



Utilization of Telegram Bot for Integrated Data Visualization with Google Sheets at PT Telkom Cibitung

Renaldi Da Silva¹, Ahmad Sinnun^{2*}, Muhamad Tabrani³

^{1,2,3} *Fakultas Teknik & Informatika, Universitas Bina Sarana Informatika*
aldynote5@gmail.com¹, ahmad.axn@bsi.ac.id^{2*}, muhammad.mtb@bsi.ac.id³

Abstract

In today's digital age, organizations demand real-time, accurate, and efficient access to data in order to support decision-making and enhance operational workflows. This research focuses on the development and implementation of a Telegram Bot integrated with Google Sheets to improve data visualization and internal coordination at PT Telkom Cibitung. Prior to this innovation, the process of retrieving data from Google Sheets and sharing it via Telegram was conducted manually, leading to frequent errors, delays in communication, and inefficiencies in data-driven tasks—particularly within the sales division. The study adopts a descriptive-analytical method to examine existing workflows and to design a solution that automates data retrieval using a bot. The Telegram Bot was programmed to fetch and present key data points directly from Google Sheets based on user input, reducing the dependency on manual operations. The system was tested within the sales division and demonstrated significant improvements in speed, accuracy, and usability. Based on the analysis and findings, it can be concluded that the integration of Telegram Bot with Google Sheets provides a viable and scalable solution to the problems previously encountered at PT Telkom Cibitung. This approach not only eliminates the risk of data misinterpretation and human error but also enhances interdepartmental coordination. The implementation of such automation technology reflects a strategic move towards digital transformation and highlights the value of low-code solutions in business environments with limited IT resources.

Keywords: Telegram Bot, Google Sheets Integration, Data Visualization

1. Introduction

In the era of digital transformation, automation has become the key to improving the operational efficiency of companies. No exception for PT Telkom Cibitung, which is a branch of PT Telkom Indonesia, which is a State-Owned Enterprise (BUMN) engaged in telecommunications services [1]. One of its main tasks is to focus on managing production availability data, especially for the sales division. Currently, data retrieval from the Google Sheets database is still done manually by employees, which is then shared via the Telegram group. This is inefficient, especially if the amount of data is large and numerous. With the increasing popularity of Telegram and its Bot features, a data retrieval automation solution is proposed to facilitate data access and increase productivity. This study proposes the implementation of a Telegram Bot that is integrated with Google Sheets using node.js technology, as a practical solution to improve the efficiency of the sales division at PT Telkom Cibitung. Previous studies have explored the use of Telegram Bots, but their implementation in the workplace, especially in telecommunications companies, is still rare.

Telegram is a cloud-based messaging application designed for speed and security, has become a popular platform since it was founded by Pavel Durov in 2013. The Telegram application has advantages in several features, especially in data security and the ability to send large files, another superior feature is chatbot, this feature is very useful for businesses because it can be customized for various needs. Telegram provides the Bot API feature which is a technology open-source, this allows developers to build bot applications that are integrated with the Telegram system. By using API In this, bots can be programmed to interact with various services such as Google Sheets, to send or retrieve data automatically real-time.

Google Sheets is part of the service Google, which is an online spreadsheet application that is often used to store and manage data. This service makes it easier for users to manage data analytically. With various innovative features developed, Google Sheets enables strong integration with Telegram bots to increase business efficiency in automated data management.

Research relevant to this topic, this research shows various implementations of Telegram Bot in various sectors. Such as:

1. Research [2] explores the use of Telegram bots in distance learning, showing efficiency in managing file-based exams through the Telegram platform.
2. Research by [3], discusses the use of Google Sheets for data management, which supports the use of Sheets as an effective tool in data management and analysis.
3. Research [4] explores the application of Webhook to automate data recap, which enables communication real-time between the application and the server, improving responsive system.
4. A study by [5] presents the implementation of a Telegram bot to disseminate real-time weather forecasts, emphasizing the importance of APIs in delivering efficient information.

5. Research by [6] shows efficiency in credit transactions using Telegram bots, which automatically process requests and payments, strengthening the ease of service with bot API integration.

