Mobile-based sensor notification application

H A Warandi and P W Wirawan
Computer Science / Informatics Department, Faculty of Science and Mathematics, Diponegoro University, Jl Prof Soedarto SH, Tembalang, Semarang, Indonesia.
E-mail: hafidhwirandi@gmail.com.

Abstract. According to National Census of Fatal Occupational Injuries survey conducted by the U.S. Bureau of Labor Statistic in 2015, there were 4836 workers killed on construction sites due to illness and fatalities, 9% injuries were due to the exposure to the hazardous environment. Wireless Sensor Networks (WSNs) is one of the technologies used to access information from the environment and sent to the computer for analysis, thus reducing exposure to the hazardous environment. However, the state of danger is not yet known to the worker directly. Fortunately a mobile application can overcome this problem by receiving notification from the sensor device. In this paper, we propose a mobile application that can receive notification from the sensor device. We design our application to communicate to the sensor device that connected to the server. Our application can receive notification if there is an alteration from sensor data such as change of normal status to dangerous status or vice versa. Notification delivery was sent by using Firebase Cloud Messaging (FCM) to send notifications to Android. In the implementation, the application can receive notification of change of status sensor data equipped with sound when notification received.

1. Introduction
Geothermal energy is thermal energy originating from within the earth. This energy has a danger like a very high temperature so that it can store the earth's core in a liquid state [1]. Liquids produced by geothermal energy such as oxygen, hydrogen ions, carbon dioxide, hydrogen sulfide, ammonia, chloride ions, and sulfate ions are corrosive [2]. These corrosive substances are harmful to workers in the geothermal industry. According to the National Census of Fatal Occupational Injuries survey conducted by the United States Bureau of Labor Statistics in 2015, 4836 workers were dying in the geothermal industry caused by diseases and disasters, 9% of victims caused by exposure to a dangerous environment and 3% caused by explosion and fire [3].

High-temperature hazards and gas concentrations of these corrosive substances can be detected using Wireless Sensor Networks (WSNs) [3]. WSNs have different detection capabilities and can be used for places that are difficult to reach. The sensor can obtain data from the environment and send it to the computer for analysis. The use of sensors reduces the risk of workers being exposed to hazardous environments.

The utilization of sensors to retrieve environmental data has been used in several studies. Research conducted by Cheung uses sensors to monitor harmful gases, temperature, and humidity in underground construction. The sensor takes data from the environment and sends it to the computer to be displayed visually. If the gas concentration is harmful, then the data is displayed in red so that information can be obtained and can also be used to make decisions to evacuate or not [3]. In this condition, the workers do not know directly about the dangerous condition.
This problem raises an opportunity to develop a mobile application. It is expected that workers can find out the status of sensors that monitor the environment because the data taken by sensors can send notifications when there is a change in data. This is made possible by the notification technology from Firebase Cloud Messaging (FCM). FCM can send notifications with support on many platforms such as Android, iOS, and others [4].

2. Application Design
Safety management system has been the focus of studies by [3]. In their study, building information modeling (BIM) was combined with a wireless sensor network that was able to monitor gas hazards visually. The system has a multi-tiered architecture consisting of sensor layers, database layers, application layer, and presentation layers. The system provides real-time monitoring of the conditions on the sensor sensors installed. However, it does not provide a notification message to mobile devices.

This paper proposes a notification application that can receive notifications from sensor devices connected to the server. The sensors are connected to the microcontroller where every certain period the microcontroller sends data to the server. The server then saves the data to the database for later use. Application users can set the value threshold read by the sensor so that when there is one condition of the data from the sensor that exceeds the threshold, the server can send notifications to the smartphone. Notification delivery is not done directly but through third-party services, namely Firebase Cloud Messaging (FCM). Figure 1 depicts the architecture of the notification system.

![Figure 1. All elements in the notification system.](image)

Applications that are built not only can receive notifications but can also request sensor data to the server with an Application Programming Interface (API). The communication mechanism between Android and the server is the request-reply. Data is requested through the API and to return values. We use the JavaScript Object Notation (JSON) format for data interchange. JSON is chosen because it has several advantages. JSON produces a smaller data stream and provides better performance, readability, flexibility compared to XML [5]. The communication mechanism between the server and smartphone and firebase is illustrated in Figure 2.
3. Implementation

We develop our application as an Android application concerning the design that was discussed previously. We use the KitKat version (API Level 19) because it supports push notification feature. Push notification feature can be realized because of the trigger from Firebase that is requested from the server. For this reason, a server-side program was needed to send a request to Firebase. The server-side program is written using Python programming with the pyfcm library, an open source library for Firebase Cloud Messaging.

