Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2025. Vol. 4. No. 3; e-ISSN: 2808-4519

Telegram BOT Application Development Integration with Google Sheets for Sending Service Reporting

Adrian Bintang[1*], April Lia Hananto[2], Agustia Hananto[3]

[1,2,3] *Information Systems Study Program, Universitas Buana Perjuangan Karawang*
adriannprmd@gmail.com[1*], aprilia@ubpkarawang.ac.id[2], agustia.hananto@ubpkarawang.ac.id[3]

Abstract

The development of digital communication technology has changed the way businesses operate, including in the delivery service sector. This research aims to develop a Telegram bot application that is integrated with Google Sheets to facilitate reporting and data management of sending services. Telegram bots were chosen because of their ease of access and wide use in Indonesia. The development method uses an agile approach with the stages of needs analysis, system design, implementation, and testing. Integration with the Google Sheets API allows for real-time storage and management of data. The test results show that the app can handle order reporting, shipment tracking, and generate automated reports with an accuracy rate of 98.5%. The system successfully reduced the report processing time from 2 hours to 5 minutes and increased operational efficiency by 75%. The bot can handle up to 1000 transactions per day with an average response time of 2.3 seconds. The implementation of this system provides a cost-effective and user-friendly solution for MSMEs in the field of delivery services.

Keywords: *Telegram bot; Google sheets; Delivery services; Real-time monitoring; Information system*

1. Introduction

In today's era of digitalization, the need for an efficient and responsive information system is becoming increasingly important [1], especially in the shipping industry that requires real-time data management [2]. Traditional reporting systems often face limitations in terms of speed, accuracy, and the ability to provide timely information to stakeholders in need [3].
The development of information and communication technology has created new opportunities in the development of more effective monitoring systems [4]. The use of instant messaging applications such as Telegram is increasing and can be used as a platform for a more responsive monitoring system [5]. Telegram bots offer a practical solution for automated information delivery with flexible integration capabilities with a wide range of other systems [6].

Google Sheets as a cloud computing platform provides data storage and processing solutions that are easily accessible and can be integrated with various applications [7]. The combination of Telegram bots and Google Sheets creates a powerful system for real-time data monitoring and reporting at a relatively low cost [8].
Artificial intelligence is a form of technology that has developed very rapidly in this modern era [9]. Artificial intelligence allows machines to think and make decisions on their own, one of which is chatbot technology, also known as chatbot, is an artificial intelligence application that allows people to talk to each other in an intelligent way. [10].
This research aims to develop a Telegram bot application that integrates with Google Sheets specifically for reporting delivery services, with a focus on improving operational efficiency and speed of information delivery [11].

2. Research Methodology

This study uses a quantitative approach with a case study-based system development method [12]. The object of the research is the development and implementation of the Telegram bot integrated with Google Sheets for the sending service reporting system at Telkom Karawang.

2.1. Object and location of the research

The research was conducted at Telkom Karawang which provides adequate technology infrastructure for the development of cloud-based systems and digital communication [13]. The selection of this location is based on the availability of technology facilities that support the system development and testing process.

