

Design and Implementation of Telegram Bot for Integrated Hospital Information System

Oka Sudana^{1*}, Ary Paramartha², Ayu Wirdiani³, Dwi Rusjyanthi⁴ 

^{1,2,3,4} Department of Information Technology, Udayana University, Badung, Indonesia

ARTICLE INFO

Article history:

Received November 17, 2021

Revised November 20, 2021

Accepted March 14, 2022

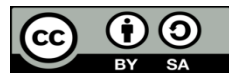
Available online April 25, 2022

Kata Kunci:

Sistem Informasi Terintegrasi, Telegram Bot, UAT, Rumah Sakit

Keywords:

Integrated Information System, Telegram Bot, UAT, Hospital



This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

Copyright © 2022 by Author. Published by Universitas Pendidikan Ganesha.

ABSTRAK

Masih banyak rumah sakit umum belum memanfaatkan telegram bots API. Telegram Bots memungkinkan multi-channel access untuk memudahkan akses data yang dimiliki oleh sistem informasi. Setiap bagian rumah sakit membutuhkan modul yang berbeda. Modul perlu diintegrasikan agar aliran pertukaran data menjadi lebih mudah. Penggunaan internet messenger seperti telegram dapat mempermudah proses integrasi yang dibutuhkan, sehingga pengguna dapat dengan mudah mendapatkan informasi dari sistem. Tujuan penelitian ini yaitu untuk mengembangkan sistem informasi rumah sakit yang terintegrasi dengan bot telegram. Pengujian sistem dilakukan di laboratorium oleh 30 pengguna sebagai pasien dan satu pengguna sebagai administrator. Pengujian sistem menggunakan metode black box dengan fokus pada input, fungsionalitas, dan output untuk semua proses antrian. Bot telegram ini menggunakan bantuan cronjobs dan webhooks untuk mengambil informasi dan menjalankan perintah pengiriman pesan untuk bot telegram. Hasil yang ditunjukkan pada penelitian ini adalah telegram bot yang dirancang untuk diuji menggunakan user acceptance test (UAT) dengan hasil respon yang sangat positif dan dianggap berhasil. Bot telegram ini memfasilitasi pasien dan pekerja rumah sakit untuk mendapatkan informasi dengan segera.

ABSTRACT

There are still many public hospitals that have not taken advantage of the Telegram bots API. Telegram Bots support Multi-Channel Access to facilitate access to data owned by information systems. Each part of the hospital requires a different module. Modules need to be integrated to make the flow of exchange data easier. The use of internet messengers such as Telegram can simplify the integration process needed so that users can quickly get information from the system. This study aims to develop a hospital information system that is integrated with the Telegram Bot. System testing was carried out in the laboratory by 30 patients and one user as an administrator. System testing using the black box method focuses on input, functionality, and output for all queuing processes. This Telegram Bot uses the help of Cronjobs and Webhooks to retrieve information and execute messaging commands for the Telegram Bot. The results shown in this study are Telegram Bot which is designed to be tested using the User Acceptance Test (UAT) with very positive responses and is considered successful. This Telegram bot facilitates patients and hospital workers to get information immediately.

1. Introduction

Currently, many hospitals have implemented a modern hospital management system. Technological developments can help reduce clinical errors, increase privacy and security and improve disease surveillance (Kania-Richmond et al., 2015; Pantartzis et al., 2017). Hospital management information systems can provide benefits in the form of an efficient process, can improve administration and control, superior care, tight cost control and increase profitability (Balaraman & Kosalram, 2013; Ran et al., 2020). All business processes that occur today have been carried out automatically, but the system is only managed by each existing module and there is a lack of evaluation carried out on the existing system by the hospital (Hari Setiaji & Wahid, 2015). The hospital is a vital organization with various information that needs to manage from multiple departments (He et al., 2021; Kosherbayeva et al., 2020). This information has numerous forms and formats. There are differences in access rights to this information, making the design of information systems require a different form (Çetin et al., 2021; Hanlin et al., 2022). If users are given the same information system with the same access rights, in that case, there is a possibility that users such as cleaners can view patient data, which is very confidential and can only be seen by doctors, nurses, or patients. Differences in information systems by department and module can help distribute access rights and information that users should obtain (Rochmah et al., 2020; Wang et al., 2016). However, users with multiple access rights who need direct and accurate information will need time to view the information.