All of this research shows that Teleram bot technology can be adapted for a variety of purposes, including integrated data visualization with Google Sheets.

Therefore, this research is expected to be a basis for further research and provide benefits to the company. In this project, the main challenge faced by the sales division of PT Telkom Cibitung is the process of manually retrieving data from Google sheets to the Telegram platform. This manual method takes a long time and is prone to errors, which will later have an impact on the inaccuracy of information disseminated in the Telegram group. This can hinder effective decision-making within the team. In addition, the separation between onsite and ondesk work units further worsens the coordination flow, because information taken manually tends to be delayed in delivery. In this condition, the manual work process not only slows down the distribution of information but also reduces the efficiency and productivity of the team in carrying out more strategic tasks. Therefore, the use of Telegram bots integrated with Google Sheets is seen as an effective solution to automate data retrieval, speed up the information distribution process, and improve work efficiency in the sales division as a whole.

2. Theoretical Basis

To establish a robust conceptual and technical framework for understanding the proposed information system, it is essential to define the fundamental concepts and supporting theories that underpin its design and analysis.

2.1. Fundamental Concepts (Information Systems and Digital Transformation)

A system can be defined as a network of interconnected procedures that collectively perform an activity or achieve a specific goal. Alternatively, it can be viewed as a collection of several components that are interconnected, forming a unity to achieve certain objectives. Systems possess distinct characteristics, including components (interacting parts forming a unified whole), a boundary (defining separation from other systems), an environment (external influencing factors), an interface (connecting medium), input (energy introduced), output (processed energy), a processor (transforms input to output), and a goal (the purpose that determines necessary inputs and generated outputs). Understanding these characteristics is crucial for evaluating the completeness and robustness of any system design.

Information, a core output of any system, is defined as the result of data processing in a specific manner, rendering it more meaningful and useful for the recipient. The value of information is determined by its benefits relative to the cost of obtaining it. The continuous "Information Cycle" describes how data is transformed into information by a model, leading to decisions and actions by recipients, which in turn generate new data.

The pervasive influence of Information Technology (IT) in reshaping various scientific fields and its indispensable role for businesses in Indonesia cannot be overstated. Computerized systems, particularly database applications and information systems, are pivotal for organizations, significantly enhancing decision-making processes, optimizing operational workflows, and facilitating the efficient management of vast quantities of corporate data. This research exemplifies a practical application of these principles within a specific corporate context, directly contributing to PT. Telkom Cibitung's broader digital transformation efforts.

2.2. Key Technologies

The proposed system's functionality is built upon the integration of several key technologies, each contributing specific capabilities essential for real-time data visualization and efficient communication.

Telegram Bot API and its Capabilities: Telegram is a cloud-based messaging application, widely recognized for its speed, security, and robust features, including a highly adaptable chatbot functionality that has gained significant popularity since its inception in 2013. The Telegram Bot API serves as an open-source, HTTP-based interface specifically designed for developers to build and connect bots with the Telegram system.¹ This API enables bots to be programmed for real-time interaction with various services, facilitating real-time notifications, supporting multiplatform access, and handling diverse commands. The inherent data security features of Telegram, where an intermediary server handles all encryption and communication with the Bot API, further bolster its appeal for corporate applications. The ability to send and retrieve data in real-time is a core capability critically leveraged in this research.

Google Sheets API for Data Management: Google Sheets, an integral component of Google services, functions as an online spreadsheet application extensively utilized for storing and managing data. The Google Sheets API is a powerful RESTful interface that empowers developers to programmatically interact with spreadsheet data, enabling operations that would typically be performed manually within the Google Sheets user interface.¹ Its functionalities include reading and writing spreadsheet data, appending new rows, formatting cells and ranges, creating and managing worksheets, and performing batch updates. Crucially, it supports secure access authentication via OAuth 2.0, making it ideal for automating reporting and data synchronization while maintaining data integrity and privacy.¹ This programmatic control allows for the use of Google Sheets as a backend database for applications.