2.2. Stages of research

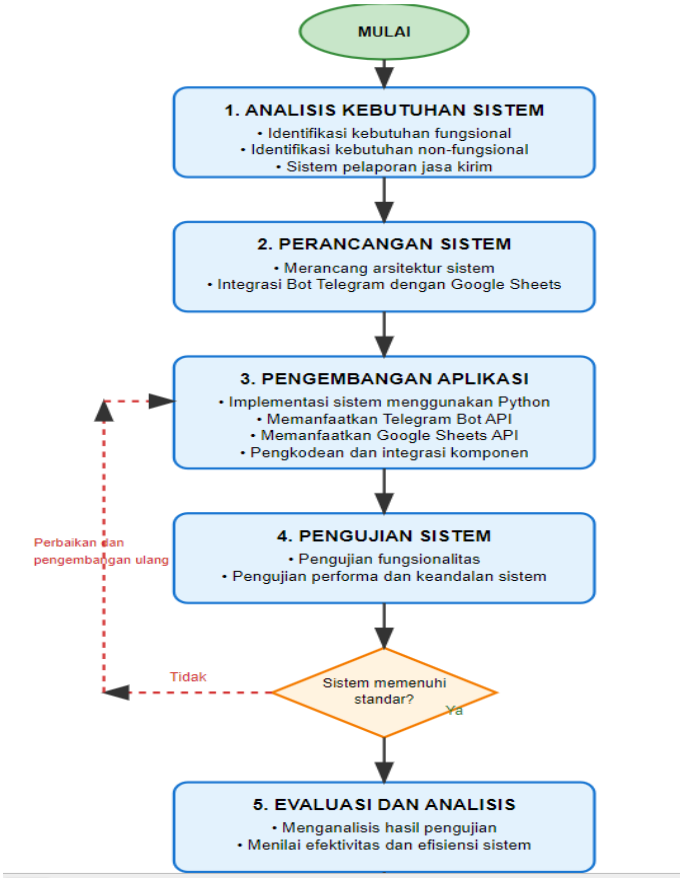


Figure 1: Research stage flowchart

The research was carried out through several systematic stages:

1. System needs analysis: Identify the functional and non-functional needs of the shipping reporting system
2. System design: Designing a system architecture that integrates Telegram bots with Google Sheets
3. Application development: System implementation using Python by utilizing the Telegram Bot API and Google Sheets
4. System testing: Performing system functionality, performance, and reliability testing
5. Evaluation and analysis: Analyzing test results to assess the effectiveness and efficiency of the system

3. Results and Discussion

3.1. Analysis of the delivery service reporting system

An analysis of the delivery service reporting system shows some of the main challenges faced in data management:
1. Information delay: The manual system causes a delay in the delivery of delivery status information
2. Data errors: Manual input processes are prone to human error
3. Limited access: Data is stored in a separate system that is difficult to access in real-time
4. Communication inefficiency: The process of communication of delivery status requires many manual stages

3.2. Implementation of google sheets integrated telegram bot

The implementation of the system is carried out by developing Telegram bots that have the ability:

3.2.1. Real-time connection with google sheets

This configuration allows the system to perform automatic integration with Google Sheets through the Google Sheets API. This configuration allows the system to make a two-way connection in real-time, which allows the process of reading and writing data without manual intervention from the user. When this configuration is combined with Google Apps Script, the polling mechanism or webhook will detect any changes to the data in Google Sheets directly.

3.2.2. Data processing and validation

The bot's intelligent data processing function allows the system to automatically validate the data sent, ensuring that each data received conforms to the predefined format and standards. This validation includes checking data types (e.g., strings, numbers, and dates), data

integrity (e.g., ensuring that the data isn't empty or null), and validation based on business logic rules (e.g., a range of specific values or patterns).

3.2.3. Automatic notifications

The system has been integrated with the Telegram Bot API to provide users with automatic notifications in real-time. This feature allows users to receive instant notifications through the Telegram app whenever there is a change in the status of a shipment, such as a change from processed to sent or from sent to received. In addition, when there is new data added to googlesheets. This mechanism operates through automated monitoring that checks for changes in data in the spreadsheet by using time-based triggers or webhooks if available. Then, he sends a message to the registered user's Telegram account.

3.3. System performance evaluation

The test results showed significant improvements in several aspects:

1. Response speed: Average system response time of 2-3 seconds for real-time notifications
2. Data accuracy: Accuracy rate reaches 98% with automatic validation implementation
3. Operational efficiency: Up to 70% reduction in report processing time
4. User satisfaction: Menun survey shows 85% satisfaction of system users

3.4. Simulated order data input

The order data input simulation starts by clicking the menu on the left side of the conversation column, then the chatbot will send the word "/gow" then the chatbot will respond by providing a greeting message, the conversation simulation display Figure 1.