The gap analysis of this research that in general hospitals have not utilized the Telegram bots API. Telegram Bots allows for Multi-Channel Access to facilitate access to data that owned by the information system. The use of internet messenger is straightforward and has widely used on smartphones. Based on

*Corresponding author.

E-mail addresses: agungokas@unud.ac.id (Oka Sudana)

research, Telegram is a multi-platform application that runs on Android, iOS, Windows, and other operating systems (Fernández-Álvarez & Rodríguez, 2022; H Setiaji & Papatungan, 2018). It has accessible features to use and design to be very light, fast, and reliable. Telegram has developed an additional feature, namely bots that can interact with users using commands (Sutikno et al., 2016). Telegram Bot assistance does not only facilitate the provision of information but also in processing it (Gregorio et al., 2017; Nobari et al., 2021). Previous research is about queue untethering based on multichannel access technology in the radiology department of a hospital using the Telegram Bot API to develop bots with multichannel access queuing technology. This technology can help patients queue virtually and save their queue number via Telegram Messenger (Dyatmika et al., 2019).

Another research uses the Telegram Bot API as a tool to create a Telegram Bot on Campus Information Sharing (Setiaji & Papatungan, 2018). Communication between the system and Telegram Bot is done via a webhook available for Telegram using the Telegram Bot API. Webhook offers a way for Telegram Bot to get the communication process without latency. Telegram Bot in this research is also can work through commands given by users. Another research is building a train chatbot to help users get information about train status, seats, etc. The Telegram Bot developed in this study used to share information and make it easier for train passengers to get information by simply accessing the Telegram Bot without opening the train website (Sahaya Sakila V et al., 2018). Telegram Bot API is used to access information, such as notifications, history, books borrowing history, book return history, book catalogs, and book orders. Users can use commands to get information from Telegram Bot (Arimbawa P et al., 2018). API or Application Programming Interface is a set of rules that can be followed and used by software programs as access to resources provided by specific other software programs that implement the API (Nayagam & Arun, 2014; Ogorodnyk et al., 2021). Thus, the use of API defines how programmers will take advantage of certain parts and features of an application (Krishna & Sharma, 2019; Liu et al., 2022). Based on their website at core.telegram.org, Telegram has two types of APIs available for software developers to use. These are Bot API and TDLib. Bot API allows people to build chatbots in the Telegram app and TDLib to build customizations in Telegram. Telegram Bot API is an HTTP-based interface created for developers who want to build bots for Telegram.

Other studies have tried to use instant message technology as a means of academic service information. This information service application is built with the Rational Unified Process (RUP) process model, the Forward Chaining method, and the Python Telepot Framework for the Telegram Bot API to run applications via Telegram instant messages (Rianto et al., 2019). Another research built a chatbot based on Telegram Bot to provide information and academic services at the Department of Informatics Engineering, University of Mataram. Telegram offers an API that can use to develop bots. The bot is built using Python, SQLite, and React JS. The prototype model is used as a development method. The bot prototype can broadcast the latest information to its registered users and provide academic services such as thesis and internship services (Zubaidi & Ramdani, 2019). However, the reason for using Telegram over many other instant messaging platforms for this research is its easy-to-use, open source-based, and easy-to-use API platform, which is available to access via their website at telegram.org. Based on the above study, the purpose of this research is to develop a hospital information system that is integrated with the Telegram Bot.

2. Method

Hospitals use information systems to transmit information in real-time and directly to employees and patients for hospitals. The information system generates information for the designed Telegram Bot using a database-level integration method. This Telegram Bot was designed to be used by two types of users, namely workers and patients. Hospital workers can use Telegram Bots as a tool to help provide information obtained from the integrated information system in the form of notification messages. For example, it can be a notification message when a new patient registers or a report that requires approval. Telegram Bot on the patient's side can assist in patient registration only by using the patient ID number obtained when the patient previously had registered at the hospital for the first time. Patients only need to enter an order to register, and then patients can enter their patient number, their health complaint, and the type of allergy they have. Thereby, the patient can register for further examination at the hospital. Figure 1 can illustrate these processes.