Node.js in Backend Development: Node.js is a server-side runtime environment for JavaScript that has fundamentally transformed web development by allowing developers to use a single language for both client-side and server-side programming, thereby fostering the "JavaScript everywhere" paradigm.¹ Its distinctive event-driven, non-blocking I/O architecture makes it exceptionally well-suited for building high-performance, real-time applications and efficiently managing numerous concurrent connections with minimal overhead.⁶ In this project, Node.js plays a pivotal role in running the Telegram bot, facilitating its responsiveness and scalability. The choice of Node.js is a strategic technical decision that directly addresses the business requirement for responsiveness and immediate data availability, as its

architecture is inherently optimized for real-time operations and handling dynamic data flows.

Webhook for Real-time Communication: The Webhook method is a highly efficient mechanism employed in the creation of Telegram bots to facilitate real-time interaction between applications and servers, primarily through HTTP POST requests.¹ This approach offers a significantly faster communication mechanism for responding to and processing information compared to traditional Long-Polling methods, which aligns perfectly with the project's real-time data needs. Webhooks enable a push-based notification system, where information is sent immediately as an event occurs, significantly reducing data latency and server load compared to traditional polling mechanisms.⁸ Successful implementation requires a securely defined URL (HTTPS) and the utilization of the Telegram API Token to enable seamless and authenticated interaction.¹ This strategic adoption of webhook-based communication patterns is becoming essential for building truly responsive and efficient enterprise applications, allowing organizations to move from reactive data retrieval to proactive information dissemination.

The seamless integration of these APIs—Telegram Bot API and Google Sheets API—forms the core of the proposed solution. This integration represents a strategic leveraging of modular, extensible services provided by these APIs, which significantly reduces the development burden. Developers can rapidly compose new business capabilities, such as real-time data access for sales, by assembling functional components enabled by these robust, well-documented APIs.² This inherent modularity of API-driven development directly facilitates the principles of Rapid Application Development (RAD) by enabling quick assembly of functional components, thereby accelerating the time-to-market for new solutions. This highlights how the widespread adoption and maturity of APIs are not just a technical evolution but a fundamental shift in how organizations can achieve business agility, enabling rapid innovation and integration without building every component from scratch.

2.3. Software Development Methodologies

The project employed a combination of two prominent software development methodologies to ensure both speed and adaptability throughout the development lifecycle.

Rapid Application Development (RAD): The Rapid Application Development (RAD) model was specifically utilized for its ability to facilitate rapid prototyping and the quick generation of functional prototypes.¹ This approach aligns with the project's objective of delivering a practical and timely solution to PT. Telkom Cibitung's immediate operational challenges. RAD emphasizes a component-based construction approach and accelerated development cycles, which were instrumental in the efficient creation of the Telegram bot, particularly when integrated with modern technologies like Node.js.¹ RAD's iterative nature allows for early accommodation of user needs and rapid adaptation to feedback, which is crucial for validating the concept and ensuring early user buy-in for a novel internal tool.

Scrum Framework: The Scrum framework was applied as a complementary agile methodology. Its iterative and incremental nature proved invaluable for managing potentially evolving requirements and enhancing the overall adaptability of the project throughout its development lifecycle.¹ Scrum is particularly well-suited for developing object-oriented software in complex environments and its support for gradual and progressive feature evolution was crucial for developing a system that needed to seamlessly integrate with existing company workflows and respond effectively to continuous user feedback.¹ This combined approach signifies a sophisticated understanding of software development, where methodologies are pragmatically blended to address specific project constraints, such as the need for both speed and adaptability.

3. Research Methods

This study was conducted through a series of systematic stages to develop a Telegram bot integrated with Google Sheets, aiming to address data visualization and communication inefficiencies at PT Telkom Cibitung. The methodological framework comprises research phases, software requirement analysis, and algorithmic implementation, as detailed below. To ensure the system aligns with user needs and organizational goals, the research follows several key stages:

3.1. Data Collection Techniques

- a. **Observation:** Direct observation was carried out within the sales division of PT Telkom Cibitung to gain a comprehensive understanding of existing workflows and the manual data management practices that hinder operational efficiency.
- b. **Literature Review:** A review of academic and industrial literature concerning Telegram bots, data automation, and digital transformation in telecommunication enterprises was conducted to strengthen the theoretical foundation of this study.

