Proceedings of the 3rd Asia Pacific International Conference on Industrial Engineering and Operations Management, Johor Bahru, Malaysia, September 13-15, 2022

The Use of Synchronization Technology on The Effort of Minimizing Medication Non-Adherence

Alfan Faturahman Gifary
Department of Informatics
Universitas Jenderal Achmad Yani
West Java, Cimahi, Indonesia
alfanfgifary18@if.unjani.ac.id

Faiza Renaldi S.T., M.Sc, Irma Santikarama S.T., M.T
Department of Information Systems
Universitas Jenderal Achmad Yani
West Java, Cimahi, Indonesia
faiza.renaldi@unjani.ac.id, irma.santikarama@lecture.unjani.ac.id

Abstract

Medication non-adherence is a health problem often found in the global community. According to Lars Osterberg, M.D, about 30% of adult patients in America fail to take their medication at the appointed time, causing an estimated $100 billion in possible costs each year. Along with the development of information technology, interventions have been carried out on non-adherence to medication in patients. Previous research developed an automatic treatment application by sending notifications in email and SMS messages on time according to doctor's recommendations; Unfortunately, now SMS messages have almost been replaced with chat application services. This study aims to overcome the focus by performing automatic reminders using cloud computing technology with job scheduling sent via cellular notifications, email, and WhatsApp so that treatment reminders will be delivered faster using cloud computing. This research begins by designing an auto-reminder function. After the design is complete, it is continued with the implementation phase of the drug reminder application. In the testing phase, ten notifications were sent via WhatsApp, cellular, and email, received by 30 respondents, and sent simultaneously on each media with an average time interval of approximately 2 seconds, 7 seconds, and 18 seconds, respectively. The test concluded that all notifications sent with the last ten users experience a delay of approximately 18 seconds for each notification media. In future research, it is recommended to implement this notification technology on Wearable Devices so that it is expected to make it easier for patients to receive notifications during their treatment period.

Keywords
Medication non-adherence, job scheduling, auto-reminder, notifications

1. Introduction

Medication adherence is the process by which patients follow their medication as prescribed, consisting of 3 main components: initiation, implementation, and discontinuation (Walsh et al, 2019). Many factors contribute to medication non-adherence, one of which is the individual level of the patient (e.g., low health literacy) (Zullig et al, 2017). The human factor is one of the causes of medical errors, an exciting topic in patient safety. It is understood that the human condition has poor memory, false senses, and poor response to physiological stress (Robson, 2013). In general, there are four categories of medical errors. One of these categories is Preventive Error; Preventive Error can come from insufficient supervision or lack of follow-up care (Al-Assaf et al, 2003). As a medical human being, mistakes are a huge problem; therefore, even though the costs incurred are significant, it is necessary to avoid the injury and death of many people (Al-Assaf et al, 2003). The forms of prevention that have been used include medication reminder applications. Today's life is full of responsibility and stress. So, people are prone to various diseases, and we must keep ourselves fit and healthy. If the patient stays at home, he may ask someone to take care of him, but when someone is not at home, is out of town, or is away from home, it is difficult for family members to remind them. They are timing their doses each time (Ameta et al, 2015).

The automatic reminder mobile application for taking medication synchronized with email and chat applications will significantly assist patients in carrying out their on-the-go treatment, especially for someone who uses email for application and chat users. Comparing chat app usage, in this case, shows that WhatsApp is used significantly more often than SMS. 85.17% of participants used WhatsApp at least once or twice a day, while only 6.69% used SMS so frequently. It leads to the conclusion that WhatsApp communication is preferred and significantly more frequent than SMS communication by the participants (Seufert et al, 2016). This survey is the reference for sending notifications via chat applications rather than SMS. Notification of implementation through a notification has been applied in previous research. Implementation is carried out using real-time algorithms, Calendar API (Zao et al, 2010), and Application Development Framework (ADF) (Ameta et al, 2015). In this study, the implementation was carried out by running job scheduling on a cloud server to push notifications for automatic medication reminders.

2. Methods

This research was carried out in three stages to achieve the research objectives. The research method in this study can be seen in Figure 1.