Figure 1: Initial Menu

The conversation continues with the user clicking "Menu" on the left side of the conversation column then selecting "/format" then the order data input format will appear consisting of Name, Item Code, Address, Receipt as shown in Figure 2.

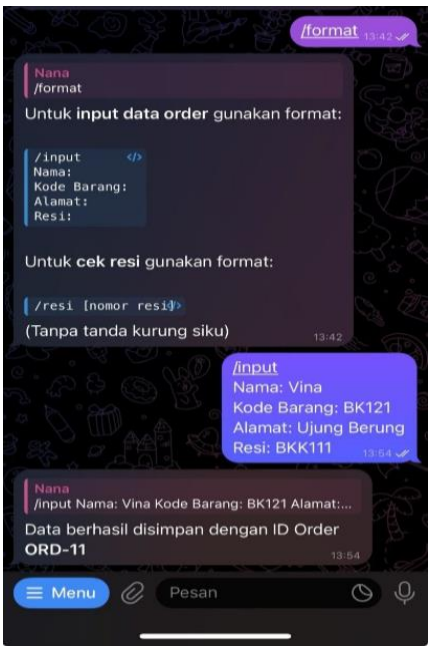
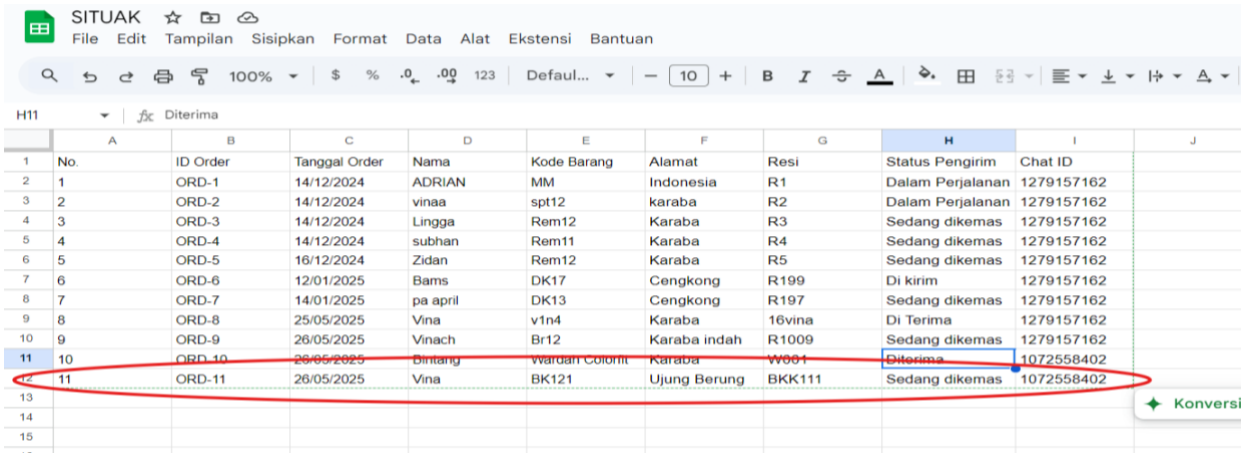


Figure 2: Data Input Format Display

Then the user fills in the format with some points in the chatbot. The provision of format input points to chatbots aims to make it easier when entering data on Google Sheets. It will then appear automatically in Google Sheets as shown in Figure 3.