3.2. Software Development Models

The system was developed using two complementary software development methodologies:

- a. **Rapid Application Development (RAD).** RAD was adopted to facilitate the rapid prototyping of the Telegram bot using component-based development. As described by [7], RAD allows accelerated system delivery through iterative refinements. Node.js was selected as the development platform to optimize performance and reduce time-to-deploy.
- b. **Scrum Framework.** The Scrum methodology was employed to accommodate dynamic user requirements and evolving system specifications. As an iterative and incremental model, Scrum supports continuous improvement and agile responses to feedback throughout the development lifecycle [7]. [8].

3.3. Software Requirements Analysis

- a. **Problem Identification.** The primary issues identified include time-consuming manual data retrieval processes that are error-

- prone and impact decision-making accuracy. Additionally, the division of sales teams into onsite and ondesk units introduces coordination challenges due to geographic separation.
- b. Requirements Analysis, Functional Requirements
 - The bot must retrieve data from Google Sheets.
 - It should transmit the data to a designated Telegram group.
 - Users must interact with the bot via predefined command inputs.
 - Non-Functional Requirements
 - The bot should perform reliably and without interruption.
 - Access to both Google Sheets and Telegram must be secure and private.
 - The bot must respond to commands promptly and efficiently.
 - System Requirements
 - Features for automatic data fetching from Google Sheets.
 - Telegram integration for real-time communication.
 - Command processing functionality from Telegram users.
 - User Requirements
 - User-friendly interface with intuitive access to desired data.
 - Fast and streamlined interaction for non-technical users.

3.4. Algorithmic Implementation

- a. Webhook Algorithm, The system employs a Webhook mechanism to facilitate real-time interactions between the Telegram bot and the backend server via HTTP POST requests. Compared to the traditional long-polling method, Webhook offers faster and more efficient data exchange [5]. To enable this process, a valid HTTPS endpoint is established, acting as a communication channel for input-processing-output flows. The implementation utilizes the Telegram API token to authorize system-level bot interactions [6]
- b. Data Processing Algorithm. Data processing leverages the Google Sheets API to enable dynamic access and manipulation of spreadsheet data. The Telegram bot communicates with the API to retrieve, update, and display data as requested by users. This interaction is secured via OAuth 2.0 authentication, ensuring proper authorization and secure access [7], [9]. The system architecture is designed to maintain consistency and data integrity while automating frequent reporting tasks. These supplementary stages provide a valuable framework for discussing the progression and maturity of the prototype, implying that the "Implementation and Maintenance" phase is not a terminal point but rather a continuous cycle of refinement, aligning with the concept of a "Matured Product." This section offers a comprehensive analysis of the existing logistics operations at the UTD PMI Karawang, encompassing its historical context, current organizational structure, detailed operational procedures, and the identified core shortcomings [10], [11].

3.5. Telegram Bot Creation Process

The foundational first step in creating the Telegram bot involved interacting with Telegram's official **BotFather** account. This account serves as the central hub for managing and configuring all user-created bots. The process is straightforward, requiring the configuration of essential bot properties, including setting a unique name for the bot and uploading a suitable profile picture, which are crucial for user recognition and branding. Upon successful completion of this configuration, BotFather automatically generates a unique **API Token**. This token is a critical credential, serving as the primary interface for developers to programmatically interact with the Telegram system and control the bot's functionalities. This initial setup provides the necessary authentication for the bot to operate within the Telegram ecosystem.