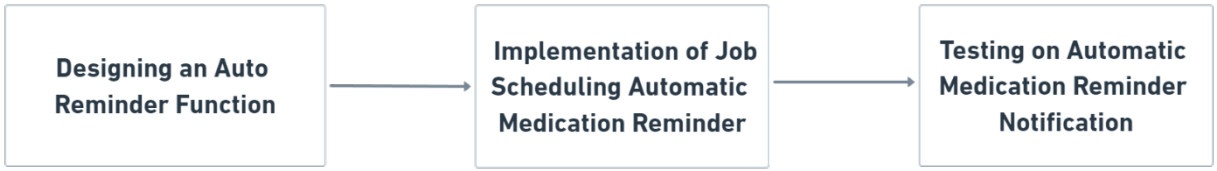

Figure 1 Research method of medicine reminder

The first stage is to design an automatic reminder function to determine the steps of the job scheduling implementation used to schedule the delivery of three notifications. Followed by the second stage, namely the implementation of Job Scheduling automatic medication reminders, the third phase is testing on automatic medication reminder notifications.

2.1. Designing an Auto Reminder Function

Functional modeling makes design problems modular and structured (Shan and Li, 2009). Therefore, the design of the automatic reminder function is carried out according to the needs of outpatients who need 24-hour attention to carry out their treatment. The automatic reminder function is designed to be able to send mobile, email and WhatsApp notifications according to patient needs which can be done more than once in 24 hours.

2.2. Implementation of Job Scheduling Automatic Medication Reminder

Automating automatic medication reminders is carried out on mobile applications and backend services. The mobile app is built using React Native, and the backend uses Node.js with a MongoDB database. In the mobile application, there is a function to input the patient's outpatient treatment schedule, which will be stored in the database as a collection of patient scheduling data, executed by the Job Scheduler. The Job Scheduler is a core component of the scheduling system in the Cloud, and This decides the execution of tasks waiting in the Job Queue as a First-In-First-Out (FIFO) order (Yang et al, 2011).

Job Scheduling is implemented in the backend, running as a background process to provide automatic medication reminder notifications. Job Scheduling is used to check periodically in one minute, and this is done so that orders to compare scheduling times can be carried out without any missed time. Four tasks are run on job scheduling; the first job is scheduled email, namely, checking the current time and scheduling time for taking medication and sending notifications via email that will run periodically in one minute. The second job is scheduled to send WhatsApp notifications, namely, checking the time and sending notifications via WhatsApp. The third job is to subtract days, namely, reducing the interval of taking medication which will be executed periodically at 00:00 and run 1x24 hours. The fourth job is untaken medicine, which is to make a re-notification within fifteen minutes after taking the drug three times. Periodically.

2.3. Testing on Automatic Medication Reminder Notification

In the testing stage on automatic medication reminder notifications, several test cases were carried out, namely measurements of the speed of time for automatic drug notification requests and testing on CPU process resources load on a cloud virtual machine server.

2.4.1. Response Time Measurement

This measurement is carried out to see how far and long three notifications (email, mobile notification, and WhatsApp) are sent to users of the automatic medication reminder application. This measurement is carried out by sending notifications to the user's device that has been registered with the application and is carried out repeatedly to determine the reliability of notifications.

2.4.2. CPU Resource Load Measurement

Measurement of CPU resource process load is carried out to determine the percentage of CPU process load used when testing notification delivery which aims to test the reliability of notifications.

3. Result and Discussion

3.1. Implementation of Medication Reminder

The implementation of automatic medication reminders is carried out on mobile applications and backend services. The mobile app is built using React Native, and the backend is built using Node.js with a MongoDB database.

3.1.1. Implementation of Job Scheduling Automatic Medication Reminder

Job Scheduling is implemented on the backend using Node.js with a job scheduler, carried out on a Cloud Virtual Machine with specifications vCPU 2 core one shared core, 1 GB RAM as a background process to provide automatic medication reminder notifications. Job Scheduling is used to check scheduling periodically within one minute, and this is done so that commands to compare scheduling times can be carried out without missing any time. Four tasks are executed on Job Scheduling. The first job is scheduling email notifications, namely checking the current time, scheduling medication time, and sending notifications via email, which will run periodically for one minute. The second job is scheduling WhatsApp notifications, checking the date and time, and sending notifications via WhatsApp. The third job is Mobile Notification scheduling which checks the date and time and sends notifications via mobile notifications. The fourth job is untaken medicine reminder, which is to make a re-notification within fifteen minutes after taking medicine three times regularly. Implementation of job scheduling on the backend can be seen in Figure 2