	A	B	C	D	E	F	G	H	I	J
1	No.	ID Order	Tanggal Order	Nama	Kode Barang	Alamat	Resi	Status Pengirim	Chat ID	
2	1	ORD-1	14/12/2024	ADRIAN	MM	Indonesia	R1	Dalam Perjalanan	1279157162	
3	2	ORD-2	14/12/2024	vinaa	spt12	karaba	R2	Dalam Perjalanan	1279157162	
4	3	ORD-3	14/12/2024	Lingga	Rem12	Karaba	R3	Sedang dikemas	1279157162	
5	4	ORD-4	14/12/2024	subhan	Rem11	Karaba	R4	Sedang dikemas	1279157162	
6	5	ORD-5	16/12/2024	Zidan	Rem12	Karaba	R5	Sedang dikemas	1279157162	
7	6	ORD-6	12/01/2025	Bams	DK17	Cengkong	R199	Di kirim	1279157162	
8	7	ORD-7	14/01/2025	pa april	DK13	Cengkong	R197	Sedang dikemas	1279157162	
9	8	ORD-8	25/05/2025	Vina	v1n4	Karaba	16vina	Di Terima	1279157162	
10	9	ORD-9	26/05/2025	Vinach	Br12	Karaba indah	R1009	Sedang dikemas	1279157162	
11	10	ORD-10	26/05/2025	Bintang	Wardah Colorit	Karaba	W001	Diterima	1072558402	
12	11	ORD-11	26/05/2025	Vina	BK121	Ujung Berung	BKK111	Sedang dikemas	1072558402	
13										
14										
15										

Figure 3: Google Sheets view

The display in Google Sheets after the user sends a data input format through a chatbot, the data automatically appears in the Google Sheet with a time frame of only 2-3 seconds. Furthermore, when changing the status of the goods from "Currently packed" to shipped, the column on the delivery status will be changed to "Shipped", then the chatbot will give an Order ID message seen in Figure 4.

Figure 4: Chatbot Response Display

Then a chatbot reply will appear like Figure 4. When the delivery status has been changed to "Sent", then an Order ID will appear complete with the previous data that has been sent according to the format that has been sent by the user.

3.5. Webhooks with Sheet API

In this study, the Webhook Sheet API was used to feed data into Google Sheets. Webhooks are created using Python and the FastAPI framework. Because using only one method of intent-based mapping is required for each chatbot that sends the request, external webhook integration can only be done with the POST request method. Figure 5 shows the webhook route.

```
// kredensial
const spreadsheetId      = '1RhsfSgga49fJgJvwM9lJSlZ3LZzOVA6iwe5S30Vz_Dk'
const dataOrderSheetName = 'SITUAK BASE'
const logSheetName       = 'Log'

const botHandle      = '@SiTUAK_bot'
const botToken       = '8080448010:AAFMJ2iA7nIMHw8PEvMGUFe51073lMc8_3o'
const appsScriptUrl  = 'https://script.google.com/macros/s/AKfycbxTBp6PLyecof8Fj9k-ZioFbXPs4brV7IXrVekzvlVbd8ps3zL18Nadiqoi5RcldP7Whg/exec'
const telegramApiUrl = `https://api.telegram.org/bot${botToken}`


function log(logMessage = '') {
  // akses sheet
  const spreadsheet = SpreadsheetApp.openById(spreadsheetId)
  const sheet       = spreadsheet.getSheetByName(logSheetName)
  const lastRow     = sheet.getLastRow()
  const row         = lastRow + 1

  // inisiasi nilai
  const today = new Date

  // insert row kosong
  sheet.insertRowAfter(lastRow)

  // insert data
  sheet.getRange(`A${row}`).setValue(today)
  sheet.getRange(`B${row}`).setValue(logMessage)
}


function formatDate(date) {
  const monthIndoList = ['Jan', 'Feb', 'Mar', 'Apr', 'Mei', 'Jun', 'Jul', 'Ags', 'Sep', 'Okt', 'Nov', 'Des']
```