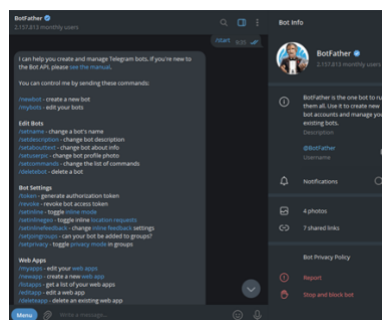


Figure 1: Telegram Bot Creating

3.6 Core System Architecture and Coding Implementation

The bot's core logic and operations are developed using JavaScript, specifically within the Node.js runtime environment. This choice leverages Node.js's asynchronous capabilities for efficient real-time processing, crucial for a responsive data visualization tool. A critical security measure implemented is the use of environment variables for storing sensitive information, such as the bot's API token. This practice prevents hardcoding of credentials directly within the source code, thereby significantly reducing vulnerability to unauthorized access. This proactive approach to security is vital for an enterprise application handling sensitive sales data, ensuring the confidentiality and integrity of information from the outset.

The development process also involved the meticulous initialization and configuration of all necessary external libraries and modules (dependencies) that the bot relies upon for its various functionalities, ensuring a stable and operational environment. To manage multi-step user interactions and maintain context across conversations, a dedicated object (e.g., detailSalesSearchState) was created and utilized for

state management. Specific functions were developed to handle user commands. The /start command functionality processes the initial user interaction, sending a welcome message and providing instructions on how to use the bot. The /Cari (Search) command functionality serves as the entry point for users to initiate a sales data search, prompting the bot to enter a search mode. An asynchronous function (async function) was implemented for user input handling and data retrieval logic, continuously listening for and processing incoming user messages. This function intelligently checks the bot's current search state and, if in search mode, attempts to retrieve relevant sales data from the linked Google Sheets based on user-provided 'Kode SF' (Sales Force Code) or 'Email'.



```

1  const { Bot, webhookCallback } = require("grammy");
2  const express = require("express");
3  const { google } = require('googleapis');
4  const { TelegramBot } = require('node-telegram-bot-api');
5
6  require("dotenv").config();

```

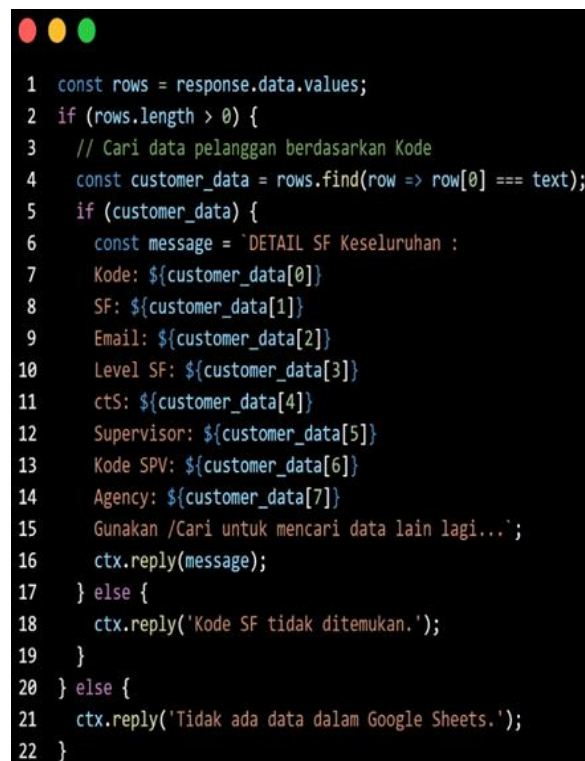
Figure 2: Core System Architecture and Coding google spreadsheet Implementation

Robust error handling was integrated through the use of try-catch blocks within the code. This mechanism is vital for gracefully managing potential errors that may arise during interactions with the Google Sheets API or other system components, ensuring that the bot provides clear and informative error messages to the user rather than crashing. This meticulous attention to error handling directly contributes to fulfilling the non-functional requirement for consistent and error-free operation and enhances the overall reliability of the system.

Finally, a specific code segment was developed for production deployment and Webhook configuration. This includes setting up Express to manage the Webhook, which allows Telegram to push updates to the bot in real-time, ensuring immediate data delivery. Additionally, code was implemented to handle proper termination signals, allowing the bot to be stopped gracefully.