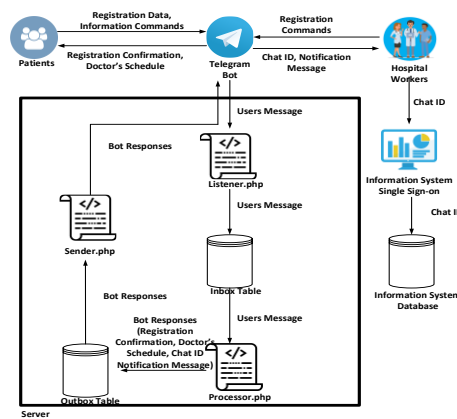


Figure 1. Telegram Bot Overview

As previously explained, this bot has two types of users, and each user has different access rights for the bot. There is a requirement to enter the single sign-on information system page to use the Telegram Bot fully on the hospital worker's side. This bot was run by three different scripts that run on the server-side of the system. The reason for this splitting of scripts is that if one of the three scripts fails, the others will still work, and the bot will not close. This bot was run using a webhook, which means the bot will wait for an event to occur to respond to it. Script listener.php is set up for webhook. If any event occurs in a message form, listener.php will save the message to a table in the database called the inbox table. Script of processor.php looped the contents of the inbox table. Then, it got the messages that were recently saved and had not been processed. Processor.php proceeded to provide the required response and keep the response to outbox.php. It is where sender.php came in and sent responses to users using saved responses and Chat ID. System testing was carried out in the laboratory by 30 users as patients and one user as administrator. System testing used the black box method, focusing on input, functionality, and output for all queuing processes. This test provided a conclusion to what extent the application can solve existing problems. Based on the Telegram Bot overview mentioned previously, the use case shows in Figure 3. This Telegram Bot has two types of users, and each has a different command for each procedure used.

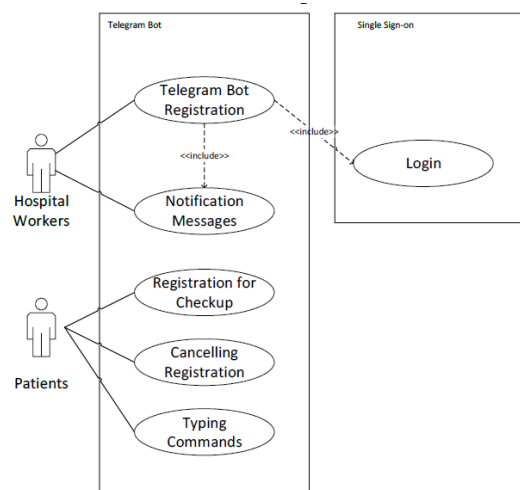


Figure 1. Telegram Bot Use Case

Hospital workers must register in advance to be able to use the Telegram Bot. First, workers have to type a command to get their Chat ID. Then, they need to log in to the hospital information system Single Sign-on feature to save their Chat ID and receive notification via telegram conversation with the bot. Patients who want to use the bot do not need to register with the bot. Instead, patients only need to use different commands for different purposes, such as registering for a health examination, getting information about the doctor's schedule, and canceling registration. Figure 3 is a flowchart for the user connect using the Telegram bot. The user start by entering a command which will bring up a list of commands that the user can choose, example for the patient to registration. The registration process can be carried out if the user successfully enters the data according to the required format, if the format does not match the user must enter it correctly

until it can finally be entered into the database. Information will be sent to user according to the command they enter.

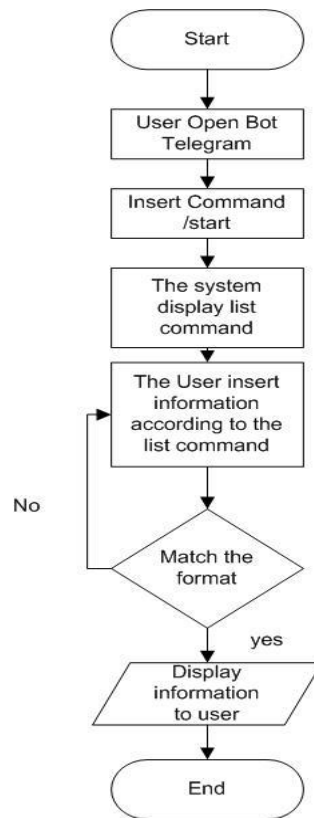


Figure 3. Telegram Bot Flowchart

3. Result and Discussion

Results

The integrated information system is a web-based system consisting of several different integrated modules through the database level that will use to fill in the information on the use of Telegram Bot. There are seven modules for various parts of the information system with different purposes and information types but are still connected through the database and have a Single Sign-on feature for ease of use. The different modules are divided into several sections, such as accounting, pharmacy, front office, HRD, service, facilities, and infrastructure modules. These modules are standard information system modules for hospital information systems. Then, the system provides information to notify this Telegram Bot and receives information from patients who register for examinations. The designed Telegram Bot has several commands that users can use. These commands describe in Table 1.