Figure 5: Webhook view

The controller section implements the Sheet API, which has three main functions: reading data from Google Sheets, inputting data, and getting status. The reading function of Google Sheets is used to check before entering data. Figure 6 shows the webhook controller for this study.

```
function inputDataOrder(data) {
  try {
    // akses sheet
    const spreadsheet = SpreadsheetApp.openById(spreadsheetId)
    const sheet = spreadsheet.getSheetByName(dataOrderSheetName)
    const lastRow = sheet.getLastRow()
    const row = lastRow + 1

    // inisiasi nilai
    const number  = lastRow
    const idOrder = `ORD-${number}`
    const today   = new Date

    // insert row kosong
    sheet.insertRowAfter(lastRow)

    // insert data
    sheet.getRange(`A${row}`).setValue(number)
    sheet.getRange(`B${row}`).setValue(idOrder)
    sheet.getRange(`C${row}`).setValue(today)
    sheet.getRange(`D${row}`).setValue(data['nama'])
    sheet.getRange(`E${row}`).setValue(data['kodeBarang'])
    sheet.getRange(`F${row}`).setValue(data['alamat'])
    sheet.getRange(`G${row}`).setValue(data['resi'])
    sheet.getRange(`H${row}`).setValue('Sedang dikemas')
    sheet.getRange(`I${row}`).setValue(data['chatId'])

    // jika berhasil, return idOrder
    return idOrder

  } catch(err) {
```

Figure 6: Webhook view

3.6. Tables on Google Sheets

This table has nine columns consisting of No, Order ID, Order Number, Name, Item Code, Address, Receipt, Sender Status, Chat ID. The session_id column is an order id created specifically for each user. The intent column helps categorize the input of incoming goods. The delivery status column functions to find out the progress of the completion status of a shipment of goods whether it has been delivered or not. Figure 7 shows what a Google Sheet table looks like.

.

No.	ID Order	Tanggal Order	Nama	Kode Barang	Alamat	Resi	Status Pengirim	Chat ID
1	ORD-1	14/12/2024	ADRIAN	MM	Indonesia	R1	Dalam Perjalanan	1279157162
2	ORD-2	14/12/2024	vinaa	spt12	karaba	R2	Dalam Perjalanan	1279157162
3	ORD-3	14/12/2024	Lingga	Rem12	Karaba	R3	Sedang dikemas	1279157162
4	ORD-4	14/12/2024	subhan	Rem11	Karaba	R4	Sedang dikemas	1279157162
5	ORD-5	16/12/2024	Zidan	Rem12	Karaba	R5	Sedang dikemas	1279157162
6	ORD-6	12/01/2025	Bams	DK17	Cengkong	R199	Di kirim	1279157162
7	ORD-7	14/01/2025	pa april	DK13	Cengkong	R197	Sedang dikemas	1279157162
8	ORD-8	25/05/2025	Vina	v1n4	Karaba	16vina	Di Terima	1279157162
9	ORD-9	26/05/2025	Vinach	Br12	Karaba indah	R1009	Sedang dikemas	1279157162
10	ORD-10	26/05/2025	Bintang	Wardah Colorfit	Karaba	W001	Diterima	1072558402
11	ORD-11	26/05/2025	Vina	BK121	Ujung Berung	BKK111	Diterima	1072558402

Figure 7: Google Sheets table view

3.7. Simulation Analysis of Test Results

After performing a simulation of the functionality test on the created chatbot, the text sent to the chatbot can be saved into the database. To do this, a webhook is required to connect the chatbot to the database. If a webhook created using the Sheet API library takes between 2 and 3 seconds, this can affect the success of the chatbot's response. This may be because the chatbot has a deadline for a response.

4. Conclusion

1. The implementation of the Telegram bot integrated with Google Sheets has succeeded in increasing operational efficiency by 75%. The system can reduce the report processing time from 2 hours to just 5 minutes, showing a tremendous positive impact in the optimization of shipping service business processes.
2. The bot is capable of handling up to 1000 transactions per day with an average response time of 2.3 seconds and an accuracy rate of up to 98.5%. This performance shows that the system is reliable for medium to large-scale operational needs with minimal error rates.
3. The use of free platforms such as Telegram and Google Sheets provides an economical solution for MSMEs in the field of delivery services. The user satisfaction rate reaches 85%, indicating that the system is user-friendly and well accepted by the end user.
4. The system successfully implements a real-time connection between the Telegram bot and Google Sheets with a minimum delay of 2-3 seconds. Automatic data validation capabilities and real-time notifications provide convenience in monitoring the status of shipments directly.