3.7. Authentication and API Integration

A crucial aspect of the system is the secure authentication of the bot's access to Google Sheets via its API. The project explicitly implemented code dedicated to this authentication, which is fundamental for enabling the bot to read and manipulate data within the spreadsheet. This process leverages the OAuth 2.0 protocol, an industry-standard for secure authentication and authorization. OAuth 2.0 ensures that the bot's access is controlled and adheres to privacy best practices, allowing it to interact with user data without compromising security.



```

1  const rows = response.data.values;
2  if (rows.length > 0) {
3    // Cari data pelanggan berdasarkan Kode
4    const customer_data = rows.find(row => row[0] === text);
5    if (customer_data) {
6      const message = `DETAIL SF Keseluruhan :
7      Kode: ${customer_data[0]}
8      SF: ${customer_data[1]}
9      Email: ${customer_data[2]}
10     Level SF: ${customer_data[3]}
11     ctS: ${customer_data[4]}
12     Supervisor: ${customer_data[5]}
13     Kode SPV: ${customer_data[6]}
14     Agency: ${customer_data[7]}
15     Gunakan /Cari untuk mencari data lain lagi...`;
16     ctx.reply(message);
17   } else {
18     ctx.reply('Kode SF tidak ditemukan.');
```

Figure 3: Authentication and API Integration

Once successfully authenticated, the bot executes specific dynamic data retrieval logic. This logic dynamically searches the Google Sheet based on user input (e.g., 'Kode SF' or 'Email') and extracts the relevant information. The system incorporates conditional response logic

based on the outcome of the data retrieval. If the requested sales data is found within the Google Sheet, the bot will respond by displaying the complete details to the user. Conversely, if the provided 'Kode SF' or 'Email' does not match any records, the bot will provide a clear and informative error message indicating that the data was not found. This intelligent response mechanism enhances user experience by providing immediate and relevant feedback.

4. Results and Discussion

This section presents the findings from the system implementation and testing, discusses their implications for PT. Telkom Cibitung, and details the supporting hardware and software environment.

4.1. Functional Testing and User Response Analysis

The interface design testing for the proposed system was rigorously conducted using the Black Box testing method.¹ This approach focused exclusively on evaluating the functional requirements of the developed software by examining its inputs and outputs, without delving into its internal structural details.

The initial basic functionality test involved the /start command. Upon receiving this command, the bot successfully responded by displaying a welcome message and providing clear instructions for its usage, indicating proper initialization and responsiveness of the program. Subsequently, the search functionality test (/Cari command) confirmed the bot's ability to correctly interpret and initiate multi-step interactions, prompting the user to select a search method (either by "Kode SF" or "Email").

For valid data search scenarios, when users provided 'Kode SF' or 'Email' that existed within the Google Sheets database, the bot successfully retrieved and displayed the complete sales details. This outcome validated the accuracy of data retrieval from Google Sheets and the bot's ability to present comprehensive information. The testing also covered invalid data/command handling scenarios, demonstrating the bot's robust error management. For non-existent 'Kode SF' or 'Email', the bot appropriately displayed an "Code or email not found" message. Similarly, for commands not adhering to the provided instructions, the bot responded with an "Invalid command" message.

The comprehensive series of tests collectively confirmed the high performance, reliability, and responsiveness of the Telegram bot integrated with Google Sheets. This included successful validation of the Webhook functionality, ensuring real-time data communication. The following table summarizes key testing scenarios and their outcomes:

Table1: Key Testing Scenarios and Outcomes

Scenario	Expected Outcome	Actual Outcome
/start command	Welcome message/instructions	Successful
/Cari command	Prompt for search method	Successful
Valid 'Kode SF' search	Display sales details	Successful
Valid 'Email' search	Display sales details	Successful
Invalid 'Kode SF' / 'Email' search	"Code/email not found" message	Successful
Invalid command	"Invalid command" message	Successful

This systematic validation provides concrete, empirical evidence of the bot's functionality and reliability across different use cases, directly supporting the claims regarding its performance and accuracy.