Table 1. Telegram Bot Commands

Commands	Functions	User Types
/start	To start a conversation.	Patient
/infodokter	To show the doctor's schedule.	Patient
/daftar	To show the command to register the preliminary examinations.	Patient
/umum	To register for a preliminary examination without insurance.	Patient
/jaminan	To register for a preliminary examination with the insurance.	Patient
/batal	To cancel an examination registration request.	Patient
/register	To get Chat ID to place on profile page in Single Sign-on dashboard.	Patient

Table 1 shows the commands that Telegram Bot users can use. Each command has a different function for different uses and user types. For example, another use for Telegram Bot is notifications which

registered hospital worker users can only use. The types of notifications that users can get can see in Table 2.

Table 2. Types of Notifications

Notifications	Module From
Registered New Patient	Service
Needs Approval Form	HRD
Medicine Out Of Stock	Pharmacy
Payment	Accountancy
Room Cleaning	Facilities and Infrastructure

Table 2 shows each type of notification that can receive by hospital staff who have previously registered. This notification will send through the conversation the worker has with the Telegram Bot. Testing for Telegram Bot and the integrated system was carried out in two types, namely for patient type users and for hospital worker type users. The test describes in Figure 4.



Figure 4. Start a Conversation

Figure 4 shows the response that the user will receive after sending the /start command. This command will start a conversation from the user to the bot. Patients can use other commands such as /daftar and /infodokter, as shown in Figure 6 and Figure 7.



Figure 5. Register for Examination

Figure 5 shows another message containing the command that will use to register for a health examination. The command is /umum for patients who register without insurance and /jaminan for insured patients.



Figure 6. Doctor's Schedule

Figure 6 shows a list of doctors' schedules available at the hospital. Patients can get this information by typing the command `/infodokter` into the Telegram Bot.

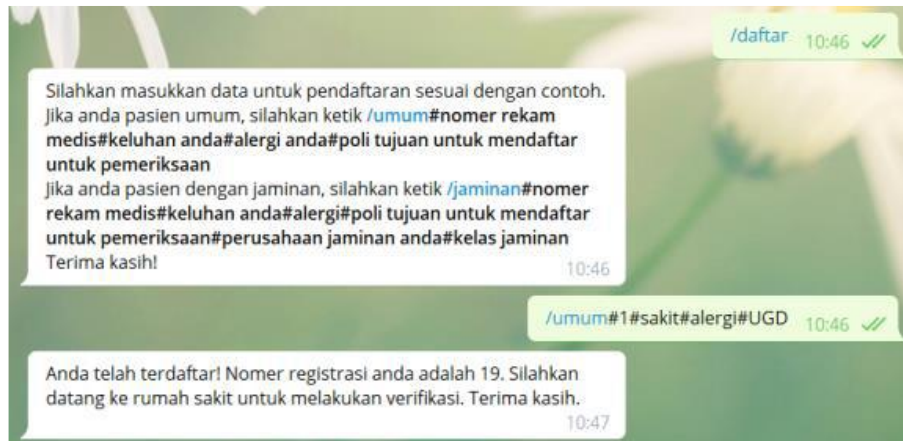


Figure 7. Register for Examination without Insurance.

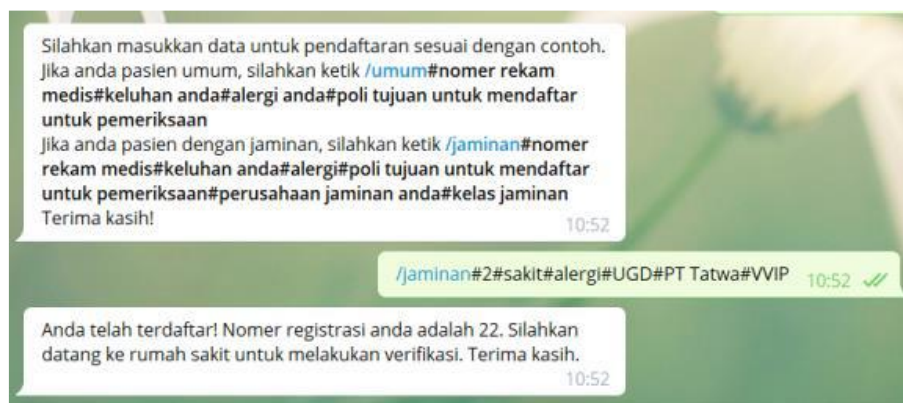


Figure 8. Registration with Insurance

Figure 7 shows an example of a user registering for an examination with `/umum` for a user without insurance or a general patient. While Figure 8 shows the commands used by a user with insurance. This type of registration has a separate command because patients need to enter more information, such as their company name and insurance class if they want to register with the insurance form.