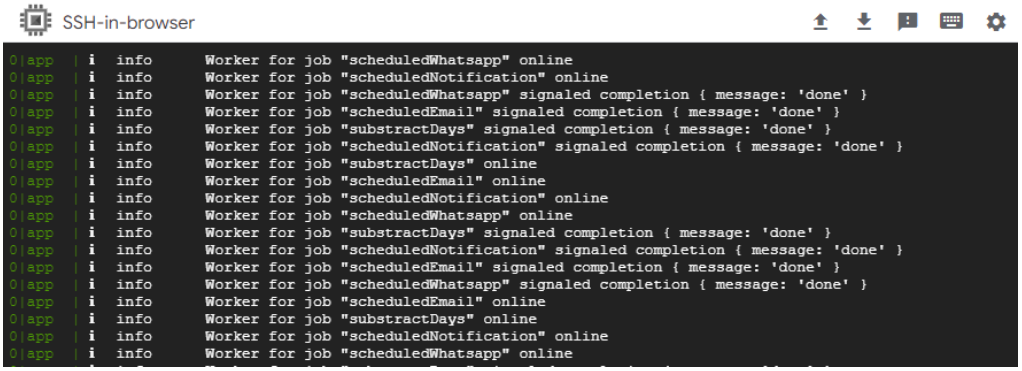

Figure 2 Backend Background Process

The logs in Figure 2 are job scheduling processes that run on the server; each job will provide information when the scheduling process is started, or processes have been completed. The "online" sign means the process is being carried out; the "done" sign means the process has been completed. Notifications sent via email will be sent via the SMTP server. (Kambourakis et al, 2020), notifications on WhatsApp are sent via API Client, and mobile notifications will be sent with Firebase Cloud Messaging (FCM) (Sanmorino and Fajri, 2018).

3.1.2. Implementation of Automatic Medication Reminder on Mobile Application

The mobile application for the client is implemented to connect the client and automatic reminders on the backend; the client performs automatic reminder settings on the scheduling function. The client sends a POST request to save the new scheduling data using the REST API. In the development of the mobile application, FCM token renewal is also carried out when authenticating logins for the need for sending mobile notifications. The results of implementing the automatic medication reminder mobile application can be seen in Figure 3.