Figure 3 shows the server-side program to make notification requests. We use the FCMNotification class library to send notifications to the server. This class requires the KEY API for the validity of using Firebase. Data that is sent to Firebase, to be pushed to an Android device, is data related to sensors such as sensor status and the time when data is obtained. The server triggers the Firebase in JSON format because JSON is lightweight and can be understood by Firebase.

```python
def trigger(user_uuid, device_uuid, device_name, user_sensor_uuid, user_sensor_name, status):
    API_KEY = "...
    push_service = FCMNotification(api_key=API_KEY)
    data_message = {
        "data": {
            "user_uuid": user_uuid,
            "device_uuid": device_uuid,
            "device_name": device_name,
            "user_sensor_uuid": user_sensor_uuid,
            "user_sensor_name": user_sensor_name,
            "status": status,
            "timestamp": time.time()
        }
    }
    result = push_service.notify_topic_subscribers(topic_name=user_uuid,
                                                   data_message=data_message)
```

Figure 3. A server program that sends notification trigger to Firebase.

The Android application will receive notification data after the application starts and runs in the background. Notification data can be received by an onMessageReceived callback method defined in the class that handles Firebase Messaging Service. When the data is received, the onMessageReceived method will run a custom method, namely handleDataMessage, which will process the incoming data and issue an alarm sound along with a notification message "sensor above threshold!". The recipient program on Android is depicted in figure 3.
public void onMessageReceived(RemoteMessage remoteMessage) {
    if (remoteMessage.getData().size() > 0) {
        try {
            JSONObject json = new JSONObject(remoteMessage.getData().toString());
            if (MySharedPreferences.getBooleanPreference(getApplicationContext(), Config.LOGIN_PREF)) {
                handleDataMessage(json);
            }
        } . . . . . .
    }
}

private void handleDataMessage(JSONObject jsonObject) throws JSONException {
    JSONObject data = jsonObject.getJSONObject("data");
    . . . . . .
    NotificationUtil.customNotification(getApplicationContext(), alarmSound, deviceId, sensorName, "sensor above threshold!");
    . . . . . .
}

Figure 4. Recipient program for Android smartphone.

When the notification is received, the user can see the notification details in the form of the sensor name, time, and sensor location, (figure 5). Users can set the threshold so that the threshold can be set if there is a standard change (figure 6). Another advantage of the program developed was the ability to monitor the state of each sensor connected to the server (figure 7).

Figure 5. Detail warning for Smartphone.  Figure 6. The user interface to set the sensor threshold  Figure 7. User interface To monitor sensor that is connected to the server

4. Application Testing
Application testing is performed to the aim of evaluating system functionality from the user’s point of view. This test ensures that users can use the application compatible with the purpose of constructing the application itself. The next test is testing the application's ability to send notifications to multiple devices at once. This test aims to test the reliability of an application when it runs on many mobile devices.

Application functionality evaluation is performed by using the black box method with several tests for application feature, described in table 1. The testing theme is prepared by considering various
things, both regarding application and supporting functionality such as availability of internet on mobile devices. The definition of a testing scenario related to the internet is considered important because the device that runs the application is a mobile device, which has mobile characteristics. These characteristics allow the absence of internet connection at the time when mobile.

The evaluation was carried out by involving two end users in the Geothermal Laboratory of the Faculty of Science and Mathematics Diponegoro University. Our application testing was conducted by simulating using Advanced Rest Client (ARC) tools to simulate sensor data that exceeds the threshold. The test results show that the user can receive all application features and use them according to the application's purpose. Both users can receive and flow application flow by the purpose of constructing application itself.

Another test is testing the sending of notifications to several Android devices simultaneously. The Android devices used for this test are ASUS Zenfone 4C, Samsung J3, Samsung J7 Prime, Redmi 4A and Redmi 3S. Each device runs an application that is developed. Observations are made when the sensor is in good condition, and the condition of the observation value exceeds the threshold.

Table 1. Application features to be tested to users.

| Feature Number | Feature to Test                                                      |
|----------------|----------------------------------------------------------------------|
| 1.             | Change the notification settings to on.                             |
| 2.             | Change the notification settings to off.                            |
| 3.             | Displaying application detail of warning records from devices.     |
| 4.             | Displaying a list of 5 warnings of changing the status of the last sensor data from all sensors. |
| 5.             | Displaying an internet connection message does not exist if the device is not connected to the internet. |
| 6.             | Display a map of the location of the tool                           |
| 7.             | Displaying data messages does not exist when sensor data is missing.|

5. Conclusion

This paper has proposed a mobile application that can receive notification status from sensors that connected to the server. The application can work because of the push notification communication mechanism from Firebase servers. The use of FCM is beneficial in that it supports real-time database dan all related notification libraries. The application has been tested, and the functionality is acceptable. In testing sessions with many devices, the developed application can receive notifications from the server at the same time because of the real-time support of FCM. For further research, this notification system has to be tested or simulated if there are multi-sensor attached to it for observation of application performance.

References
[1] Towler B F 2014 G The Future Energy (Australia: University of Queensland) p 237
[2] Nogara J and Zarrouk S J 2018 Renewable and Sustainable Energy Reviews 82 1333
[3] Cheung W -F, Lin T -H and Lin Y -C 2018 Sensors 18 436
[4] Srivastava N, Shree U, Chauhan N R and Tiwari D K 2017 International Journal of Innovative Research in Science, Engineering and Technology 16 11
[5] Zunke S and D’Souza V 2014 International Journal of Computer Science and Network 3 257