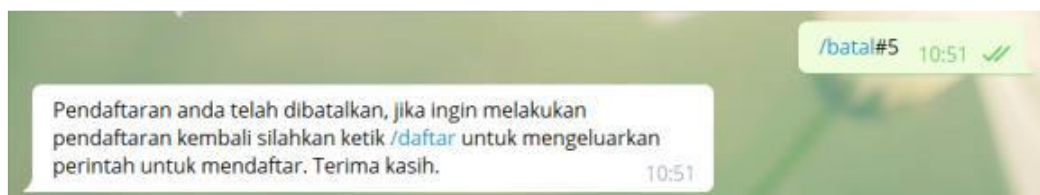


Figure 9. Unregistering for Examination

Figure 9 shows the feature for unregistering an examination. Patients can cancel a recently registered examination request by simply using the `/batal` command.



Figure 10. Register to Telegram Bot

Figure 10 shows the /register command for a hospital worker user. It is used for users to get their Telegram Chat ID, which will be used for notification purposes of this Telegram Bot. A part of the information system profile page required to enter the Telegram Bot Chat ID. Users will enter the Chat ID sent by Telegram Bot to the Telegram ID column on this profile page. Hence Telegram Bot can send notifications to users. Notices for hospital workers are vary based on their job differences. Other hospital workers cannot get this notification because the notification is set to be sent to users with doctor occupation type. System evaluation is the last part of the system development stage. This section will show possible errors or bugs in the system. Tests are carried out when the user uses the wrong command and cannot be recognized by the bot. The bot will display a notification that the bot does not understand what the user is trying to say and a list of commands that the bot understands.

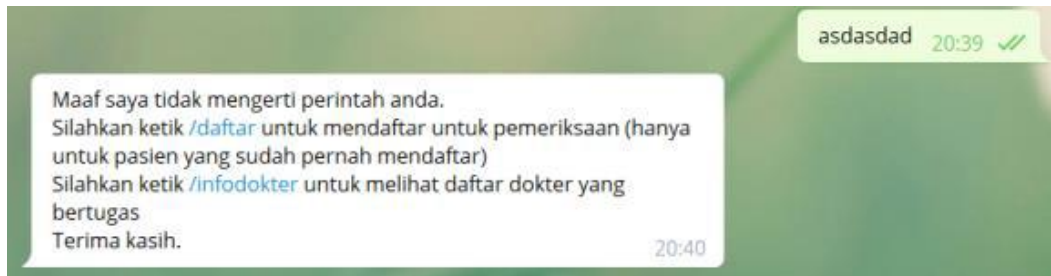


Figure 13. Incorrect Command

The test is carried out when a power outage and lost internet connection. The bot will not function because the supporting facilities, the internet are not connected to the application. Thus, the bot cannot poll, and the server cannot process data. System testing is done by asking testers to try all the features on the system. The test is carried out using a single sign on system to enter the system first and using the tester's Telegram account to get features on the Telegram Bot developed. Before testing begins, it is ensured that all system features and Telegram Bots do not have bugs and errors in their use. Examiners who have tried the Telegram system and bot are given a questionnaire which will be processed using the UAT (User Acceptance Testing) method and Likert's Summated Rating scale. The test continues by using the UAT method and a Likert scale of 1 to 5. The questionnaire divides into four aspects; suitability, performance, safety, and usability. The enormous response from respondents regarding the suitability aspect is strongly agreed (SA) with a percentage value of 48.6% and agree (A) with a percentage of 43% responses. The enormous response from respondents regarding the performance aspect is agreed (A) with a percentage of 66.7% responses, followed by moderately agreed (MA) of 18.4%, and strongly agreed (SA) of 13.4%.