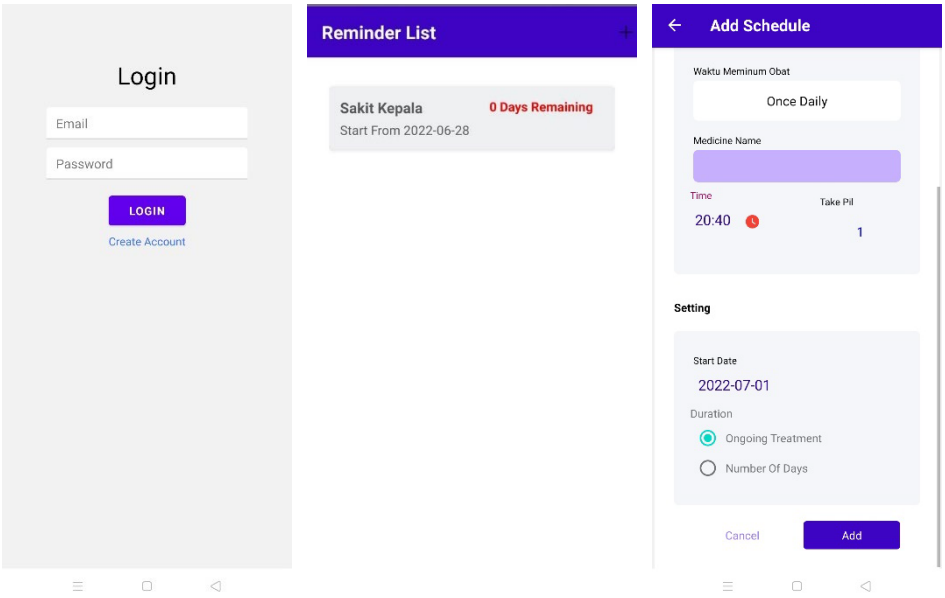

Figure 3 Implementation of automatic medication reminder mobile application

The first stage that users need to do is authenticate so that the FCM token is updated in the database; FCM also ensures that push notifications can be done safely (Sanmorino and Fajri, 2018). then, the user makes scheduling arrangements for taking medication by entering scheduling data. Notifications will be received immediately after the scheduled time according to the previously scheduled time; the following is the display of the notification sent, which can be seen in Figure 4.

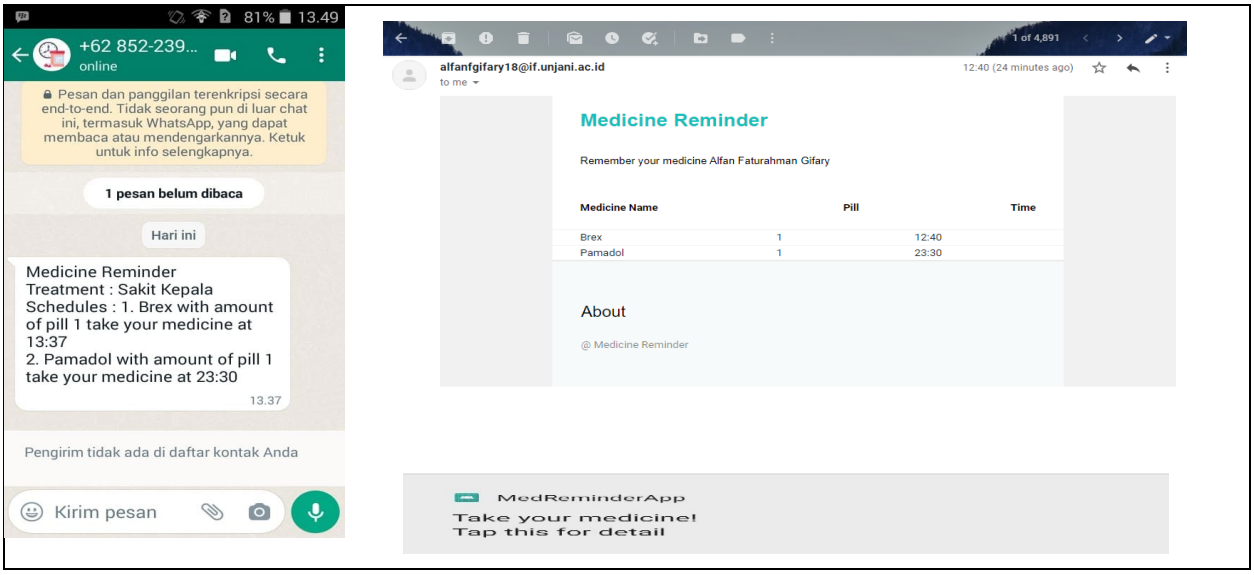

Figure 4 Medicine Reminder Notifications

Users receive notifications through three media (WhatsApp, Email, and mobile notification) based on the schedule made; the notification contains the schedule for taking the medicine that must be taken and the number of pills.

3.2. Testing on Automatic Medication Reminder Notifications

Automated medication reminder notifications are tested to determine the load of resources spent when sending many notifications on a virtual machine and to find out the response time for sending notifications in order to find out how fast the delivery is via three notification delivery media. The test was carried out by making a notification sending request which was carried out ten times to 30 respondents and sent simultaneously via email, WhatsApp, and mobile notifications. There are several stages of testing carried out, the first stage is scheduling data collection, and the second stage is testing the delivery of notifications simultaneously.

3.2.1. Scheduling Data Collection

To perform notification testing, it takes 30 data in the scheduling collection, all scheduling time data is set at the same time to perform notification testing.

3.2.2. Notification Test with Postman

After all, data is collected in the database, and testing is carried out using the Postman platform to test the response time for each notification request. Metrics can be seen in Figure 5.