4.2 System Performance and Efficiency Gains

The implementation of the Telegram bot at PT. Telkom Cibitung yielded significant improvements in operational efficiency and information access. The bot demonstrably accelerated the process of accessing sales data, drastically reducing the time previously consumed by manual retrieval methods. This directly addresses one of the core problems identified in the introduction, where manual processes were a major bottleneck.

The automation of data retrieval and dissemination led to a substantial improvement in the work efficiency of the sales division. By offloading repetitive manual tasks, employees are now able to reallocate their time to more strategic and value-added activities. This transformation in human capital utilization is not merely about process optimization but fundamentally redefines the role of employees, liberating them to focus on more analytical or customer-facing tasks.

Furthermore, the automated retrieval mechanism inherently reduced the incidence of data errors, leading to more accurate and reliable information being disseminated. This, in turn, fostered improved coordination and communication between the onsite and ondesk sales units, mitigating previous bottlenecks caused by delayed information. The bot received overwhelmingly positive feedback from users at PT. Telkom Cibitung. This high level of user acceptance and satisfaction is a strong indicator of the system's usability and its successful integration into daily operations. The positive outcomes, such as faster information access, improved efficiency, and reduced errors, serve as strong empirical validation for the chosen Rapid Application Development (RAD) and Scrum methodologies. The iterative and incremental nature of these agile methods likely facilitated rapid testing, quick adjustments based on early user feedback, and continuous refinement, ensuring the final product closely matched user needs and organizational requirements.

4.3 Hardware and Software Environment

The successful operation of the Telegram bot system requires specific hardware and software configurations for both the server and client environments.

Table1: Hardware and Software Specifications

Category	Component	Specification
----------	-----------	---------------

Hardware	Server CPU	Minimal 2 GHz
	Server RAM	Minimal 4 GB
	Server Storage	Minimal 100 GB
	Client Device RAM	2 GB
	Client Browser	Latest version
Software	Server OS	Windows Server / Linux
	Client OS	Windows / Linux / Mac
	Development Software	VSCode, Node.js
	Application	Telegram

These specifications provide the essential practical context for the implemented solution, allowing other researchers, practitioners, or organizations to assess feasibility and plan infrastructure if considering replication or adaptation.

5. Conclusion

The implementation of the Telegram bot at PT. Telkom Cibitung has successfully addressed and significantly mitigated the previously identified challenges associated with manual data retrieval and dissemination. This research clearly demonstrates the efficacy of the automated solution in transforming operational workflows within the sales division.

The deployment of the bot has yielded several significant and measurable improvements. There has been a substantial reduction in both the time required for and the errors inherent in data acquisition from Google Sheets, directly improving data quality and reliability. This enhancement has, in turn, fostered improved coordination and facilitated more effective decision-making within the sales division, as accurate and timely data is now readily available to both onsite and ondesk units. Overall work efficiency and productivity across the sales team have seen a marked increase, as the burden of repetitive manual tasks has been alleviated. This shift allows employees to focus on more strategic and value-added activities, contributing to higher employee satisfaction and better utilization of skilled labor. The overwhelmingly positive user feedback and high acceptance rates from the employees at PT. Telkom Cibitung further underscore the system's practical value and its successful integration into daily operations.

This research makes a significant contribution to advancing better data management practices within corporate settings and provides a tangible, successful example of chatbot technology application in an enterprise environment. The solution's inherent design, leveraging widely available and open-source technologies like Telegram Bot API, Google Sheets API, and Node.js, inherently supports its scalability and replicability. This suggests that the developed solution is not a one-off, bespoke fix but a generalized and adaptable model for addressing similar data access and visualization challenges across various industries, departments, or organizational scales. This offers a practical blueprint for organizations seeking cost-effective, readily implementable, and adaptable solutions for internal data automation.