The enormous respondent's response regarding the safety aspect is agreed (A) with a percentage of 50% and strongly agreed (SA) of 37.5%. The enormous response from respondents regarding the usability aspect is agreed (A) with a percentage of 50%, followed by strongly agreed (SA) of 44.5%. The data obtained through the UAT is measured using the LSR (Likert's Summated Rating) scale method. LSR aims to compare a person's attitude score with the scale distribution of other people groups. The answers to the statements are expressed in choices that accommodate answers between disagree to strongly agreed. A calculation to measure the success rate of the application using the LSR scale technique on the results of the questionnaire can see as follows. It can conclude from the enormous percentage of each aspect, such as the suitability aspect, which generates an agreed response (SA) with a 48.6% percentage. On the other hand, the performance aspect received an agreed response (A) of 66.7%. The safety aspect received an agreed response (A) of 50%. Then, the usability aspect received an agreed response (A) of 50%. The results to calculate the success of its implementation based on the assessment of the questionnaire results that the respondents have filled out are described using the LRS scale (Likert's Summated Rating) with a range of 768-960. A score of 768-960 means very positive, a score of 576-768 means positive (the system is considered quite successful), a score of 384-576 means negatives (the system is considered less successful), a score of 192-384 means very negative (the system is considered failed).

Discussion

The system that has been built is compared with other systems referenced in this research to better understand which points this system brings those other systems have not. The comparison will be presented with a table. It will compare each feature possessed by each system.

- App1: Hospital Telegram Bot
- App2: Queue Bot
- App3: Rail Bot (Sakila et al., 2018)
- App4: Bot UIII (Setiaji & Paputungan, 2018)

Table 3. Comparison Table

Feature	System Code / Name			
	App 1	App 2	App 3	App 4
Dashboard for Chat Log	✓	X	X	✓
Can handle messages via commands	✓	✓	✓	✓
Can handle error messages from users	✓	✓	✓	✓
Notification Features for Users	✓	✓	✓	✓

Table 3 shows a comparison between the Telegram Bot system built with several systems used as a reference for making the Telegram Bot module in this hospital management information system. Each application that supports the feature will be written as a supporter, while the applications that do not support the feature describes as follows: The dashboard feature for message logs is found in the web-based hospital management information system, the Telegram Bot module designed in this research, and the UIII Bot (Risanty & Sopiyan, 2017; Setiaji & Paputungan, 2018). This feature is used as a feature that can help system administrators to view message logs between Telegram users and bots (Arimbawa P et al., 2018; Zubaidi & Ramdani, 2019). The messaging feature using commands is owned by every Telegram Bot referenced in this research. This feature allows users to use the Telegram Bot more efficiently by using the commands available in the Telegram Bot (Halim et al., 2021; Rahman & Yaqin, 2019). The error message handling feature can be overcome by every Telegram Bot system referenced in this research (Aisyah et al., 2021; Asmazori, 2021; Soraya et al., 2021). The Telegram Bot system developed in this research minimizes message errors by issuing a message that there is an error in the command or typing of the message sent by the user. The notification message feature for users is available on every referenced Telegram Bot system design. This feature can be in the form of automatic messages sent by Telegram Bots when certain events occur (Arimbawa P et al., 2018; Risanty & Sopiyan, 2017).

4. Conclusions and Suggestions

This research concludes that the Telegram Bot can develop for Integrated Hospital Information System. The business processes that have been built adapted to the flow in the hospital, such as the patient registration process through the website and Telegram bots and the notification process for each module in the hospital management information system. This bot facilitates patients and hospital workers. The patients can quickly register at the hospital easily because of BOTS usage for registration assistance, and hospital workers get information immediately.

5. References

Aisyah, R. N., Istiqomah, D. M., & Muchlisin, M. (2021). Rising English Students' Motivation in Online Learning Platform: Telegram Apps Support. *Journal of Ultimate Research and Trends in Education*, 3(2). <https://doi.org/10.31849/utamax.v3i2.6464>.

Arimbawa P, D. A. K., Gede Darma Putra, I. K., & Sukarsa, I. M. (2018). Library System Using Radio Frequency Identification (RFID) and Telegram Bot API. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, 9(1), 40. <https://doi.org/10.24843/lkjiti.2018.v09.i01.p05>.

Asmazori, M. (2021). Rancang Bangun Alat Pendeteksi NOx dan CO Berbasis Mikrokontroler ESP32 dengan Notifikasi Via Telegram dan Suara. *JITCE (Journal of Information Technology and Computer Engineering)*, 5(2). <https://doi.org/10.25077/jitce.5.02.57-62.2021>.