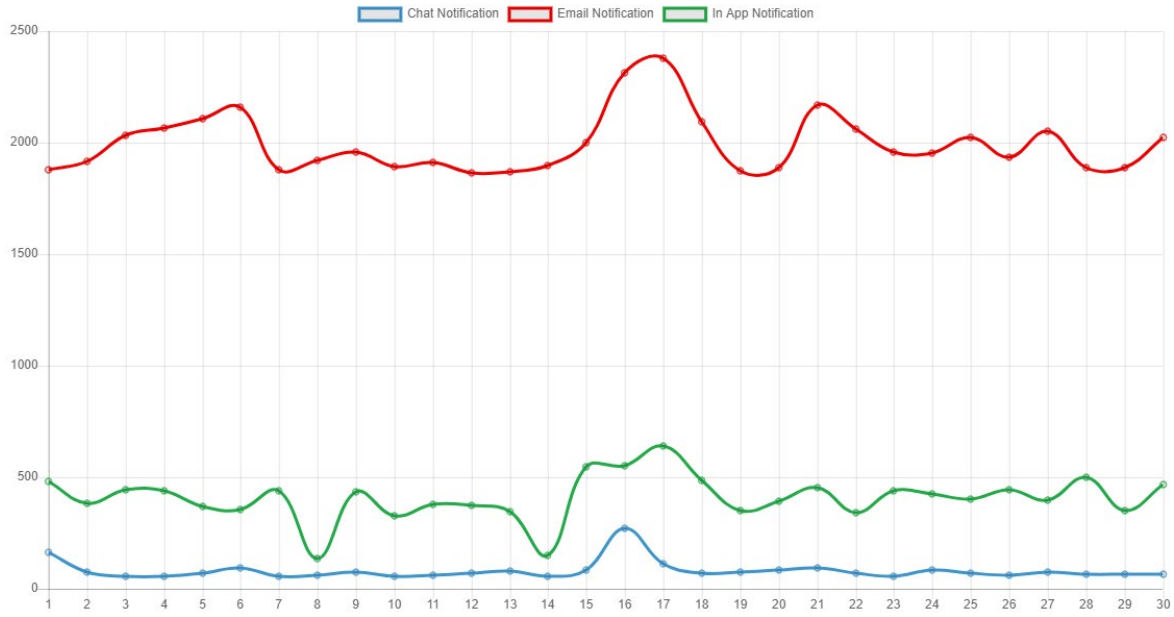

Figure 5 Response Time Metrics with Postman

Notification requests sent include email, mobile notifications, and WhatsApp notifications. The results of the tests conducted by Postman concluded that all notifications were sent in a total of 74 seconds, with an average response time of 475 MS for mobile notifications, 114.5 MS for WhatsApp notifications, and 1951 MS for email notifications. Metrics can be seen in Figure 5.

3.2.3. Notification Test with Google Trace

Tests using Google Trace API requests are carried out directly on the backend, in contrast to Postman, which makes requests on the Postman platform. At the time specified in the scheduling data, Job Scheduling will do its work by executing the request for sending notifications. The test using tracer can be seen in Figure 6.