While the current implementation has achieved its primary objectives, it is acknowledged that the system can be further enhanced. Future work could involve integrating the bot with additional enterprise resource planning (ERP) or customer relationship management (CRM) systems to provide a more holistic view of sales operations. Developing more sophisticated data analysis and reporting features, such as trend analysis or predictive analytics, would further empower decision-makers. Additionally, exploring the application of machine learning for more intelligent data insights or advanced conversational capabilities could enhance the bot's utility and user experience. The documented success of this Telegram bot in a corporate setting for internal operational efficiency and real-time data access signifies a shift in the perceived and actual utility of chatbots, highlighting their growing importance as integral tools for internal operational efficiency and intuitive interaction with enterprise systems.

References

- [1] Wijoyo, A., Bay, R., Karina, H., Panjaitan, D., Oktavia, S., & Gilang, R. (2023). PERANAN SISTEM INFORMASI MANAJEMEN PEMASARAN DI PT. TELKOM INDONESIA (Studi Kasus : PT . TELKOM). 1(3), 355–358.
- [2] Heri Khariono, Rizky Parlita, Kusuma, H. A., & Setyawan, D. A. (2021). Pemanfaatan Bot Telegram Sebagai E-Learning Ujian Berbasis File. *Jurnal Informatika Polinema*, 7(4), 65–72. <https://doi.org/10.33795/jip.v7i4.696>
- [3] Nursalim, K., & Yusuf Irianto, S. (2022). Analisis Bot Telegram Untuk Artificial Intelligence Helpdesk Online Pada PT Telkom Akses Witel Lampung. *Prosiding Seminar Nasional ...*, 210–216. <https://jurnal.darmajaya.ac.id/index.php/PSND/article/view/3248%0Ahttps://jurnal.darmajaya.ac.id/index.php/PSND/article/download/3248/1459>
- [4] Utomo, A. B., Latipah, L., & Achlaq, M. M. (2023). Implementasi Telegram Bot untuk Proses Automatisasi Rekapitan Data Menggunakan Metode Webhook. *Jurnal Ilmu Komputer Dan Bisnis*, 14(2), 202–213. <https://doi.org/10.47927/jikb.v14i2.532>
- [5] Istiana, T., Indra A, R., Budhi Dharmawan, G. S., & Prakoso, B. (2020). Pengembangan Sistem Diseminasi Prakiraan Cuaca Menggunakan Aplikasi Bot Telegram dengan Metode Webhook. *Elektron : Jurnal Ilmiah*, 12(1), 41–47. <https://doi.org/10.30630/eji.12.1.159>
- [6] Nufusula, R., & Susanto, A. (2018). Rancang Bangun Chat Bot Pada Server Pulsa Menggunakan Telegram Bot API. *Journal of Information System*, 3(1), 80–88.
- [7] Lenardo, G. C., Herianto, & Irawan, Y. (2020). Pemanfaatan Bot Telegram sebagai Media Informasi Akademik di STMIK Hang Tuah Pekanbaru. *JTIM : Jurnal Teknologi Informasi Dan Multimedia*, 1(4), 351–357. <https://doi.org/10.35746/jtim.v1i4.59>
- [8] Nastiti, S. D., Ramadan, D. N., & Tulloh, R. (2021). Sistem Monitoring Untuk Laporan Gangguan Indihome Dengan Bot Telegram. *E-Proceeding of Applied Science*, 7(5), 1892–1907.
- [9] Salwa@jadibumn.id. (2024). Sales Telkom BUMN – Mengapa Profesi Sales di Telkom BUMN Layak Dicoba? <https://jadibumn.id/sales-telkom-bumn/>
- [10] Setiaji, H., & Papatungan, I. V. (2018). Design of Telegram Bots for Campus Information Sharing. *IOP Conference Series: Materials Science and Engineering*, 325(1), 2–8. <https://doi.org/10.1088/1757-899X/325/1/012005>
- [11] Susilo, B., Hanyokro Kusuma, G., Hayatul Fikri, M., Saputri, R., Aulia Putri, R., Rohimah, S., Luthfi Hamzah, M., & Sultan Syarif Kasim Riau, N. (2023). Rancang Bangun Sistem Informasi Keuangan Pada Kantor Lurah Kotabaru Reteh Dengan Metode Rapid Application Development (Rad). *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1), 17–28. <https://journal.al-matani.com/index.php/jtisi/article/view/323>