Balaraman, P., & Kosalram, K. (2013). E -Hospital Management & Hospital Information Systems – Changing Trends. *International Journal of Information Engineering and Electronic Business*, 5(1), 50–58. <https://doi.org/10.5815/ijieeb.2013.01.06>.

Çetin, S. B., Cebeci, F., Eray, O., Coşkun, M., & Gözkaya, M. (2021). Emergency Nurse Triage in the Hospital Information Management System: A quality Improvement Study. *International Emergency Nursing*, 59. <https://doi.org/10.1016/j.ienj.2021.101069>.

- Dyatmika, D., Sudana, O., & Putri, G. A. A. (2019). Untethering the Queue Based on Multi Channel Access (MCA) Technology at Hospital Radiology Section. *Journal of Information Systems Engineering and Business Intelligence*, 5(2), 146. <https://doi.org/10.20473/jisebi.5.2.146-155>.
- Fernández-Álvarez, P., & Rodríguez, R. J. (2022). Extraction and Analysis of Retrievable Memory Artifacts from Windows Telegram Desktop Application. *Forensic Science International: Digital Investigation*, 40. <https://doi.org/10.1016/j.fsidi.2022.301342>.
- Gregorio, J., Gardel, A., & Alarcos, B. (2017). Forensic Analysis of Telegram Messenger for Windows Phone. *Digital Investigation*, 22. <https://doi.org/10.1016/j.diin.2017.07.004>.
- Halim, I. H. A., Ibrahim, A., Mahamad, M. F., & Fuzi, M. (2021). Automated Alert System for River Water Level and Water Quality Assessment using Telegram Bot API. *Journal of Computing Research and Innovation*, 6(3). <https://doi.org/10.24191/jcrinn.v6i3.234>.
- Hanlin, E. R., Chan, H. K., Hansen, M., Wendelberger, B., Shah, M. I., Bosson, N., Gausche-Hill, M., & VanBuren, J. M. (2022). Epidemiology of Out-of-Hospital Pediatric Airway Management in The 2019 National Emergency Medical Services Information System Data Set. *Resuscitation*, 173. <https://doi.org/10.1016/j.resuscitation.2022.01.008>.
- He, Z., Zhi, M., Ji, J., Lian, J., Guo, R., Sun, J., & Liu, Y. (2021). Patient Satisfaction with Non-Clinical Nursing Care Provided by The Nursing Assistant Under Different Management Models in Chinese Public Tertiary Hospital. *Applied Nursing Research*. <https://doi.org/10.1016/j.apnr.2021.151431>.
- Kania-Richmond, A., Reece, B. F., Suter, E., & Verhoef, M. J. (2015). The Professional Role of Massage Therapists in Patient Care in Canadian Urban Hospitals - A Mixed Methods Study. *BMC Complementary and Alternative Medicine*, 15(1), 1–10. <https://doi.org/10.1186/s12906-015-0536-4>.
- Kosherbayeva, L., Kalmakhanov, S., Hailey, D., Pazilov, S., Seiduanova, L., Kozhamkul, R., & Jaworzynska, M. (2020). Rethinking Priorities in Hospital Management: A Case from Central Asia. *Health Policy and Technology*, 9(3).
- Krishna, S. M. H., & Sharma, R. (2019). Survey on Application Programming Interfaces in Software Defined Networks and Network Function Virtualization. *Global Transitions Proceedings*, 2(2). <https://doi.org/10.1016/j.gltp.2021.08.018>.
- Liu, X., Meneghin, M., & Shapiro, V. (2022). An Application Programming Interface for Multiscale Shape-Material Modeling. *Advances in Engineering Software*, 161. <https://doi.org/10.1016/j.advengsoft.2021.103046>.
- Nayagam, K. G., & Arun, K. (2014). Building Applications with Social Networking API's. *International Journal Advanced Networking and Applications*, 5(5), 2070–2075.
- Nobari, A. D., Sarraf, M. H. K. M., Neshati, M., & Daneshvar, F. E. (2021). Characteristics of Viral Messages on Telegram; The world's Largest Hybrid Public and Private Messenger. *Expert Systems with Applications*, 168. <https://doi.org/10.1016/j.eswa.2020.114303>.
- Ogorodnyk, O., Larsen, M., Martinsen, K., & Lyngstad, O. V. (2021). Development of Application Programming Interface Prototype for Injection Molding Machines. *Procedia CIRP*, 97. <https://doi.org/10.1016/j.procir.2020.07.005>.
- Pantartzis, E., Edum-Fotwe, F. T., & Price, A. D. F. (2017). Sustainable Healthcare Facilities: Reconciling Bed Capacity and Local Needs. *International Journal of Sustainable Built Environment*, 6(1), 54–68. <https://doi.org/10.1016/j.ijbsbe.2017.01.003>.
- Rahman, M. F., & Yaqin, M. A. (2019). Absensi Qr Code Berbasis E-Confirmation (Bot Telegram) Dan E-Notification dengan Teknologi Google Access. *Teknika: Engineering and Sains Journal*, 3(1). <https://doi.org/10.5281/zenodo.3549675>.
- Ran, X., Zhou, F., Zhong, M., Liu, Y., & Zhang, J. (2020). Innovative Applications of Patient Experience Big Data in Modern Hospital Management Improve Healthcare Quality. *Chinese Medical Sciences Journal*, 35(4). <https://doi.org/10.24920/003855>.
- Rianto, A., Rahmatulloh, A., & Firmansyah, T. (2019). Telegram Bot for Automation of Academic. *Journal Publications & Informatics Engineering Research*, 3(2), 73–78. <https://doi.org/10.33395/sinkron.v3i2.10023>.
- Risanty, R. D., & Sopiyan, A. (2017). Pembuatan Aplikasi Kuesioner Evaluasi Belajar Mengajar Menggunakan Bot Telegram Pada Fakultas Teknik Universitas Muhammadiyah Jakarta (Ft-Umj) Dengan Metode Polling. *Jurnal Prosiding Seminar Nasional Sains Dan Teknologi*, 1(2), 1–9.
- Rochmah, T. N., Fakhruzzaman, M. N., & Yustiawan, T. (2020). Hospital Staff Acceptance toward Management Information Systems in Indonesia. *Health Policy and Technology*, 9(3). <https://doi.org/10.1016/j.hlpt.2020.07.004>.
- Sakila V., S, Shrivastava, A., Ansari, M., Kar, L. K., & Kumar, M. (2018). RAILBOT: A Railway (IRCTC) Chatbot. *International Journal of Engineering Science and Computing*, 8(4), 16723–16726.