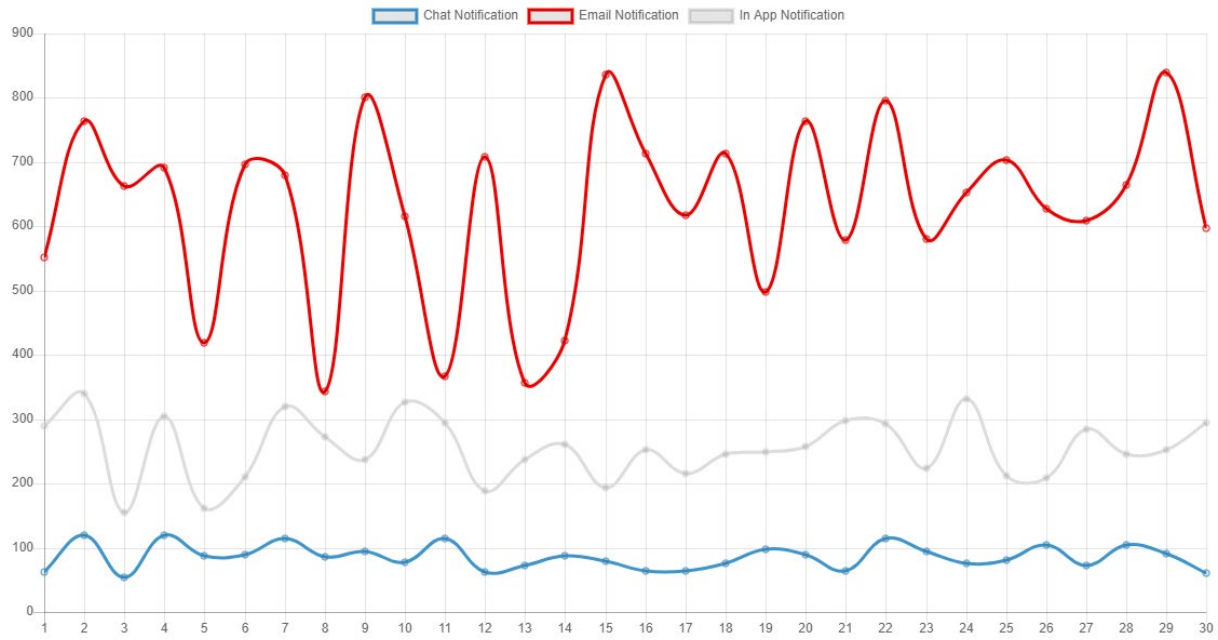

Figure 6 Response Time Metrics with Tracer

The test results concluded that notifications were sent for 18 seconds with an average response time of 256 ms for mobile notifications, 89 ms for WhatsApp notifications, and 671 ms for email notifications. All notifications sent with multiple notifications have a delay of 18 seconds in the last ten notification queues.

Table 1 Response Time Comparison

Media Type	Postman	Google Trace
WhatsApp	114.5 ms	89 ms
Mobile Notification	475 ms	256 ms
Email	1951 ms	671 ms

As in Table 1, Tests with Google Trace are faster than Postman tests because requests are made directly on the backend run by a cloud virtual machine.

3.2.4. Virtual Machine Resource Load

Checking virtual machine resources is done to find out how many resources are used to do push notifications with job scheduling techniques. The percentage of resources can be seen in Figure 7 and igure 8.

PID	Command Line	CPU %
2663	node /home/alfanfaturahman10/med-reminder-backend/app	5.54%
2729	/home/alfanfaturahman10/med-reminder-backend/node_modules/puppeteer/.l...	0.38%
2699	/home/alfanfaturahman10/med-reminder-backend/node_modules/puppeteer/.l...	0.32%
493	/opt/google-cloud-ops-agent/subagents/opentelemetry-collector/otelopscol --c...	0.12%
492	/opt/google-cloud-ops-agent/subagents/fluent-bit/bin/fluent-bit --config /run/g...	0.03%
2752	/home/alfanfaturahman10/med-reminder-backend/node_modules/puppeteer/.l...	0.03%
816	PM2 v5.2.0: God Daemon (/home/alfanfaturahman10/.p	0.03%
415	/usr/bin/google_guest_agent	0.02%

Figure 7 CPU Process on Virtual Machine

CPU resources when running the scheduling process using job scheduling are at ±5% CPU usage, and the percentage of CPU usage can drop to ±4%.