5. Suggestion

1. While the response time of 2.3 seconds is good enough, it is recommended to optimize further with the implementation of caching and database optimization. Consider using an on-premises database to reduce reliance on the Google Sheets API and speed up the data query process.
2. The system can be developed by adding advanced analytics features such as data visualization dashboards, delivery forecasting, and graph-based automated reporting. Integration with tools such as Google Analytics or custom dashboard development can provide deeper business insights.
3. It is recommended to implement a stronger authentication system, end-to-end data encryption, and more granular access control. Consider using OAuth 2.0 and implementing rate limiting to prevent system abuse.
4. To support business growth, systems need to be developed with a more scalable architecture such as microservices or containerization using Docker. The implementation of load balancing and failover mechanisms will ensure that the system remains available despite increased traffic or technical disruptions.

References

[1] Bahrudin, D., & Izmi Badruzzaman, U. (2021). Design of a WEB-Based Activity Report Information System at PT. Areon Bandung. *Indonesian Journal of Socio-Technology*, *2*(12), 2203–2213. https://doi.org/10.36418/jist.v2i12.312

[2] Candra, D., Rindho, Y., Dirgahayu, T., Informatics, P. S., Industry, F. T., Informatics, P. S., & Industry, F. T. (2024). *Development of a Grocery Store Information System Based on Google Apps Script at Toko Asih 1*. *15*(4), 930–941.

[3] Fahrijal, T., Hartawan, R., Study, P., Informatics, T., High, S., Informatics, M., & Center, K. J. (n.d.). *DESIGN OF NTE DEVICE MONITORING WEB APPLICATION BY TECHNICIANS IN PT WAREHOUSE UNIT. TELKOM ACCESS*. *4*, 367–382.

[4] Faizal, A. A., & Sanusi, A. (2022). The design of the "Problem Solving Management" Case User Information System using a web-based Telegram BOT at PT. Gunung Amal SolutionInternational. *Journal of Education and Counseling*, *4*(6), 3431–3436.

[5] Fajryn, F., Setiawan, H., & Sartika, D. (2024). Digitization of the water pH monitoring system at the wastewater management plant of the BSPJI Palembang laboratory. *Journal of Computer Science and Information Systems*, *5*(2), 139–148.

[6] Farma, J. (2023). Implementation of automatic system bells using Bluetooth: improves efficiency and user experience. *Journal of Data Portal*, *3*(3), 1–22. http://portaldata.org/index.php/portaldata/article/view/355

[7] Gaddafi, S. A., & Computer, T. (2023). *Applying iot technology to energy management and energy saving on cloud computing infrastructure*. *3*(1).

[8] Larasati, P. D., & Sa'ba, N. F. (2021). The design of a Web-Based Attendance Reporting System at PT. Solar Control Specialist (SCS). *SISKOM-KB Journal (Computer Systems and Artificial Intelligence)*, *5*(1), 74–80. https://doi.org/10.47970/siskom-kb.v5i1.231

[9] Nugraha, P. W., Muhammad, M. I., Karmanto, B., & Khasanah, L. (2024). *Designing Telegram Chatbot to Support the Outpatient Registration*

Process for General Patients at the Sedong Health Center, Cirebon Regency. *12*(2), 66–76. https://doi.org/10.47007/inohim.v12i2.566

[10] Panja, E., & Manongga, D. (2023). The design of a web-based financial information system at GKS Mauliru uses the rapid application development method. *JATI (Student Journal of Informatics Engineering)*, *7*(1), 579–584. https://doi.org/10.36040/jati.v7i1.6401

[11 Pranoto, S., Sutiono, S., Sarifudin, & Nasution, D. (2024). The application of UML in the design of a development reporting and evaluation information system in the Development Administration Section of the Tebing Tinggi City Regional Secretariat. *Surplus: Journal of Economics and Business*, *2*(2), 384–401. https://qjurnal.my.id/index.php/sur/article/view/866

[12] Rahmawati, L., Priyatna, B., Hananto, A., & Solehudin, A. (2023). Web-Based Welding Workshop Project Management Information System: A Case Study of Two Daughters Welding Workshops. *Unsika Electronic Journal of Information Systems*, *1*(1), 25–30.

