115**, 105010 (2024). https://doi.org/10.1016/j.ijdrr.2024.105010
10. Earthworm Central Earthworm Central, http://www.earthwormcentral.org/
11. J. P. Y. Tan, M. W. J. Tan, R. M. Towle, et al., mHealth app to facilitate remote care for patients with COVID-19: Rapid development of the DrCovid+ App. JMIR Form Res. 7, e38555 (2023). https://doi.org/10.2196/38555
12. J. X. Zhao, J. Zhang, A. Asano, Y. Ohno, T. Oouchi, T. Takahashi, H. Ogawa, K. Irikura, H. K. Thio, P. G. Somerville, Y. Fukushima, and Y. Fukushima, Attenuation Relations of Strong Ground Motion in Japan Using Site Classification Based on

Predominant Period. Bull. Seismol. Soc. Am. **96**, 898–913 (2006). https://doi.org/10.1785/0120050122

13. T. Y. Hsu and A. Pratomo, Early peak ground acceleration prediction for on-site earthquake early warning using LSTM neural network. Front Earth Sci (Lausanne). **10**, 911947 (2022). https://doi.org/10.3389/FEART.2022.911947
14. B. L. N. Kennett and E. R. Engdahl, Traveltimes for global earthquake location and phase identification. Geophys J Int. **105**, 429–465 (1991). https://doi.org/10.1111/j.1365-246X.1991.tb06724.x