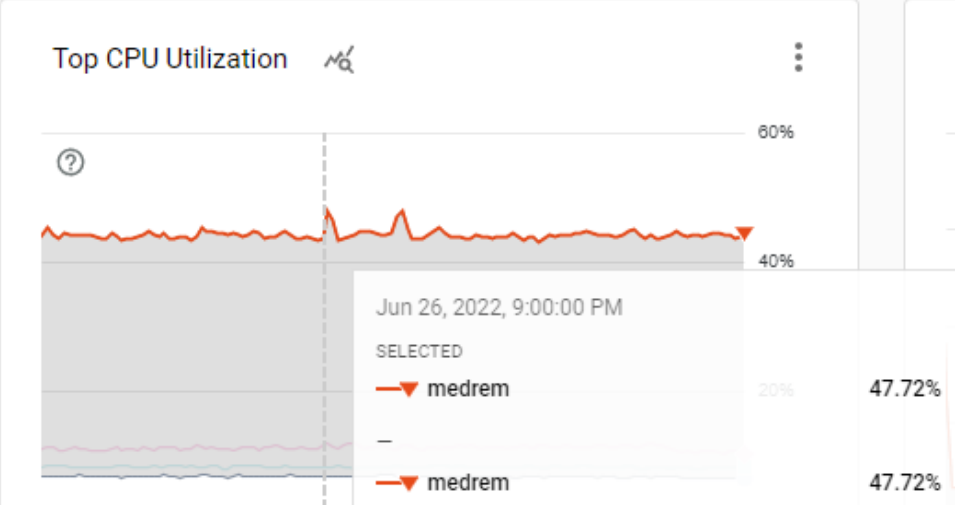

Figure 8 CPU Utilization on Virtual Machine

When testing the delivery of 30 notifications, CPU resources increased by approximately ±4% from 43%. Metrics can be seen in Figure 8.

4. Conclusion

This study focuses on implementing and testing the automatic medication reminder notification function, which is implemented on a mobile app and backend using job scheduling techniques on a cloud virtual machine. Notifications are sent via mobile notifications, email, and WhatsApp. In this study, the test results conclude that the experiment of sending ten notifications sent to 30 respondents received all notifications and experienced a delay of 18 seconds for the last ten users for each notification media. The Job Scheduling process on the backend does not take up a significant CPU load because it only consumes ±5% of the CPU processing on the virtual machine used in this study.

References

Al-Assaf, A. F., et al. *Preventing Errors in Healthcare: A Call for Action.* 2003.

Ameta, Deepti, et al. "Medication Reminder and Healthcare – An Android Application." *International Journal of Managing Public Sector Information and Communication Technologies*, vol. 6, no. 2, 2015, pp. 39–48, doi:10.5121/ijmpict.2015.6204.

Kambourakis, Georgios, et al. *What Email Servers Can Tell to Johnny: An Empirical Study of Provider-to-Provider Email Security*. 2020.

Robson, Wayne. *Prescribing Errors: Taking the Human Factor into Account*. 2013.

Sanmorino, Ahmad, and Ricky Maulana Fajri. *The Design of Notification System on Android Smartphone for Academic Announcement*. 2018.

Seufert, Michael, et al. *Group-Based Communication in WhatsApp*. 2016.

Shan, Hongbo, and Shuxia Li. *Function Principle Structure Model for Conceptual Design*. 2009.

Walsh, Caroline A., et al. "The Association between Medication Non-Adherence and Adverse Health Outcomes in Ageing Populations: A Systematic Review and Meta-Analysis." *British Journal of Clinical Pharmacology*, 2019.

Yang, Bo, et al. "An Utility-Based Job Scheduling Algorithm for Cloud Computing Considering Reliability Factor." *Proceedings - 2011 International Conference on Cloud and Service Computing, CSC 2011*, 2011.

Zao, John K., et al. "Smart Phone Based Medicine In-Take Scheduler, Reminder and Monitor." *12th IEEE International Conference on E-Health Networking, Application and Services, Healthcom 2010*, 2010.

Zullig, Leah L., et al. "Medication Adherence: A Practical Measurement Selection Guide Using Case Studies." *Patient Education and Counseling*, 2017.

Biographies

Alfan Faturahman Gifary is a bachelor's degree student at the Universitas Jenderal Achmad Yani, West Java Indonesia, and joined in informatics in 2018. His research interests are in the fields of information systems, web development, and Cloud Computing.

Faiza Renaldi, S.T., M.Sc is a lecturer in the department of Information Systems, Faculty of Science and Informatics, Universitas Jenderal Achmad Yani Indonesia. He received his master of business informatics at Universiteit Utrecht, The Netherlands in 2006. His research interests are related to health informatics, information systems/information technology management, e-government, agile project management, and IT entrepreneurship.

Irma Santikarama, S.T., M.T received a Bachelor's degree in Information System from Universitas Kristen Maranatha and Master's degree in Informatics from Institut Teknologi Bandung. Now, she is a lecture of Information Systems Department, Faculty of Science and Informatics, Universitas Jenderal Achmad Yani. Her interest area related to information system, eGoverment, and agile project management.