- Sakila V, Sahaya, Shrivastava, A., Ansari, A. M., Kar, L. K., & Kumar, M. (2018). RAILBOT : A Railway (IRCTC) Chatbot. *International Journal of Engineering Science and Computing*, 8(4), 16723–16726.
- Setiaji, H, & Papatungan, I. V. (2018). Design of Telegram Bots for Campus Information Sharing. *IOP Conference Series: Materials Science and Engineering*, 325. <https://doi.org/10.1088/1757-899X/325/1/012005>.
- Setiaji, Hari, & Wahid, F. (2015). Masalah Implementasi Sistem Informasi Rumah Sakit : Pelajaran dari Beberapa Proyek. *Seminar Nasional Informatika Medis VI. Yogyakarta*, 97–105.
- Soraya, A., Riyanto, E., & Solikhin. (2021). Aplikasi Presensi dengan Barcode Scanner dan Raspberry Pi Terintegrasi Bot Telegram. *Jurnal Informatika UPGRIS*, 6(2). <https://doi.org/10.26877/jiu.v6i2.6567>.
- Sutikno, T., Handayani, L., Stiawan, D., Riyadi, M. ., & Subroto, I. M. I. (2016). WhatsApp, Viber and Telegram: which is the Best for Instant Messaging? *International Journal of Electrical and Computer Engineering*, 6(3), 909–914. <https://doi.org/10.11591/ijece.v6i3.pp909-914>.
- Wang, C., Chen, Y., Yang, F., Yu, X., & Wang, J. (2016). Efficacy of Computer-Based Endoscope Cleaning and Disinfection Using A Hospital Management Information System. *American Journal of Infection Control*, 44(8). <https://doi.org/10.1016/j.ajic.2016.03.065>.
- Zubaidi, A., & Ramdani. (2019). Information Sharing and Academic Services Based on Telegram BOT in Informatics. *Information Sharing and Academic Services Based on Telegram BOT in Informatics*, 1(1), 103–110. <https://doi.org/10.29303/jtika.v1i1.27>.