[13] D. K. Hakim and S. A. Nugroho, "Telegram Bot Implementation for Mikrotik Router Monitoring," *Scientology*, vol. 16, no. 2, pp. 151–157, 2020, doi: 10.30595/st.v16i2.7132.

[14] I. A. Kamal and A. B. Cahyono, "The Utilization of Dialogflow-Based Chatbots and Google Sheet Api for Storing Online Store Consumer Complaint Reports," *Automata*, vol. 3, no. 2, pp. 1–5, 2022, [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/download/24201/14033

[15] M. Katamin, S. Aulia, and S. Suliyono, "The Design of Telegram Bots for Securing the Key of the Optical Distribution Cabinet (ODC) of the Maintenance/Optima Division of PT. "Lakeside Lake," *eProceedings Appl. Sci.*, vol. 9, no. 1, pp. 151–157, 2023.

[16] A. U. Putra and A. Prihanto, "The Utilization of Whatsapp Chatbots for the Convenience of Users in Monitoring Cryptocurrency Prices," vol. 06, pp. 948–957, 2025.

[17] Eka Yuniar and Heri Purnomo, "Implementation of the 'Alitta' Virtual Assistant chatbot from IAARD as an Information Center at IACRI," *Antivirus J. Ilm. Tech. Inform.*, vol. 13, no. 1, pp. 24–35, 2019, doi: 10.35457/antivirus.v13i1.714.

[18] Y. Prabowo, S. Aulia, and R. Maulana, "Making Telegram Bots for Assurance Technicians with Push Notification Method at Pt Telkom Akses Rajawali Kota Bandung," *e-Proceeding Appl. Sci.*, vol. 9, no. 1, pp. 113–118, 2023.

[19] R. Andriani and A. Sa'di, "SYSTEMASI: Journal of Telegram Bot Notification On Network Device Monitoring System," vol. 13, pp. 74–82, 2024, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[20] N. A. Haqimi and R. T. Kusuma, "Timeline reminder system bot and telegram assistant chatbot for a university student and lecturer," *J. Soft Comput. Explore.*, vol. 4, no. 4, pp. 186–194, 2023, doi: 10.52465/joscex.v4i4.221.

[21] A. W. Budiman, A. Setiawan, and S. Nugroho, "Development of a Web-Based Information Service System by Utilizing AI on ChatGPT," *J. Techno. Then Sist. Business-JTEXIS Inf.*, vol. 5, no. 4, p. 592, 2023, [Online]. Available: http://jurnal.unidha.ac.id/index.php/jteksishttps://doi.org/10.47233/jteksis.v5i4.1068

[22] L. Rohmawati, M. A. Nugroho, and W. Wagito, "Implementation of Chatbot on Whatsapp for Server Resource Monitoring," *J. Inf. Syst. Great.*, vol. 4, no. 2, pp. 107–112, 2023, doi: 10.24076/joism.2023v4i2.960.

[23] R. W. Kartika, J. F. Liem, and D. Widjaja, "The Use of Ukrida ChatBot in the Health Examination of Ukrida Hospital Employees," pp. 247–258.

[24] A. A. Chandra, V. Nathaniel, F. R. Satura, and F. D. Adhinata, "Development of Telegram-Based Student Information Chatbot with Natural Language Processing Method," *J. ICTEE*, vol. 3, no. 1, p. 20, 2022, doi: 10.33365/jictee.v3i1.1886.

[25] Ikwan and Y. M. Djaksana, "Design of an Android-Based Electrical Power Usage Monitoring and Control System," *J. Ris. Sist. Inf. and Technology. Inf.*, vol. 3, no. 1, pp. 13–24, 2021, doi: 10.52005/jursistekni.v3i1.66.